WASSA 2023

The 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis

Proceedings of the Workshop

July 14, 2023

The WASSA organizers gratefully acknowledge the support from the following sponsors.

Gold

Google

©2023 Association for Computational Linguistics

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

ISBN 978-1-959429-87-6

Introduction

While the first editions of the Workshop on Computational Approaches to Subjectivity Analysis (WASSA) focused mainly on sentiment and subjectivity analysis, the scope of this workshop has become both wider and deeper over the years, expanding the number of languages and tasks that are considered, while simultaneously looking at more specific subjective tasks.

The topics of this edition of WASSA range from exploring how harmful language proliferates in social media and news, to looking at the effects of using large pretrained language models on well-known sentiment/emotion analysis tasks, or creating resources for new tasks that focus on subjective language. Regarding languages, in this edition English is unsurprisingly the most represented (in 29 papers), with 2 papers each including Mandarin Chinese, Dutch, German, Hindi, and Spanish. This is followed by a long tail of languages that appear in only one paper (Algerian Arabic, Basque, Czech, Dari, French, Gujarati, Italian, Japanese, Kannada, Korean, Malayalam, Maltese, Modern Standard Arabic, Norwegian, and Tamil). Two papers also address multilingual emotion detection.

This year, WASSA offered two shared tasks: Shared-Task 1: Empathy Detection and Emotion Classification, which builds upon the previous shared task with a new track on empathy, emotion and self-disclosure detection in conversation at the speech-turn level, using a new, unpublished dataset. Shared-Task 2: Emotion Classification on code-mixed text messages (Roman Urdu + English), on the other hand, is a novel shared task focusing on multi-class and multi-label emotion classification on code-mixed (Roman Urdu + English) text messages. Two papers also submitted work on multilingual emotion analysis.

For the main workshop, we received 71 papers, one of the largest number of submissions for WASSA in its history. While we accepted a comparable large number, namely 43 (60.6%), we would like to point out that the competition has been quite tough – most submitted papers were of very high quality. For Shared Task 1 we received 12 system description paper submissions, while for Shared Task 2 we received 7 papers. In total, 61 papers will be presented at the workshop.

We also decided to award the best paper. Starting with the reviewer scores, the organizers of the workshop anonymously proposed candidates and then voted on them. The winner of the Best Paper Awards of this year's WASSA is *Multilingual Language Models are not Multicultural: A Case Study in Emotion by Shreya Havaldar, Sunny Rai, Bhumika Singhal, Langchen Liu Sharath Chandra Guntuku, and Lyle Ungar*.

On top of the main workshop and shared task presentations, we are happy to have two invited speakers who will hold keynotes at WASSA 2023: Professor Emily Öhman from Wakeda University and Professor David Jurgens from the University of Michigan.

We would like to thank the ACL Organizers and Workshop chairs for their help and support during the preparation. We thank Google for their gold sponsorship. We also thank the OpenReview support team for their technical support. Finally, we especially thank the program committee for the time and effort they spent on reviewing, especially to those who were willing to perform emergency reviews. The quality of the program is a reflection of this hard work.

Jeremy Barnes, Orphée De Clercq, Roman Klinger

WASSA 2023 Chairs

Organizing Committee

Program Chairs

Jeremy Barnes, University of the Basque Country UPV/EHU, Spain
Orphée De Clercq, Ghent University, Belgium
Roman Klinger, University of Stuttgart, Germany

Shared Task 1 Chairs

Valentin Barriere, Centro Nacional de Inteligencia Artificial, Chile
Salvatore Giorgi, University of Pennsylvania, USA
Joao Sedoc, New York University, USA
Shabnam Tafreshi, University of Maryland, USA

Shared Task 2 Chairs

Iqra Ameer, Yale University, USA
Necva Bölücü, Hacettepe University, Republic of Türkiye
Hua Xu, Yale University, USA
Ali Al Bataineh, Norwich University, USA

Program Committee

Program Committee

Rodrigo Agerri, University of the Basque Country
Wazir Ali, College of Business Management (CBM)
Murali Balusu, Georgia Institute of Technology
Sivaji Bandyopadhyay, Jadavpur University
Tilman Beck, TU Darmstadt
Emerson Cabrera Paraiso, Pontifícia Universidade Católica do Paraná
Luna De Bruyne, Universiteit Gent
Loic De Langhe, Universiteit Gent
Koji Eguchi, Hiroshima University
Annerose Eichel, University of Stuttgart, Universität Stuttgart
Thomas Nikolaus Haider, Universität Passau
Mohamad Hardyman Barawi, Universiti Malaysia Sarawak
Yoshihiko Hayashi, Waseda University
Kokil Jaidka, National University of Singapore
Salud María Jiménez-Zafra, Universidad de Jaén
Abhinav Joshi, Indian Institute of Technology, Kanpur
Akbar Karimi, Phillips-Universität Marburg
Elma Kerz, Rheinisch Westfälische Technische Hochschule Aachen
Mayank Kulkarni, Amazon
Sofie Labat, Universiteit Gent
Els Lefever, Ghent University
Natalia V Loukachevitch, Lomonosov Moscow State University
Petter Mæhlum, University of Oslo
Aaron Maladry, Universiteit Gent
María-Teresa Martín-Valdivia, Universidad de Jaén
Eugenio Martinez-Camara, Universidad de Jaén
Amita Misra, Amazon
Ashutosh Modi, IIT Kanpur
Constantin Orasan, University of Surrey
Jessica Ouyang, University of Texas at Dallas
Flor Miriam Plaza del Arco, Bocconi University
Joan Plepi, Math and CS, Marburg University
Anil Ramakrishna, Amazon
Irene Russo, Consiglio Nazionale delle Ricerche
Iñaki San Vicente, Orai NLP Technologies
Salim Sazzed, Old Dominion University
Pranaydeep Singh, Universiteit Gent
Efstathios Stamatatos, University of the Aegean
Zeerak Talat, Simon Fraser University
Sai Kiran Tanikella, IIT Kanpur, Indian Institute of Technology, Kanpur
Samia Touileb, University og Bergen, Norway
Amine Trabelsi, Université de Sherbrooke
Enrica Troiano, Vrije Universiteit Amsterdam
L. Alfonso Ureña, Universidad de Jaén
Aswathy Velutharambath, University of Stuttgart, Universität Stuttgart
Jacky Visser, University of Dundee

Jin Wang, Yunnan University
Charles Welch, University of Marburg
Michael Wiegand, Alpen-Adria Universität Klagenfurt
Tak-Lam Wong, Douglas College
Arkaitz Zubiaga, Queen Mary University of London

Invited Speakers

David Jurgens, University of Michigan
Emily Öhman, Wakeda University

Keynote Talk: Invited 1

David Jurgens
University of Michigan
2023-07-14 12:30:00 – Room: **Harbour C**

Abstract: NLP studies of communication often focus on the individual: What we say, when we say it, and how we say it. Yet, the larger social context beyond the individual also plays an important role in our communication — just think of things you can say to your friends but not your parents. How does the social context influence our communication style and content? In this talk, I will describe recent work from my group studying the influence of this context by examining how we choose who to communicate with, how we interpret messages, and how we phrase messages. Across these studies, I will motivate a causal approach for NLP when studying communication behavior to move beyond descriptive analyses to more precise estimates of the effects of social context.

Bio: David Jurgens is an assistant professor at the University of Michigan School of Information where he leads the Blablablab. He holds a PhD in Computer Science from the University of California, Los Angeles. His research focuses on the intersection between NLP and computational social science venues and has won the Cozzarelli Prize, Cialdini Prize, best paper at ICWSM and W-NUT, and best paper nomination at ACL and Web Science.

Keynote Talk: Invited 2

Emily Öhman
Waseda University

2023-07-14 16:00:00 – Room: **Harbour C**

Abstract: Affective Datafication of Narratives: measuring affect, emotion, and mood in literary texts
Our understanding of affect, emotion, and mood - despite the distinct nuances each term holds - often becomes blurred, leading to a usage that is almost interchangeable, particularly within sentiment analysis and NLP. In contrast, traditional fields such as literary studies hold on to more rigid definitions of these terms and how they are understood both in theory and practice. This can easily foster a disconnect between emerging fields such as computational literary studies and the more established qualitative counterparts. This disconnect unfortunately hinders the free exchange of innovative research ideas and methodologies. This talk aims to bridge this gap, highlighting the unique roles of affect, emotion, and mood in narratives and how we can attempt to robustly measure them. We will delve into the interplay of these terms, exploring how they shape and are shaped by authors, readers, and researchers focusing on the operationalization and translation involved in the analysis of emotion-laden phenomena. This exploration will underscore the need for a more comprehensive and nuanced understanding, encouraging synergy between tradition and innovation in emotion detection in general and literary research in particular.

Bio: Emily Öhman is currently a tenure-track Assistant professor of Digital Humanities at Waseda University. She received her PhD in Language Technology from the University of Helsinki, where her work centered on building multilingual emotion detection resources for downstream tasks.

Her research interests lie within digital humanities and NLP, more specifically sentiment analysis and emotion detection, often doing collaborations with various disciplines such as history, literature, and political science. Her recent projects have focused on negative emotions in literature using affect as a proxy for the literary concept of mood and most recently contrasting the semantic spaces of shame and guilt in Japanese and English social media posts.

Table of Contents

PESTO: A Post-User Fusion Network for Rumour Detection on Social Media
Erxue Min and Sophia Ananiadou 1

Sentimental Matters - Predicting Literary Quality by Sentiment Analysis and Stylometric Features
Yuri Bizzoni, Pascale Feldkamp Moreira, Mads Rosendahl Thomsen and Kristoffer Nielbo . . . 11

Instruction Tuning for Few-Shot Aspect-Based Sentiment Analysis
Siddharth Varia, Shuai Wang, Kishaloy Halder, Robert Vacareanu, Miguel Ballesteros, Yassine Benajiba, Neha Anna John, Rishita Anubhai, Smaranda Muresan and Dan Roth 19

You Are What You Read: Inferring Personality From Consumed Textual Content
Adam Sutton, Almog Simchon, Matthew Edwards and Stephan Lewandowsky 28

UNIDECOR: A Unified Deception Corpus for Cross-Corpus Deception Detection
Aswathy Velutharambath and Roman Klinger 39

Discourse Mode Categorization of Bengali Social Media Health Text
Salim Sazzed 52

Emotion and Sentiment Guided Paraphrasing
Justin J Xie and Ameeta Agrawal 58

Emotions in Spoken Language - Do we need acoustics?
Nadine Probol and Margot Mieskes 71

Understanding Emotion Valence is a Joint Deep Learning Task
Gabriel Roccabruna, Seyed Mahed Mousavi and Giuseppe Riccardi 85

Czech-ing the News: Article Trustworthiness Dataset for Czech
Matyas Bohacek, Michal Bravansky, Filip Trhlík and Vaclav Moravec 96

Towards Detecting Harmful Agendas in News Articles
Melanie Subbiah, Amrita Bhattacharjee, Yilun Hua, Tharindu Sandaruwan Kumarage, Huan Liu and Kathleen McKeown 110

GSAC: A Gujarati Sentiment Analysis Corpus from Twitter
Monil Gokani and Radhika Mamidi 129

A Dataset for Explainable Sentiment Analysis in the German Automotive Industry
Andrea Zielinski, Calvin Spolwind, Henning Kroll and Anna Grimm 138

Examining Bias in Opinion Summarisation through the Perspective of Opinion Diversity
Nannan Huang, Lin Tian, Haytham M. Fayek and Xiuzhen Zhang 149

Fluency Matters! Controllable Style Transfer with Syntax Guidance
Ji-Eun Han and Kyung-Ah Sohn 162

ChatGPT for Suicide Risk Assessment on Social Media: Quantitative Evaluation of Model Performance, Potentials and Limitations
Hamideh Ghanadian, Isar Nejadgholi and Hussein Al Osman 172

Unsupervised Domain Adaptation using Lexical Transformations and Label Injection for Twitter Data
Akshat Gupta, Xiaomo Liu and Sameena Shah 184

Transformer-based cynical expression detection in a corpus of Spanish YouTube reviews
Samuel Gonzalez-Lopez and Steven Bethard 194

Multilingual Language Models are not Multicultural: A Case Study in Emotion
Shreya Havaldar, Bhumika Singhal, Sunny Rai, Langchen Liu, Sharath Chandra Guntuku and Lyle Ungar 202

Painsight: An Extendable Opinion Mining Framework for Detecting Pain Points Based on Online Customer Reviews
Yukyung Lee, Jaehee Kim, Doyoon Kim, Yookyung Kho, Younsun Kim and Pilsung Kang . . 215

Context-Dependent Embedding Utterance Representations for Emotion Recognition in Conversations
Patrícia Pereira, Helena Silva Moniz, Isabel Dias and Joao Paulo Carvalho 228

Combining Active Learning and Task Adaptation with BERT for Cost-Effective Annotation of Social Media Datasets
Jens Lemmens and Walter Daelemans 237

Improving Dutch Vaccine Hesitancy Monitoring via Multi-Label Data Augmentation with GPT-3.5
Jens Van Nooten and Walter Daelemans 251

Emotion Analysis of Tweets Banning Education in Afghanistan
Mohammad Ali Hussiny and Lilja Øvrelid 271

Identifying Slurs and Lexical Hate Speech via Light-Weight Dimension Projection in Embedding Space
Sanne Hoeken, Sina Zarrieß and Ozge Alacam 278

Sentiment and Emotion Classification in Low-resource Settings
Jeremy Barnes 290

Analyzing Subjectivity Using a Transformer-Based Regressor Trained on Naïve Speakers' Judgements
Elena Savinova and Fermin Moscoso Del Prado 305

A Fine Line Between Irony and Sincerity: Identifying Bias in Transformer Models for Irony Detection
Aaron Maladry, Els Lefever, Cynthia Van Hee and Veronique Hoste 315

ChatGPT is fun, but it is not funny! Humor is still challenging Large Language Models
Sophie Jentzsch and Kristian Kersting 325

How to Control Sentiment in Text Generation: A Survey of the State-of-the-Art in Sentiment-Control Techniques
Michela Lorandi and Anya Belz 341

Transformer-based Prediction of Emotional Reactions to Online Social Network Posts
Irene Benedetto, Moreno La Quatra, Luca Cagliero, Luca Vassio and Martino Trevisan 354

Transfer Learning for Code-Mixed Data: Do Pretraining Languages Matter?
Kushal Tatariya, Heather Lent and Miryam De Lhoneux 365

Can ChatGPT Understand Causal Language in Science Claims?
Yuheun Kim, Lu Guo, Bei Yu and Yingya Li 379

Systematic Evaluation of GPT-3 for Zero-Shot Personality Estimation
Adithya V Ganesan, Yash Kumar Lal, August Håkan Nilsson and H. Schwartz 390

Utterance Emotion Dynamics in Children's Poems: Emotional Changes Across Age
Daniela Teodorescu, Alona Fyshe and Saif M. Mohammad 401

Annotating and Training for Population Subjective Views
Maria Alexeeva, Caroline Hyland, Keith Alcock, Allegra A. Beal Cohen, Hubert Kanyamahanga, Isaac Kobby Anni and Mihai Surdeanu . . . 416

Exploration of Contrastive Learning Strategies toward more Robust Stance Detection
Udhaya Kumar Rajendran and Amine Trabelsi . . . 431

Adapting Emotion Detection to Analyze Influence Campaigns on Social Media
Ankita Bhaumik, Andy Bernhardt, Gregorios A Katsios, Ning Sa and Tomek Strzalkowski . . 441

Not Just Iconic: Emoji Interpretation is Shaped by Use
Brianna O'Boyle and Gabriel Doyle . . . 452

The Paradox of Multilingual Emotion Detection
Luna De Bruyne . . . 458

Sadness and Anxiety Language in Reddit Messages Before and After Quitting a Job
Molly E. Ireland, Micah Iserman and Kiki Rose Adams . . . 467

Communicating Climate Change: A Comparison Between Tweets and Speeches by German Members of Parliament
Robin Schaefer, Christoph Maximilian Abels, Stephan Lewandowsky and Manfred Stede . . . 479

Modelling Political Aggression on Social Media Platforms
Akash Rawat, Nazia Nafis, Dnyaneshwar Bhadane, Diptesh Kanojia and Rudra Murthy 497

Findings of WASSA 2023 Shared Task on Empathy, Emotion and Personality Detection in Conversation and Reactions to News Articles
Valentin Barriere, João Sedoc, Shabnam Tafreshi and Salvatore Giorgi . . . 511

YNU-HPCC at WASSA-2023 Shared Task 1: Large-scale Language Model with LoRA Fine-Tuning for Empathy Detection and Emotion Classification
Yukun Wang, Jin Wang and Xuejie Zhang . . . 526

AdityaPatkar at WASSA 2023 Empathy, Emotion, and Personality Shared Task: RoBERTa-Based Emotion Classification of Essays, Improving Performance on Imbalanced Data
Aditya Patkar, Suraj Chandrashekhar and Ram Mohan Rao Kadiyala . . . 531

Curtin OCAI at WASSA 2023 Empathy, Emotion and Personality Shared Task: Demographic-Aware Prediction Using Multiple Transformers
Md Rakibul Hasan, Md Zakir Hossain, Tom Gedeon, Susannah Soon and Shafin Rahman . . . 536

Team_Hawk at WASSA 2023 Empathy, Emotion, and Personality Shared Task: Multi-tasking Multi-encoder based transformers for Empathy and Emotion Prediction in Conversations
Addepalli Sai Srinivas, Nabarun Barua and Santanu Pal . . . 542

NCUEE-NLP at WASSA 2023 Shared Task 1: Empathy and Emotion Prediction Using Sentiment-Enhanced RoBERTa Transformers
Tzu-Mi Lin, Jung-Ying Chang and Lung-Hao Lee . . . 548

Domain Transfer for Empathy, Distress, and Personality Prediction
Fabio Gruschka, Allison Lahnala, Charles Welch and Lucie Flek . . . 553

Converge at WASSA 2023 Empathy, Emotion and Personality Shared Task: A Transformer-based Approach for Multi-Label Emotion Classification
Aditya Paranjape, Gaurav Kolhatkar, Yash Patwardhan, Omkar Gokhale and Shweta Dharmadhikari . . . 558

PICT-CLRL at WASSA 2023 Empathy, Emotion and Personality Shared Task: Empathy and Distress Detection using Ensembles of Transformer Models
Tanmay Chavan, Kshitij Deshpande and Sheetal Sonawane 564

Team Bias Busters at WASSA 2023 Empathy, Emotion and Personality Shared Task: Emotion Detection with Generative Pretrained Transformers
Andrew Nedilko and Yi Chu 569

HIT-SCIR at WASSA 2023: Empathy and Emotion Analysis at the Utterance-Level and the Essay-Level
Xin Lu, Zhuojun Li, Yanpeng Tong, Yanyan Zhao and Bing Qin 574

VISU at WASSA 2023 Shared Task: Detecting Emotions in Reaction to News Stories Using Transformers and Stacked Embeddings
Vivek Kumar, Prayag Tiwari and Sushmita Singh 581

Findings of WASSA 2023 Shared Task: Multi-Label and Multi-Class Emotion Classification on Code-Mixed Text Messages
Iqra Ameer, Necva Bölücü, Hua Xu and Ali Al Bataineh 587

Emotion classification on code-mixed text messages via soft prompt tuning
Jinghui Zhang, Dongming Yang, Siyu Bao, Lina Cao and Shunguo Fan 596

PrecogIIITH@WASSA2023: Emotion Detection for Urdu-English Code-mixed Text
Bhaskara Hanuma Vedula, Prashant Kodali, Manish Shrivastava and Ponnurangam Kumaraguru 601

BpHigh at WASSA 2023: Using Contrastive Learning to build Sentence Transformer models for Multi-Class Emotion Classification in Code-mixed Urdu
Bhavish Pahwa 606

YNU-HPCC at WASSA 2023: Using Text-Mixed Data Augmentation for Emotion Classification on Code-Mixed Text Message
Xuqiao Ran, You Zhang, Jin Wang, Dan Xu and Xuejie Zhang 611

Generative Pretrained Transformers for Emotion Detection in a Code-Switching Setting
Andrew Nedilko 616

Program

Friday, July 14, 2023

09:10 - 09:00 *Opening Remarks*

09:10 - 10:25 *Session 1: Sentiment, Emotion, and Irony Recognition*

A Fine Line Between Irony and Sincerity: Identifying Bias in Transformer Models for Irony Detection
Aaron Maladry, Els Lefever, Cynthia Van Hee and Veronique Hoste

Instruction Tuning for Few-Shot Aspect-Based Sentiment Analysis
Siddharth Varia, Shuai Wang, Kishaloy Halder, Robert Vacareanu, Miguel Ballesteros, Yassine Benajiba, Neha Anna John, Rishita Anubhai, Smaranda Muresan and Dan Roth

Unsupervised Domain Adaptation using Lexical Transformations and Label Injection for Twitter Data
Akshat Gupta, Xiaomo Liu and Sameena Shah

Context-Dependent Embedding Utterance Representations for Emotion Recognition in Conversations
Patrícia Pereira, Helena Silva Moniz, Isabel Dias and Joao Paulo Carvalho

Multilingual Language Models are not Multicultural: A Case Study in Emotion
Shreya Havaldar, Bhumika Singhal, Sunny Rai, Langchen Liu, Sharath Chandra Guntuku and Lyle Ungar

10:30 - 11:00 *Coffee Break*

12:30 - 11:00 *Hybrid Poster Session*

12:30 - 13:30 *Invited Talk 1: David Jurgens*

13:30 - 14:30 *Lunch Break*

14:30 - 15:30 *Shared task session*

Findings of WASSA 2023 Shared Task on Empathy, Emotion and Personality Detection in Conversation and Reactions to News Articles
Valentin Barriere, João Sedoc, Shabnam Tafreshi and Salvatore Giorgi

Friday, July 14, 2023 (continued)

HIT-SCIR at WASSA 2023: Empathy and Emotion Analysis at the Utterance-Level and the Essay-Level
Xin Lu, Zhuojun Li, Yanpeng Tong, Yanyan Zhao and Bing Qin

Findings of WASSA 2023 Shared Task: Multi-Label and Multi-Class Emotion Classification on Code-Mixed Text Messages
Iqra Ameer, Necva Bölücü, Hua Xu and Ali Al Bataineh

YNU-HPCC at WASSA 2023: Using Text-Mixed Data Augmentation for Emotion Classification on Code-Mixed Text Message
Xuqiao Ran, You Zhang, Jin Wang, Dan Xu and Xuejie Zhang

15:30 - 16:00 *Coffee Break*

16:00 - 17:00 *Invited Talk 2: Emily Öhman*

17:00 - 18:00 *Session 2: Harmful Speech and Dissatisfaction Recognition*

Identifying Slurs and Lexical Hate Speech via Light-Weight Dimension Projection in Embedding Space
Sanne Hoeken, Sina Zarrieß and Ozge Alacam

Modelling Political Aggression on Social Media Platforms
Akash Rawat, Nazia Nafis, Dnyaneshwar Bhadane, Diptesh Kanojia and Rudra Murthy

Towards Detecting Harmful Agendas in News Articles
Melanie Subbiah, Amrita Bhattacharjee, Yilun Hua, Tharindu Sandaruwan Kumarage, Huan Liu and Kathleen McKeown

Painsight: An Extendable Opinion Mining Framework for Detecting Pain Points Based on Online Customer Reviews
Yukyung Lee, Jaehee Kim, Doyoon Kim, Yookyung Kho, Younsun Kim and Pilsung Kang

18:00 - 18:15 *Closing Remarks and Best Paper Awards*

PESTO: A Post-User Fusion Network for Rumour Detection on Social Media

Erxue Min
Independent Researcher
China
erxue.min@gmail.com

Sophia Ananiadou
National Centre for Text Mining
University of Manchester
United Kingdom
sophia.ananiadou@manchester.ac.uk

Abstract

Rumour detection on social media is an important topic due to the challenges of misinformation propagation and slow verification of misleading information. Most previous work focus on the response posts on social media, ignoring the useful characteristics of involved users and their relations. In this paper, we propose a novel framework, Post-User Fusion Network (PESTO), which models the patterns of rumours from both post diffusion and user social networks. Specifically, we propose a novel Chronologically-masked Transformer architecture to model both temporal sequence and diffusion structure of rumours, and apply a Relational Graph Convolutional Network to model the social relations of involved users, with a fusion network based on self-attention mechanism to incorporate the two aspects. Additionally, two data augmentation techniques are leveraged to improve the robustness and accuracy of our models. Empirical results on four datasets of English tweets show the superiority of the proposed method.

1 Introduction

Rumours, are unverified statements found in social media platforms, which can be damaging if they spread false information with social, economic and political impact (Del Vicario et al., 2016; Zubiaga et al., 2018). For instance: during the period of the U.S. 2016 presidential election, almost 529 different rumours about candidates were propagated on Facebook and Twitter which influenced voters' attitudes (Jin et al., 2017). To this end, it is important for social network platforms to develop effective strategies to combat against fake news and rumours. Recognising misinformation from social media is challenging due to different sources of information required to gather in order to conduct an extensive analysis and reasoning on these sources. Early efforts to tackle rumour detection and misinformation in social network platforms relied on manual verification from users or experts, however, this kind of approach is inefficient due to the substantial human effort and time to recognise a rumour after it has emerged. In recent years, automatic social context based rumour detection has attracted increasing attention. This area of research utilizes the collective wisdom of the social platforms by extracting signals from comments and/or replies towards a source claim (Ma et al., 2016, 2017, 2018; Han et al., 2019; Kochkina et al., 2018; Yuan et al., 2019; Bian et al., 2020; Khoo et al., 2020; Kochkina and Liakata, 2020; Huang et al., 2019). The key idea behind these work is that users from social media would contribute opinions, clues and evidence for distinguishing between false and valid information for rumour detection. Therefore, the content of communication threads and the interaction between posts would be useful for rumour detection. However, apart from the threads of responses, the characteristics of the social network of users can also provide important clues for inferring news veracity. For example, eye-catching rumours usually attract mostly bot accounts to spread, who tend to follow many accounts but with few or no followers (Gilani et al., 2019), such implicit patterns can also support the veracity of a claim. Therefore, in this paper, we aim to propose a method which can model the post diffusion and the user social network jointly to detect social rumours. In terms of post diffusion modeling, a typical line of methods have exploited the characteristics of diffusion structure, such as tree-structured RvNN (Ma et al., 2018), Bi-GCN (Bian et al., 2020) and DSL (Huang et al., 2019), but ignore the temporal information and the implicit connections between posts. Sequence-based models such Recurrent neural networks (RNNs) (Ma et al., 2016), PLAN (Khoo et al., 2020) and DCM (Veyseh et al., 2019) flatten the tree structure and arrange posts in chronological order. They overcome some limitations of tree models but underexploit the diffusion

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 1–10
July 14, 2023 ©2023 Association for Computational Linguistics

structure. For this sake, in the paper, we propose a Chronologically-masked Transformer architecture, which integrates both temporal and structural information to effectively model the rumour diffusion patterns. In terms of user network modeling, many off-the-shelf graph neural networks such as Graph Convolutional Network (GCN) (Kipf and Welling, 2016), GraphSAGE (Hamilton et al., 2017), Graph Attention Network (GAT) (Velivcković et al., 2017), Relational Graph Convolutional Network (RGCN) (Schlichtkrull et al., 2018) can be leveraged. Considering that A-follow-B and A-followed-by-B are different relations, we adopt RGCN for user network representation. In order to fuse the information in two aspects, we propose to use a self-attention layer for final information aggregation. Since many existing rumour detection datasets are in small scale, we propose two data augmentation techniques: Connection dropping and Sub-conversation training to assist model training. We name the entire architecture as Post-User Fusion Network (PESTO). Our experimental evaluation shows PESTO improves performance over previous approaches. The contributions of our work are as follows:

- We propose a Chronologically-masked Transformer architecture to model the post diffusion patterns of rumours, with both temporal and structural information considered.
- We leverage a Relational Graph Convolutional Network to represent the user social network, and integrate it with the chronologically-masked Transformer via a Fusion network based on self-attention.
- We adopt two data augmentation techniques: Connection dropping and Sub-conversation training, to reduce overfitting, making our model more robust and stable.

2 Related Work

Existing detection approaches of fake claims can be generally categories into three groups based on the information utilized: (i) the content of the claim, (ii) knowledge from trustworthy sources and (iii) social response to the claim. Our work in this paper falls into the last group, which exploits social replies and the involved user network to detect rumours. In this section, we briefly introduce each group of work.

Content-based Detection: This line of studies studied specific linguistic cues such as verb quantity, word classes, word length, pronouns, non-objectivity (Rubin and Lukoianova, 2015; Feng et al., 2012; Potthast et al., 2017). These features are useful to detect satires or onion news, but might be unique to domains or topics.

Knowledge-based Detection: Fact checking websites such as politifact.com and snope.com leverage manual verification to debunk fake news or rumours, but fail to match the rapid emergence rate of misinformation nowadays. Automated fact checking techniques rely on truthworthy sources such as Wikipedia, but they might not work for latest news without evidences.

Social Response-based Detection Social response information such as reply contents and propagation structures have been shown to be particularly useful for classifying rumours. Ma et al. (Ma et al., 2017) uses tree kernel to capture the similarity of propagation trees by counting their similar sub-structures in order to identify different types of rumours on Twitter. Ma et al. (Ma et al., 2018) make use of tree-structured recursive neural network to model the propagation tree, and information from different nodes is aggregated recursively in either a bottom-up or a top-down manner. Bian et al. (Bian et al., 2020) also propose a bi-directional graph model named Bi-GCN to explore both propagation and aggregation patterns by operating on both top-down and bottom-up propagation of rumours. However, the focus in these works is on using the static tree structure of Tweet propagation, ignoring the temporal order and implicit connections between posts. For this sake, Veyseh et al. (Veyseh et al., 2019) and Khoo et al. (Khoo et al., 2020) propose to apply self-attention mechanism (Vaswani et al., 2017) to model implicit connections, but their direct usage of self-attention does not consider the propagation and aggregation characteristic of news conversation and underexploit the explicit diffusion structure. All of previous work do not take user networks into consideration, which provides important evidences for detection (Yang et al., 2019; Shu et al., 2019).

3 Preliminaries

3.1 Problem Statement

We define rumour detection as predicting the label (e.g., Rumour or Non-rumour) of a source post on social media, given all its responding posts and the response relations between them. A rumour detection dataset is a set of threads: $\mathbf{T} = \{T_1, T_2, ...T_{|\mathbf{T}|}\}$, where $T_i = \{p^i_1, p^i_2, ...p^i_{M_i}, u^i_1, u^i_2, ...u^i_{N_i}, G^P_i, G^U_i, G^{UP}_i\}$ is the i-th event, where M_i and N_i denotes the number of posts and involved users in T_i respectively, p^i_j denotes the j-th post and u^i_k denotes the k-th user. p^i_1 is the source post and others are corresponding retweeted posts or responsive posts in chronological order. G^P_i is the propagation structure of posts. Specifically, G^P_i is defined as a graph $\langle V^P_i, E^P_i \rangle$, where $V^P_i = \{p^i_1, p^i_2, ..., p^i_{M_i}\}$, and $E^P_i = \{e^P_{i(st)} | s, t = 1, ..., M_i\}$ that represents the set of edges from responsive posts to responded posts. Likewise, G^U_i is defined as a graph $\langle V^U_i, E^U_i \rangle$, where $V^U_i = \{u^i_1, u^i_2, ...u^i_{N_i}\}$. and $E^U_i = \{e^U_{i(st)} | s, t = 1, 2, ..., N_i\}$ represents the set of edges from users to the users they follow. $G^{UP}_i = \{V^U_i \cup V^P_i, E^{UP}_i\}$ is the user-publish-post graph, where $E^{UP}_i = \{e^{UP}_{i(st)} | s = 1, ..., N_i, t = 1, ..., M_i\}$ denotes the set of edges from users to the posts they published. Each event T_i is associated with a ground-truth label $y_i \in \{F, T\}$ (i.e., False Rumour or True Rumour). In certain cases, the dataset contains four fine-grained class $\{N, F, T, U\}$ (i.e., Non-rumour, False Rumour, True Rumour and Unverified Rumour). We formulate this task as a supervised classification problem, which aims at learning a classifier f from labeled events, that is $f : T_i \rightarrow y_i$.

3.2 Architecture of Transformer

The Transformer model (Vaswani et al., 2017) employs an encoder-decoder architecture, consisting of stacked encoder and decoder layers. Each encoder layer consists of two sub-layers: a self-attention layer and a position-wise feed-forward network. The self-attention layer employs h attention heads. Each attention head operates on the same input sequence $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_n)$ of n elements where $\mathbf{x}_i \in \mathbb{R}^d$, and computes a new sequence $\mathbf{Z} = (\mathbf{z}_1, ..., \mathbf{z}_n)$ of the same length where $\mathbf{z}_i \in \mathbb{R}^{d_k}$. To be specific, each $\mathbf{x}_i$ is firstly linearly transformed into a query vector, a key vector and a value vector:

$$\mathbf{q}_i = \mathbf{W}^Q \mathbf{x}_i, \mathbf{k}_i = \mathbf{W}^K \mathbf{x}_i, \mathbf{v}_i = \mathbf{W}^V \mathbf{x}_i, \quad (1)$$

where $\mathbf{W}^K, \mathbf{W}^Q, \mathbf{W}^V \in \mathbb{R}^{d_k \times d}$ are layer-specific trainable parameter matrices. Then, each element $\mathbf{z}_i$ is computed as the weighted sum of $\mathbf{v}_j$:

$$\mathbf{z}_i = \sum_{j=1}^{n} \frac{\exp(e_{ij})}{\sum_{k=1}^{n} \exp(e_{ik})} \mathbf{v}_j \quad (2)$$

and e_{ij} is the unnormalized attention score computed via a compatibility function, e.g., Scaled dot product, that compares $\mathbf{q}_i$ and $\mathbf{k}_i$, using:

$$e_{ij} = \frac{\mathbf{q}_i^{\mathrm{T}} \mathbf{k}_j}{\sqrt{d_k}}. \quad (3)$$

Note that all these parameter matrices, $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V$, are unique for each attention head. Then, the outputs of all the attention heads are concatenated. Finally, the concatenated vector is fed to a parameterized linear transformation to obtain the output of the self-attention sublayer:

$$\hat{\mathbf{z}}_i = \mathbf{W}^O \, \mathrm{Concat}(\mathbf{z}^1_i, ..., \mathbf{z}^h_i). \quad (4)$$

Finally, a position-wise feed-forward network is used to produce the output node embeddings $\tilde{\mathbf{z}}_i$:

$$\tilde{\mathbf{z}}_i = \mathrm{FFN}(\hat{z}_i) = \mathbf{W}_2 \sigma(\mathbf{W}_1 \hat{z}_i + \mathbf{b}_1) + \mathbf{b}_2, \quad (5)$$

where, $\mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_1, \mathbf{b}_2$ are parameters, σ is the non-linear function.

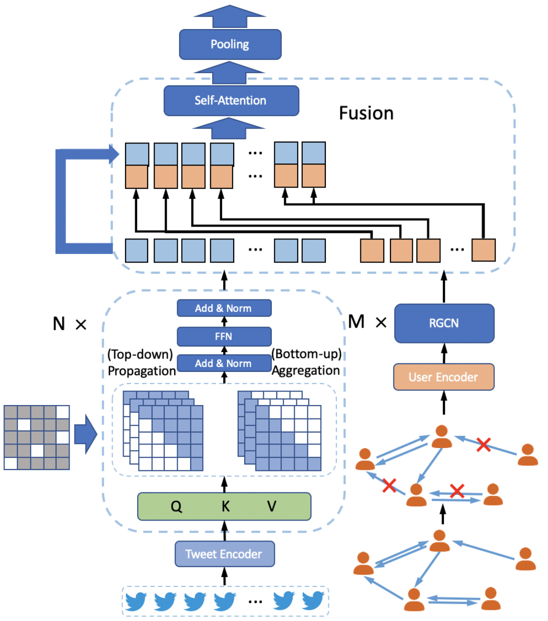


Figure 1: The architecture of PESTO. The left bottom part is the Chronologically-masked Transformer architecture, the right bottom part is the user network modeling architecture, with connection dropping mechanism applied to both parts. The upper part of the architecture is the fusion network for aggregation of the two views.

4 Methodology

4.1 Overview of Model Architecture

In this section, we introduce our proposed Post-User Fusion Network (PESTO). The core idea of

Table 1: Detailed meta features of post and user nodes

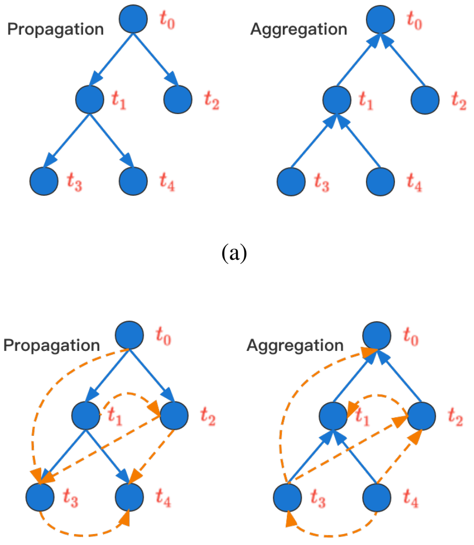
Type	Feature name	Example
Post	Post type	0/1/2*
	Retweet Count	10
	Reply Count	10
	Like Count	10
	Quote Count	10
	Created time	1501143981
	Sentiment Score	0.8
User	is_verified	1
	Following Count	100
	Followers Count	1000
	Tweet Count	1000
	List Count	10
	Account created time	1458483921
	Description length	20

* 0 denotes tweet, 1 denotes retweet, 2 denotes reply.

PESTO is to learn discriminative representations for both post propagation tree and the user social network respectively, and then fuse them based on self-attention mechanism. The overall architecture of the proposed model is illustrated in Figure 1. Our model consists of four major parts: 1) Posts/User Feature Encoder, which encodes the text and meta features of a post/user into a dense vector. 2) Chronologically-masked Transformer, which learns the representation of the post tree. 3) Relational Graph Convolutional Network, which learns the representations of the user-follow network. 4) Fusion Network based on Self-Attention, which learns the global representation of post-user pairs.

4.2 Post/User Feature Encoder

Each post/user node contains two types of features: text features which are short sequences of words $\mathbf{x}$ and meta features $\mathbf{m}$ (e.g., follower count, following count, retweet count, etc, details are listed in table 1). For each post, the text features are the post content, which contains distinctive patterns such as exaggerated expressions or negative stance, and for each user, the text features are the user description, which contains some bot-like flags or political stance that implies the credibility of users. We use the same encoder architecture to represent both post and user nodes. There are many methods to represent texts in rumour detection, such as TF-IDF (Aizawa, 2003), Convolutional Neural Network (CNN) (Kalchbrenner et al., 2014), LSTM (Hochreiter and Schmidhuber, 1997), Transformer (Vaswani et al., 2017) and BERT (Wolf et al., 2019). In our work, we apply word embeddings with CNN as our textual feature extractor, which shows the best performance and efficiency in our experiments. Specifically, we first embed each word in the text into a $k-$dimensional dense semantic representation using public pre-trained word vector Glove (Pennington et al., 2014). Then, a convolutional layer with window sizes of $2, 3, 4$ is applied, followed by a max-pooling layer to obtain the final text representation $\mathbf{h}_x$. After that, we concatenate $\mathbf{h}_x$ and $\mathbf{m}$ and use a linear layer to obtain the final representation of the node. For event T_i, we obtain the feature representation of all posts $\mathbf{P}^i = \{\mathbf{p}_1^i, \mathbf{p}_2^i, ...\mathbf{p}_{M_i}^i\}$, and representation of all users $\mathbf{U}^i = \{\mathbf{u}_1^i, \mathbf{u}_2^i, ..., \mathbf{u}_{N_i}^i\}$. We discard the superscript i in the following sections for simplicity.



(a)

(b)

Figure 2: Illustration of diffusion trees. The blue lines denote responsive relations, and the orange lines denotes implicit relations

4.3 Chronologically-masked Transformer for Representation of Post Diffusion Tree

Many post tree modeling methods such as tree-structured RvNN (Ma et al., 2018), Bi-GCN (Bian et al., 2020) and DSL (Huang et al., 2019) attempt to learn the representation of post diffusion tree from two directions: Top-down (Propagation) and Bottom-up (Aggregation) as illustrated in Figure 2(a), to capture structural and semantic features. However, as illustrated in Figure 2(b), each user is often able to observe and respond to all existing posts at the time of writing a post in the conversation, while this lines of methods ignore the implicit interactions between unconnected posts, as well as the important temporal order. Therefore, we propose a Chronological-masked Transformer to model both temporal and structural characteristics of post diffusion. Specifically, we propose a chronologically-masked self-attention mechanism, which models the Top-down information spreading and Bottom-up aggregation separately in each layer based on the chronological order, and involves the diffusion tree structure into attention calculation via extra learnable position parameters. In the multi-head self-attention layers of standard Transformer (Vaswani et al., 2017), the state in i-th position can attend to any other position in the whole sequence, here we propose to adopt a chronologically-masking mechanism to inject the structure of both propagation and aggregation into multi-head self-attention mechanism (Vaswani et al., 2017). Specifically, As illustrated in the left bottom part of Figure 1, we first divide the heads in each self-attention layer into two groups: *propagation* heads and *aggregation* heads. For *propagation* heads, we restrict the head representation to only aggregate information from all position j with $(j \leqslant i)$ when calculating the output embedding at position i. Likewise, for *aggregation* heads, we mask the attention score from position j with $(j < i)$ for position i. The weighted sum of values at positions i for *propagation* heads and *aggregation* heads are computed as:

$$\mathbf{z}_i^p = \sum_{j=i}^{M_i} \frac{\exp(e_{ij})}{\sum_{k=i}^{M_i} \exp(e_{ik})} \mathbf{v}_j, \tag{6}$$

$$\mathbf{z}_i^a = \sum_{j=0}^{i} \frac{\exp(e_{ij})}{\sum_{k=0}^{j} \exp(e_{ik})} \mathbf{v}_j \tag{7}$$

, Furthermore, since the masking mechanism only utilizes the chronological information, in order to involve explicit spreading structure (i.e., the tree structure), we modify the calculation of attention score in Equation 3 to a structure-aware version as follows:

$$e_{ij} = \frac{\mathbf{q}_i^{\mathrm{T}} \mathbf{k}_j + \alpha_{\phi(i,j)}}{\sqrt{d_k}}, \tag{8}$$

where $\alpha_{\phi(i,j)}$ is a learnable scalar indexed by $\phi(i,j)$, and shared across all layers. $\phi(i,j)$ is the relative position between post i and post j:

$$\phi(i,j) = \begin{cases} d_i - d_j & p_i \text{ is the parent of } p_j \\ d_j - d_i + d_{\max} & p_i \text{ is the child of } p_j \\ 0 & i = j \\ 2d_{\max} & \text{in different branches} \end{cases} \tag{9}$$

, where d_i denotes the depth of post i in the spreading tree and d_{max} is the maximum depth. Through the learnable position parameters, the attention score can capture the meaningful structural information between post i and post j.

The final representation at position i before the FFN layer is the concatenation of all head presentation, denoted as:

$$\hat{\mathbf{z}}_i = \mathbf{W}^O \mathrm{Concat}(\mathbf{z}_{i,1}^p, ..., \mathbf{z}_{i,n_p}^p, \mathbf{z}_{i,1}^a, ..., \mathbf{z}_{i,n_a}^a) \tag{10}$$

, where n_p,n_a denote the number of propagation heads and aggregation heads, $\mathbf{W}^O$ is trainable parameters. Given input feature matrix of all posts $\mathbf{P}$, we obtain $\hat{\mathbf{P}} = \{\hat{\mathbf{p}}_1, \hat{\mathbf{p}}_2, ...\hat{\mathbf{p}}_M\}$ after the representation of the Chronologically-masked Transformer Network.

4.4 User Network Representation

We introduce our representation module for user social network in this section. Given the representation of all users $\mathbf{U} = \{\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_N\}$ and the adjacent matrix $\mathbf{A}^U$ of user-follow relation set E^U, we attempt to learn a structure-aware representation for each node in the following graph. Since the followers and followings describe two separate aspects of a user's characteristics, we consider neighbours of the two categories separately. Specifically, we generate the user-followed a adjacent matrix $\mathbf{A}^{U\prime} = \mathbf{A}^{U\top}$.

We also generate the symmetric friendship adjacent matrix $\mathbf{A}^{U\prime\prime} = \mathbf{A}^U \cdot \mathbf{A}^{U\prime}$. Given the three adjacent matrices and node features, we adopt RGCN (Schlichtkrull et al., 2018) to represent the graph. The feature update equation can be formulated as follows:

$$\mathbf{h}_i^{(t+1)} = \sigma\Big(\sum_{r\in\mathcal{R}} \sum_{j\in\mathcal{N}_i^r} \frac{1}{|\mathcal{N}_i^r|} \mathbf{W}_r^{(t)} \mathbf{h}_j^{(t)} + \mathbf{W}_{\text{root}}^{(t)} \mathbf{h}_i^{(t)}\Big) \tag{11}$$

where $\mathcal{N}_i^r$ denotes the set of neighbor indices of node i under relation $r \in R$, with corresponding adjacent matrix $\mathbf{A}^r \in \{\mathbf{A}^U, \mathbf{A}^{U\prime}, \mathbf{A}^{U\prime\prime}\}$, $\mathbf{W}_r^{(t)}$ is the parameter matrix of relation r in layer t, $\mathbf{W}_{\text{root}}^{(t)}$ is the parameter matrix of target node. After the transformation of multiple RGCN layers, we obtain the structure-aware representation of all users: $\hat{\mathbf{U}} = \{\hat{\mathbf{u}}_1, \hat{\mathbf{u}}_2, ...\hat{\mathbf{u}}_N\}$.

4.5 Post-User Fusion Network

Once we have obtain the representation of posts and users denoted as $\hat{\mathbf{P}}_i$ and $\hat{\mathbf{U}}_i$ for event T_i, we fuse the information of posts and users via a fusion network. According to user-publish-post graph G_i^{UP}, We first concatenate the hidden vectors of m-th post and n-th user, if n-th user is the author of m-th post. Note that a user can write multiple posts but a post only has one author. Therefore, we obtain the fused representation matrix $H_i = \{\mathbf{h}_1^i, \mathbf{h}_2^i, ..., \mathbf{h}_{M_i}^i\}$, where $\mathbf{h}_j^i = \text{Concat}(\hat{\mathbf{p}}_j^i, \hat{\mathbf{u}}_{u(j)}^i)$, where $u(j)$ denotes the index of user who is the author of j-th post. In order to capture the semantic relations between the fused post-user pairs, we further use a self-attention layer to obtain the final representation of all post-user pairs, denoted by $\hat{H} = \{\hat{\mathbf{h}}_1, \hat{\mathbf{h}}_2, ..., \hat{\mathbf{h}}_M\}$. Afterwards, a mean pooling layer is applied to obtain the aggregated representation $\mathbf{c}$, followed by several fully-connected layers and a Softmax layer to get the vector of probabilities for all classes. We train all the parameters in the Network by minizing the cross-entropy of the prediction and ground truth labels over the entire dataset T.

4.6 Data Augmentation Mechanism

Since existing datasets for rumour detection are mostly in small scale, overfitting is a serious issue in this domain. For this sake, we use two data augmentation mechanism to mitigate this problem: Connection dropping and Sub-conversation training.

- **Connection dropping:** We adopt two versions of Connection dropping operation for the user graph and post graph. For user graph, we apply the same strategy as (Bian et al., 2020): supposing the total number of edges in the user following graph A^U is N_U and the dropping rate is p_u, then the adjacency matrix with edge dropping is $\widetilde{A}^U = A^U - A^U_{\text{drop}}$, where A^U_{drop} is the matrix constructed using $N_U \times p_u$ edges randomly sampled from A^U. The edge dropping operation is performed before input A^U into each RGCN layer, and the $A^{U\prime}, A^{U\prime\prime}$ are calculated based on $\widetilde{A}^U$. For post spreading tree, since we learn all implicit correlation between posts using self-attention, we propose to use an attention dropping mechanism, which randomly set the attention score before Softmax as $-inf$ with rate p_p.
- **Sub-conversation training:** In order to improve the robustness and early-detection capability of our model, we adopt a sub-conversation training technique. To be specific, we randomly set a time threshold t_{early}, with $t_{min} < t_{early} < t_{last}$ for each event during training, where t_{min} is the minimum detection time and t_{last} is the time of the last tweet in the event. The posts after the time is removed, so does the corresponding users. This technique enables models to learn invariant features during the whole life cycle of a event.

Table 2: Statistics of the datasets

Statistic	Twitter15	Twitter16	PolitiFact	GossipCop
# of posts	331,612	204,820	130872	880640
# of user	276,663	173,487	89238	568482
# of events	1490	818	574	6880
# of True rumors	374	205	\	\
# of False rumors	370	205	231	2313
# of Unverified rumors	374	203	\	\
# of Non-rumors	372	205	343	4567
Avg. # of posts / event	223	251	228	128
Max # of posts / event	1,768	2,765	3294	1038
Min # of posts / event	55	81	32	12

5 Experimental Results

In this section, we first compare the performance of our proposed PESTO method with several baseline models. Then, ablation studies are conducted to illustrate the impacts of each module. Afterwards, early detection performance is evaluated. Empirical results show the superiority of the proposed method.

5.1 Datasets and Baselines

We evaluate our proposed method on four publicly available Twitter datasets: Twitter15 and Twitter16 (Ma et al., 2017), PolitiFact and GossipCop (Shu et al., 2020). The statistics are listed in Table 2. Since in the original datasets, each instance only contains the tweet propagation tree, we use Twitter academic API[1] to search the corresponding user of each tweet and the following relations between users. Each source tweet is annotated with one of the four class labels, i.e., Non-rumour (N), False rumor (F), True rumor (T), and Unverified rumor (U). We compare our method with several baselines:

- DTC (Castillo et al., 2011): A Decision Tree classifier based on various handcrafted features to obtain information credibility.

---

[1] https://developer.twitter.com/en/products/twitter-api/academic-research

- SVM-TS (Ma et al., 2017): A linear SVM classifier that utilizes handcraft features to construct time-series model.
- SVM-TK (Ma et al., 2017): A SVM classifier with a tree kernel based on the propagation structure of rumours.
- RvNN (Ma et al., 2018): A tree-structured recursive neural network with GRU units that learn the propagation structure
- PPC_RNN+CNN (Liu and Wu, 2018): A model combining RNN and CNN, which learns the rumour representations through the characteristics of users in the rumour propagation path.
- Bi-GCN (Bian et al., 2020): A GCN-based rumour detection model using bi-directional propagation structure.
- DCM (Veyseh et al., 2019): A rumour detection model based on post-level self-attention mechanisom.
- PESTO-U: A variant of PESTO, with the user network modeling part removed.
- PESTO: Our proposed PESTO, with all modules included.

5.2 Experimental Setup

In all experiments, we used the Glove 100d embeddings (Pennington et al., 2014) to represent each token in a tweet or user profile because they are trained using tweet corpus. For the chronologically-masked Transformer, the hidden size is 128, the layer number is 4, the head number is 8. For the RGCN Network, the layer number is 2, the hidden size is 128. The dropout rate of both networks is 0.2, and the edge dropping rate is also 0.2. We use the Adam optimizer with 6000 warm start-up steps. For all datasets, we evaluate the Accuracy (ACC) over all categories and F1 measure (F1) on each class.

5.3 Overall Performance

Table 3 shows the performance of the proposed method and all the baselines on Twitter15 and Twitter16, respectively. First, it is apparent that all the deep learning methods outperform those using handcrafted features significantly, showing that deep neural networks are able to learn better representations of rumours. Second, the proposed method and its variants outperform other deep learning methods in terms of all metrics, which indicates the superiority of PESTO. As for RvNN, it only uses the hidden feature vector of all the leaf nodes, which implies that it is heavily influenced by the information of latest posts. As for Bi-GCN, it only relies on the explicit responsive path, ignoring the implicit relations between posts. As for DCM, it simply use the self-attention layer without modification, ignoring the propagation and aggregation characteristics of rumours. PESTO-U outperforms previous methods, demonstrating the effectiveness of the proposed chronologically-masked self-attention architecture. PESTO has better performance compared with PESTO-U, indicating the user following network contains valuable information for detection.

Table 3: Overall results on Twitter15 and Twitter16

Twitter15

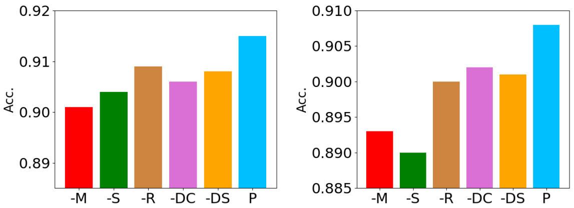
Method	ACC	N	F	T	U
DTC	0.779	0.415	0.355	0.733	0.317
SVM-TS	0.544	0.796	0.472	0.404	0.483
SVM-TK	0.750	0.804	0.698	0.765	0.733
RvNN	0.723	0.682	0.758	0.821	0.654
PPC RNN+CNN	0.477	0.359	0.507	0.300	0.640
Bi-GCN	0.886	0.891	0.860	0.930	0.864
DCM	0.770	0.814	0.764	0.775	0.743
PESTO-U	0.895	0.897	0.896	0.888	0.900
PESTO	**0.915**	**0.912**	**0.922**	**0.921**	**0.904**

Twitter16

Method	ACC	N	F	T	U
DTC	0.473	0.254	0.080	0.190	0.482
SVM-TS	0.574	0.755	0.420	0.571	0.526
SVM-TK	0.732	0.740	0.709	0.836	0.686
RvNN	0.737	0.662	0.743	0.835	0.708
PPC RNN+CNN	0.564	0.591	0.543	0.394	0.674
Bi-GCN	0.880	0.847	0.869	0.937	0.865
DCM	0.768	0.825	0.751	0.768	0.789
PESTO-U	0.891	0.906	0.891	0.890	0.875
PESTO	**0.908**	**0.902**	**0.914**	**0.915**	**0.901**

Table 4: Overall results on PolitiFact and GossipCop

Dataset	PolitiFact		GossipCop	
Method	ACC	F1	ACC	F1
DTC	0.753	0.749	0.772	0.769
SVM-TS	0.757	0.759	0.789	0.783
SVM-TK	0.731	0.721	0.753	0.745
RvNN	0.790	0.778	0.798	0.796
PPC RNN+CNN	0.744	0.760	0.776	0.776
Bi-GCN	0.821	0.819	0.811	0.802
DCM	0.812	0.810	0.810	0.809
PESTO-U	0.832	0.821	0.821	0.816
PESTO	**0.845**	**0.836**	**0.834**	**0.831**

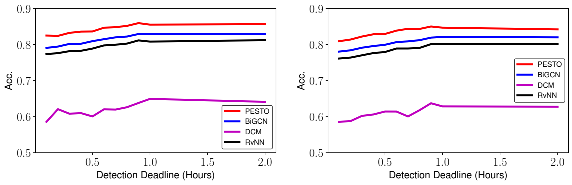


(a) Twitter15 (b) Twitter16

Figure 3: The performance of the PESTO and its variants. **-M**,**-S**,**-R**,**-DC**,**-DS** are abbreviations of different variants, which are elaborated in Section 5.4

5.4 Ablation study

To demonstrate the effectiveness of each module of PESTO, we conduct ablation analysis on Twitter15 and Twitter16 in this section. We compare **PESTO** with its variants -**M**,-**S**,-**R**,-**DC**,-**DS** which represent our model (1) without chronological **M**asking for post Transformer,(2) without **S**tructure-aware attention for post Transformer,(3) with **R**GCN replaced by GCN,(4) without **C**onnection dropping and (5) Without **S**ub-conversation training. As illustrated in Figure 3, each parts contribute to PESTO. The impacts of **M** and **S** show that involving intrinsic characteristic of the spreading tree improves the performance. RGCN is better than GCN for user network modeling, indicating that treating user-following network as directed graph retrains more valuable information. The contribution of **DC** and **DS** shows the importance of robust training.



(a) Twitter15 (b) Twitter16

Figure 4: The performance of early detection.

5.5 Early Rumour Detection

Detecting rumours at the early stage of propagation is crucial to reduce the negative effects of rumours. For the early detection task, we select a series of detection deadlines and only utilize the posts released before the deadlines and the corresponding induced user network to evaluate the performance in terms of accuracy. Figure 4 shows the performances of RvNN, Bi-GCN, DCM and our PESTO model at various deadlines on Twitter15 and Twitter16 datasets. We can find that the performance of PESTO is stably superior to other models.

6 Conclusion

In this paper, we address the task of rumour detection with social contexts. A novel Post-User Fusion Network (PESTO) is proposed to learn both post propagation patterns and user network patterns in a rumour event. To be specific, we model the post diffusion patterns using a novel chronologically-masked Transformer, and use RGCN to represent the user social network, then a fusion module based on self-attention is applied to integrate the two aspects. Experiments show that PESTO outperforms state-of-the-art baselines significantly.

7 Acknowledgement

Erxue Min acknowledges the support from The University of Manchester - China Scholarship Council joint scholarship.

References

Akiko Aizawa. 2003. An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, 39(1):45–65.

Tian Bian, Xi Xiao, Tingyang Xu, Peilin Zhao, Wenbing Huang, Yu Rong, and Junzhou Huang. 2020. Rumor detection on social media with bi-directional graph convolutional networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 549–556.

Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. 2011. Information credibility on twitter. In *Proceedings of the 20th international conference on World wide web*, pages 675–684.

Michela Del Vicario, Alessandro Bessi, Fabiana Zollo, Fabio Petroni, Antonio Scala, Guido Caldarelli, H Eugene Stanley, and Walter Quattrociocchi. 2016. The spreading of misinformation online. *Proceedings of the National Academy of Sciences*, 113(3):554–559.

Song Feng, Ritwik Banerjee, and Yejin Choi. 2012. Syntactic stylometry for deception detection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 171–175.

Zafar Gilani, Reza Farahbakhsh, Gareth Tyson, and Jon Crowcroft. 2019. A large-scale behavioural analysis of bots and humans on twitter. *ACM Transactions on the Web (TWEB)*, 13(1):1–23.

William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *arXiv preprint arXiv:1706.02216*.

Sooji Han, Jie Gao, and Fabio Ciravegna. 2019. Data augmentation for rumor detection using context-sensitive neural language model with large-scale credibility corpus.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Qi Huang, Chuan Zhou, Jia Wu, Mingwen Wang, and Bin Wang. 2019. Deep structure learning for rumor detection on twitter. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Zhiwei Jin, Juan Cao, Han Guo, Yongdong Zhang, Yu Wang, and Jiebo Luo. 2017. Detection and analysis of 2016 us presidential election related rumors on twitter. In *International conference on social computing, behavioral-cultural modeling and prediction and behavior representation in modeling and simulation*, pages 14–24. Springer.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.

Ling Min Serena Khoo, Hai Leong Chieu, Zhong Qian, and Jing Jiang. 2020. Interpretable rumor detection in microblogs by attending to user interactions. *arXiv preprint arXiv:2001.10667*.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Elena Kochkina and Maria Liakata. 2020. Estimating predictive uncertainty for rumour verification models. *arXiv preprint arXiv:2005.07174*.

Elena Kochkina, Maria Liakata, and Arkaitz Zubiaga. 2018. All-in-one: Multi-task learning for rumour verification. *arXiv preprint arXiv:1806.03713*.

Yang Liu and Yi-Fang Brook Wu. 2018. Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In *Thirty-second AAAI conference on artificial intelligence*.

Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J Jansen, Kam-Fai Wong, and Meeyoung Cha. 2016. Detecting rumors from microblogs with recurrent neural networks.

Jing Ma, Wei Gao, and Kam-Fai Wong. 2017. Detect rumors in microblog posts using propagation structure via kernel learning. Association for Computational Linguistics.

Jing Ma, Wei Gao, and Kam-Fai Wong. 2018. Rumor detection on twitter with tree-structured recursive neural networks. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Martin Potthast, Johannes Kiesel, Kevin Reinartz, Janek Bevendorff, and Benno Stein. 2017. A stylometric inquiry into hyperpartisan and fake news. *arXiv preprint arXiv:1702.05638*.

Victoria L Rubin and Tatiana Lukoianova. 2015. Truth and deception at the rhetorical structure level. *Journal of the Association for Information Science and Technology*, 66(5):905–917.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer.

Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan Liu. 2020. Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. *Big data*, 8(3):171–188.

Kai Shu, Suhang Wang, and Huan Liu. 2019. Beyond news contents: The role of social context for fake news detection. In *Proceedings of the twelfth ACM international conference on web search and data mining*, pages 312–320.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Petar Velivcković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Amir Pouran Ben Veyseh, My T Thai, Thien Huu Nguyen, and Dejing Dou. 2019. Rumor detection in social networks via deep contextual modeling. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 113–120.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Shuo Yang, Kai Shu, Suhang Wang, Renjie Gu, Fan Wu, and Huan Liu. 2019. Unsupervised fake news detection on social media: A generative approach. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 5644–5651.

Chunyuan Yuan, Qianwen Ma, Wei Zhou, Jizhong Han, and Songlin Hu. 2019. Jointly embedding the local and global relations of heterogeneous graph for rumor detection. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 796–805. IEEE.

Arkaitz Zubiaga, Ahmet Aker, Kalina Bontcheva, Maria Liakata, and Rob Procter. 2018. Detection and resolution of rumours in social media: A survey. *ACM Computing Surveys (CSUR)*, 51(2):1–36.

Sentimental Matters
Predicting Literary Quality with Sentiment Analysis and Stylistic Features

Yuri Bizzoni
Aarhus University, Denmark
`yuri.bizzoni@cc.au.dk`

Pascale Feldkamp Moreira
Aarhus University, Denmark

Mads Rosendahl Thomsen
Aarhus University, Denmark
`madsrt@cc.au.dk`

Kristoffer L. Nielbo
Aarhus University, Denmark
`kln@cas.au.dk`

Abstract

The task of predicting reader appreciation or literary quality has been the object of several studies. It remains, however, a challenging problem in quantitative literary analyses and computational linguistics alike, as its definition can vary a lot depending on the genre of literary texts considered, the features adopted, and the annotation system employed. This paper attempts to evaluate the impact on reader appreciation, defined as online users' ratings, of sentiment range and sentiment arc patterns versus traditional stylometric features. We run our experiments on a corpus of English-language literary fiction, showing that stylometric features alone are helpful in modelling literary quality, but can be outperformed by analysing the novels' sentimental profile.

1 Introduction

The question of what literary quality "is" is as complex as it is old. It may be argued that "literary quality" is an empty concept, since individual tastes of narrative and literature can differ widely among readers. Yet it is possible that a set of textual and narrative characteristics tend to improve or damage the appreciation of a literary piece independently from genre expectations and preferences. This persistent intuition, while controversial, has been amply discussed through the history of literary criticism, and also stands at the foundation of most rhetorical or writing advice. The idea of an intersubjective agreement on literary quality may be also sustained by the convergence of large numbers of readers (and when considering canons, generations of readers) on certain titles rather than others (Koolen et al., 2020a; Walsh and Antoniak, 2021b). In the quest of defining principles of literary quality, quantitative analyses ask two questions: whether it is possible to define literary quality at all; and whether it is possible to individuate textual patterns that contribute to make a text more appreciated. In this paper we aim to explore the interplay of the sentiment and stylometric characteristics of narrative texts and their role in the perception of literary quality.

2 Related works

Traditionally, quantitative studies of literary quality have relied on texts' stylometric properties, ranging from the percentage of adverbs (Koolen et al., 2020b) to the count of the most frequent n-grams in a text (van Cranenburgh and Koolen, 2020), to model the success or quality of literary works. More recent works, nonetheless, have emphasized the potential of sentiment analysis (Alm, 2008; Jain et al., 2017), at the word (Mohammad, 2018), sentence (Mäntylä et al., 2018) or paragraph (Li et al., 2019) level, to uncover meaningful mechanisms in the reading experience (Drobot, 2013; Cambria et al., 2017; Kim and Klinger, 2018; Brooke et al., 2015; Jockers, 2017), usually by drawing scores from human annotations (Mohammad and Turney, 2013) or induced lexica (Islam et al., 2020).While most studies have focused on the valence of sentiment arcs, Hu et al. (2021) and Bizzoni et al. (2022a) have tried to model the persistence, coherence, and predictability of novels' sentiment arcs, using fractal analysis (Mandelbrot and Ness, 1968; Mandelbrot, 1982, 1997; Beran, 1994; Eke et al., 2002; Kuznetsov et al., 2013), a method of studying patterns in complex systems, exploring the degree of predictability or self-similarity of narratives – a method that appears to capture meaningful patterns impacting reading experience. Naturally, beyond which features to consider, another great challenge of studying literary quality is that of finding "oracles" of quality. Measures of quality have been approximated by looking at readers' ratings on plat-

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 11–18
July 14, 2023 ©2023 Association for Computational Linguistics

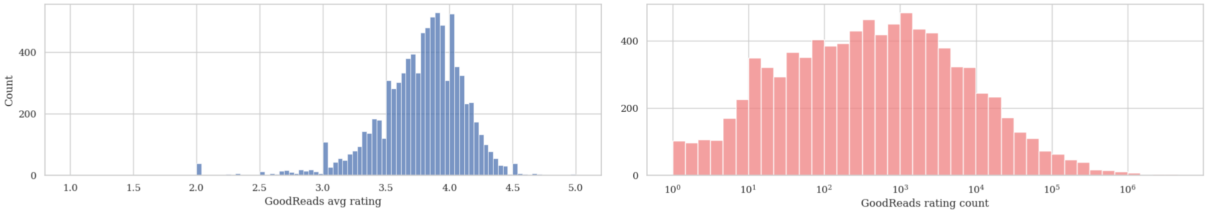


Figure 1: Distribution of GoodReads' ratings and number of ratings in our corpus. Note that the latter is logarithmically scaled.

forms such as GoodReads (Kousha et al., 2017), or by relying on established literary canons (Wilkens, 2012). Despite their diversity, different concepts of quality display large overlaps (Walsh and Antoniak, 2021a), thus to a degree allowing for the comparison across canons and preferences(Underwood, 2019; Wilkens, 2012).

3 Methods

3.1 Corpus

We use the Chicago corpus: over 9,000 English-language novels written in, or translated into English from 1880 to 2000, compiled based on the number of libraries that hold a copy, with a preference for more widely held titles. As such, the corpus is diverse, ranging from well-known genres of popular fiction to important works of "highbrow" literature, including novels from Nobel Prize winners (Bizzoni et al., 2022b) and other prestigious awards, as well as texts included in canonical collections like the Norton Anthology (Shesgreen, 2009). Yet, the corpus has an obvious cultural and geographic bias, with a strong over-representation of Anglophone authors. For this study, we used the whole corpus, as well as a subset of the corpus where 140 titles were filtered out because of their very low rating on GoodReads. We refer to this as the filtered corpus.

	Titles	**Authors**
Number	9089	3150
Number below 2.5 rating	140	118
Avg. ratings	3.74	3.69

Table 1: Number of titles and authors in the corpus and below the rating of 2.5, and avg. number of ratings

3.2 Quality Measures

As a source of quality judgments we decided to opt for **GoodReads**' average ratings. [1] This metric has limitations: i.a., reducing very different reader preferences and backgrounds to one single score (ratings or "stars"), conflating underlying motivations and important differences among readers. Still, the resource has a uniquely large number of users, facilitating an unprecedented amount of data for quantitative literary analysis, where popular titles are graded by hundreds of thousands of users (Kousha et al., 2017). The advantage of GoodReads is its wide audience, not only in terms of numbers, but because it reaches across genres and curricula (Walsh and Antoniak, 2021a), deriving its scores from a particularly diverse pool of readers, as the platform is accessed from several countries, by users of different genders, ages, etc.

3.3 Stylometric Features

Considering traditional stylometric features, we examine texts' adjusted **lexical diversity** as a measure of proven stylistic importance with obvious cognitive effects on the readers (Torruella and Capsada, 2013); the texts' ratio of **compressibility**, a measure of redundancy and formulaicity (Benedetto et al., 2002; van Cranenburgh and Bod, 2017); five different measures of textual **readability**[2], (based on, i.a., sentence length, word length, and number of syllables),and several grammatical and **syntactic** features, such as the frequency of parts of speech and of a selection of syntagms such as subjects, passive auxiliaries and relative clauses (see Appendix).

3.4 Sentiment Analysis

To build the sentiment arcs of each novel we opted for a simple and "classic" sentiment analysis algorithm: the **VADER** model (Hutto and Gilbert,

---

[1] https://www.goodreads.com

[2] The Flesch Reading Ease, the Flesch-Kincaid Grade Level, the SMOG Readability Formula, the Automated Readability Index, and the New Dale–Chall Readability Formula.

2014), applied at the sentence level. We chose this method because it is transparent, being based on a lexicon and a set of rules. It is widely employed and shows good performance and consistency across various domains (Ribeiro et al., 2016; Reagan et al., 2016), which is an ideal feature when dealing with narrative, as it enables the comparison across genres, while its origins in social media analysis do not appear to hinder the annotation of literary texts (Bizzoni et al., 2022b). Moreover, plotted arcs appear comparable to the **Syuzet-package** (Elkins and Chun, 2019), one specifically developed for narrative texts (Jockers, 2017), while side-stepping some of the problems of this package (Swafford, 2015), such as those inherent to word-based annotation. To assure the validity of the method, we manually inspected a selection of novels at global and local level (fig. 2, 3). As fig. 2 and 3 show, the high and dips appear to adequately correspond to narrative events, and performance is also good on the sentence-level when looking at the VADER annotation of, for example, the first lines and the corresponding text.[3]

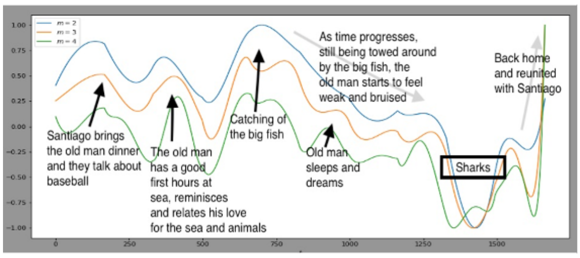


Figure 2: Sentiment arc of Hemingway's *The Old Man and the Sea* with different polynomial fits (m = polynomial degree). Y-axis values represent compound sentiment score (VADER). Values on the x-axis represent the narrative progression by number of sentences.

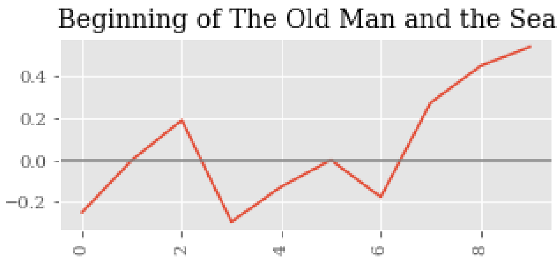


Figure 3: First sentences of Hemingway's *The Old Man and the Sea*, annotated with VADER.

From the annotated arcs, we extracted simple sentiment-arc features: **mean sentiment**, its **standard deviation**, the **mean sentiment of the ending 10 percent of each arc**, the **mean sentiment of the beginning 10 percent of each arc**, as well as the **difference between the main part of the arc and the ending (10 percent)**. Moreover, we computed two more complex measures of arc coherence: their **Hurst** exponent, based on the detrended version of arcs, which is a measure of the long-term memory or persistence of a time series, and their **Approximate Entropy**, which is a measure of the complexity or irregularity of a time series, quantifying the likelihood that patterns will repeat at a later time. These measures of arcs' dynamics have recently proved promising for literary quality modelling (Hu et al., 2021; Bizzoni et al., 2022b).

3.5 Models

As we are particularly interested in the combinations of features that can more accurately predict ratings, we prefer relatively simple and interpretable regression models, using a small set of "classic" algorithms such as **Linear Regression**, **Lasso** and **Bayesian Ridge** (see the complete list in Appendix). Our interest in identifying combinations of features that can accurately predict ratings goes beyond simply achieving high prediction accuracy; we also prioritize interpretability of our model, making explicit the relationships between predictors and outcomes. Simple and interpretable regression models, such as Linear Regression, Lasso, and Bayesian Ridge, provide a number of benefits in this context. First of all, these models allow for direct and straightforward interpretations of feature influences. For example, the coefficients in linear regression quantify the change in response variable for a unit change in the predictors. This is especially useful in our case as we aim to under-

---

[3] Corresponding text: "He was an old man who fished alone in a skiff in the Gulf Stream and he had gone eighty-four days now without taking a fish. In the first forty days a boy had been with him. But after forty days without a fish the boy's parents had told him that the old man was now definitely and finally salão, which is the worst form of unlucky, and the boy had gone at their orders in another boat which caught three good fish the first week. It made the boy sad to see the old man come in each day with his skiff empty and he always went down to help him carry either the coiled lines or the gaff and harpoon and the sail that was furled around the mast. The sail was patched with flour sacks and, furled, it looked like the flag of permanent defeat. The old man was thin and gaunt with deep wrinkles in the back of his neck. The brown blotches of the benevolent skin cancer the sun brings from its reflection on the tropic sea were on his cheeks. The blotches ran well down the sides of his face and his hands had the deep-creased scars from handling heavy fish on the cords. But none of these scars were fresh. They were as old as erosions in a fishless desert. Everything about him was old except his eyes and they were the same color as the sea and were cheerful and undefeated."

	baseline	**Linear**	**Ridge**	**Lasso**	**ElasticNet**	**BayesRidge**	**Huber**	**Polynomial**	**TheilSen**
r2	-1.07	0.23 (0.21)	0.23 (0.21)	0.04 (0.03)	0.05 (0.04)	**0.24** (0.21)	0.13 (0.11)	-0.02 (0.16)	0.22 (0.23)
neg_rmse	0.72	-0.14 (-0.15)	-0.14 (-0.15)	-0.22 (-0.22)	-0.20 (-0.20)	**-0.14** (-0.15)	-0.16 (-0.16)	-0.28 (-0.15)	-0.15 (-0.15)
r2 (filtered)	-0.944	0.061 (0.04)	0.063 (0.05)	0.04 (-0.02)	0.04 (-0.02)	**0.07** (0.05)	-0.46 (0.05)	-0.40 (-0.01)	-0.18 (-0.01)
neg_rmse (filtered)	0.445	-0.10 (-0.1)	-0.09 (-0.1)	-0.10 (-0.11)	-0.10 (-0.11)	**-0.09** (-0.11)	-0.15 (-0.1)	-0.15 (-0.11)	-0.12 (-0.11)

Table 2: Performance (r2 and negative MSE) comparison of regression models using 5-fold cross-validation for the whole (upper) and filtered (lower) corpus, with and without sentiment features (in parenthesis). Lasso and ElasticNet underperform on the larger data-set due to coefficient shrinkage, while Polynomial Regression likely overfits. The best-performing model is Bayesian Ridge. A random baseline is included for comparison.

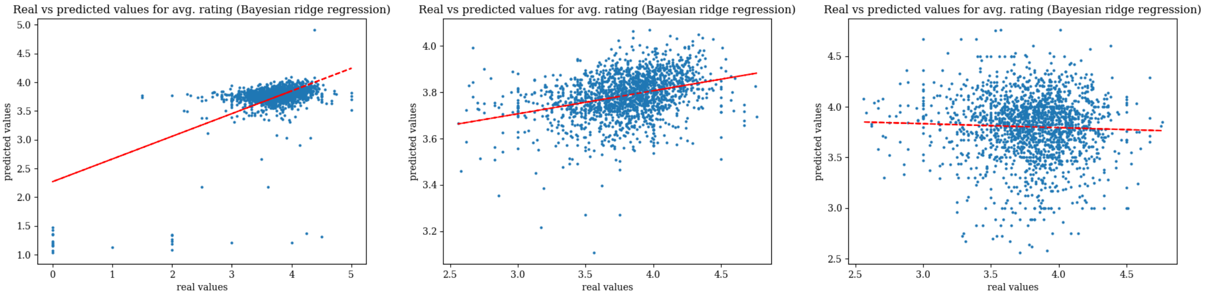


Figure 4: Distribution of real and predicted avg. rating values using Bayesian ridge regression. From left to right: 1) Whole corpus. Notice how ratings under 2.5 appear particularly predictable, despite their scarcity. 2) Filtered corpus. Even in the narrower interval ratings are relatively predictable. 3) Distribution of real and predicted avg. rating values in a random baseline for comparison.

stand not just how well we can predict the ratings, but how each individual feature influences these predictions. Secondly, these models are less prone to overfitting compared to deeper machine learning approaches. While deeper models can potentially yield higher predictive performance, they can also lead to models that are too complex, fitting the noise in our data rather than the underlying relationships. This would reduce the generalizability of our findings and potentially make them less reliable. Finally, using simpler models decreases the computational cost, which can be significant for more complex machine learning algorithms. This efficiency allows for more extensive model tuning and repeated testing, increasing the robustness of our results.

4 Results

Most models tested show predictive power, i.e., perform better than random. Their performance is reported in Table 2. This is our first important result since it would have been entirely possible that none of the chosen features had anything to do with large-scale reader appreciation. The behaviour of our models shows that combinations of some of the selected textual and narrative features can predict novels' average ratings on GoodReads. A second important finding is that sentiment measures improve the performance of almost all models: while a combination of syntactic, readability, and redundancy measures is already enough to partly model ratings, the novels' average sentiment, variation in sentiment intensity, and the overall predictability and persistence of the sentiment arcs increase our ability to predict perceived quality. When looking at the distribution of most models' predictions, we find an evident split: not only does the vast majority of GoodReads' ratings (in our corpus) fall between 3 and 5, with few low scores, but the distinction between very low-rated and the rest of the novels appears to be very easy to model: low rating titles have a distinctive textual and sentiment profile. To make sure we are not incurring in inflated scores due to the special predictability of this "low-rating group", we repeated the experiment with only the novels with a higher rating than 2.5 (still the majority, ca. 8900 titles). Also in this case, the models performed better than random: able to predict the "quality slope" better than chance (see fig. 4 for a visualization of model performance). Given the relative tightness of the scale and the potential volatility of the scores themselves, we find the models' performance far from obvious. We finally looked at the most predictive features. When modelling the whole corpus, readers' judgments of quality appear inversely related to punctuation, text compressibility, reading ease, verb, pronoun and adverb frequency, and directly

	coefficient
Whole corpus	
Punctuation freq.	-3.261
Text compressibility	-2.841
Flesch reading ease	-2.205
Stopword freq.	-2.100
Verb freq.	-1.502
Pronoun freq.	-1.502
Flesch-Kincaid grade level	1.380
Adverb freq.	-1.004
Noun freq.	0.941
Lexical richness	0.697
Filtered corpus	
Pronoun freq.	-1.419
Nominal subject freq.	-0.761
Lexical richness	0.602
Adjective freq.	-0.436
New Dale–Chall readability formula	-0.351
Stopword freq.	-0.323
Relative clause modifier freq.	-0.263
Text compressibility	-0.231

Table 3: Most important non-sentiment features for the best performing model (Bayesian Ridge) in the whole (upper) and filtered corpus (lower).

	coefficient
Whole corpus	
Approximate entropy	-1.500
Mean sentiment	-1.352
Difference between main and ending	1.152
Beginning sentiment	-0.935
Ending sentiment	0.861
Hurst	-0.649
Std. deviation sentiment	0.295
Filtered corpus	
Hurst	0.576
Std. deviation sentiment	0.214
Beginning sentiment	-0.169
Approximate entropy	0.148
Mean sentiment	0.082

Table 4: Most important sentiment features for the best performing model (Bayesian Ridge) in the whole (upper) and filtered corpus (lower).

related to lexical richness and reading difficulty. A simplistic style combined with many verbs, adverbs and pronouns is linked to lower ratings. The most important sentiment measures were, negatively, approximate entropy, Hurst and mean sentiment, and positively, the difference between the arc's mean and the ending's sentiment, and the ending sentiment. In other words, texts that have particularly chaotic and unpredictable arcs receive low scores, while higher average sentiment and endings with more positive values receive higher scores. When filtering out the "low-rating few", the landscape changes. Novels have a higher perceived quality if they tend towards fewer pronouns, explicit subjects, adjectives, stopwords, relative clauses and repetitions, a higher lexical richness, more nouns and a slightly easier vocabulary. These features suggest a style that is more sophisticated, diverse in vocabulary, and concise, with simpler or more direct sentences, and less reliant on nominal subjects and adjectives. At the sentiment level, the Hurst exponent is the strongest predictor: GoodReads users favour novels that have more persistent sentiment arcs without being too flat nor repetitive in their sentimental palette (having a higher standard deviation and slightly higher approximate entropy). Literary quality appears associated with novels that have strong, coherent, and dynamic emotional progressions and a broader range of sentiment, with more intricate and nuanced changes. They may also start in the low end of sentiment and maintain a slightly more positive tone throughout. Overall, these measures seem to point to an equilibrium between simplicity and diversity, both at the stylistic and at the sentiment level.

5 Conclusion and future works

We have tried a new set of experiments in the highly challenging task of modelling literary quality, represented as the online average ratings of readers, from a small set of textual and sentiment features. While a similar attempts have been made before (on smaller corpora), to the best of our knowledge, we are the first to show that the *addition* of several sentiment-related features improves the predictive power of most models. The sentiment features considered here were of two kinds: a global kind, such as the mean sentiment of a novel; and a dynamic kind, such as the level of entropy and fractality of the sentiment arcs. We have also found that the bottom 2% of titles elicit distinctly lower ratings, and that their appreciation is partly predictable through the textual features we have included. Finally, we analysed the features needed to predict perceived literary quality, noting that a balance between simplicity and diversity seems to characterize more appreciated titles. Naturally this is a study on a complex subject. In the future we aim to repeat the experiment optimizing for quality proxies beyond GoodReads ratings to study convergences between ways of defining quality, and use a larger set of features. We may also set it as a classification problem, and attempt more sophisticated models, as long as some interpretability remains.

References

Ebba Cecilia Ovesdotter Alm. 2008. *Affect in text and speech*. University of Illinois at Urbana-Champaign.

Dario Benedetto, Emanuele Caglioti, and Vittorio Loreto. 2002. Language Trees and Zipping. *Physical Review Letters*, 88(4):1–5.

Jan Beran. 1994. *Statistics for Long-Memory Processes*, 1 edition. Chapman and Hall/CRC, New York.

Yuri Bizzoni, Telma Peura, Kristoffer Nielbo, and Mads Thomsen. 2022a. Fractal sentiments and fairy tales-fractal scaling of narrative arcs as predictor of the perceived quality of andersen's fairy tales. *Journal of Data Mining & Digital Humanities*.

Yuri Bizzoni, Telma Peura, Kristoffer Nielbo, and Mads Thomsen. 2022b. Fractality of sentiment arcs for literary quality assessment: The case of nobel laureates. In *Proceedings of the 2nd International Workshop on Natural Language Processing for Digital Humanities*, pages 31–41, Taipei, Taiwan. Association for Computational Linguistics.

Julian Brooke, Adam Hammond, and Graeme Hirst. 2015. Gutentag: an nlp-driven tool for digital humanities research in the project gutenberg corpus. In *Proceedings of the Fourth Workshop on Computational Linguistics for Literature*, pages 42–47.

Erik Cambria, Dipankar Das, Sivaji Bandyopadhyay, and Antonio Feraco. 2017. Affective computing and sentiment analysis. In *A practical guide to sentiment analysis*, pages 1–10. Springer.

Irina-Ana Drobot. 2013. Affective narratology. the emotional structure of stories. *Philologica Jassyensia*, 9(2):338.

A. Eke, P. Herman, L. Kocsis, and L. R. Kozak. 2002. Fractal characterization of complexity in temporal physiological signals. *Physiological Measurement*, 23(1):R1.

Katherine Elkins and Jon Chun. 2019. Can Sentiment Analysis Reveal Structure in a Plotless Novel? ArXiv:1910.01441 [cs].

Qiyue Hu, Bin Liu, Mads Rosendahl Thomsen, Jianbo Gao, and Kristoffer L Nielbo. 2021. Dynamic evolution of sentiments in never let me go: Insights from multifractal theory and its implications for literary analysis. *Digital Scholarship in the Humanities*, 36(2):322–332.

Clayton Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the international AAAI conference on web and social media*, volume 8, pages 216–225.

SM Mazharul Islam, Xin Luna Dong, and Gerard de Melo. 2020. Domain-specific sentiment lexicons induced from labeled documents. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6576–6587.

Swapnil Jain, Shrikant Malviya, Rohit Mishra, and Uma Shanker Tiwary. 2017. Sentiment analysis: An empirical comparative study of various machine learning approaches. In *Proceedings of the 14th International Conference on Natural Language Processing (ICON-2017)*, pages 112–121, Kolkata, India. NLP Association of India.

Matthew Jockers. 2017. Syuzhet: Extracts sentiment and sentiment-derived plot arcs from text (version 1.0. 1).

Evgeny Kim and Roman Klinger. 2018. A survey on sentiment and emotion analysis for computational literary studies. *arXiv preprint arXiv:1808.03137*.

Corina Koolen, Karina van Dalen-Oskam, Andreas van Cranenburgh, and Erica Nagelhout. 2020a. Literary quality in the eye of the Dutch reader: The national reader survey. *Poetics*, 79:1–13.

Corina Koolen, Karina van Dalen-Oskam, Andreas van Cranenburgh, and Erica Nagelhout. 2020b. Literary quality in the eye of the dutch reader: The national reader survey. *Poetics*, 79:101439.

Kayvan Kousha, Mike Thelwall, and Mahshid Abdoli. 2017. Goodreads reviews to assess the wider impacts of books. 68(8):2004–2016.

Nikita Kuznetsov, Scott Bonnette, Jianbo Gao, and Michael A. Riley. 2013. Adaptive Fractal Analysis Reveals Limits to Fractal Scaling in Center of Pressure Trajectories. *Annals of Biomedical Engineering*, 41(8):1646–1660.

Xin Li, Lidong Bing, Wenxuan Zhang, and Wai Lam. 2019. Exploiting BERT for end-to-end aspect-based sentiment analysis. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 34–41, Hong Kong, China. Association for Computational Linguistics.

Benoit Mandelbrot. 1982. *The Fractal Geometry of Nature*. Times Books, San Francisco.

Benoit B. Mandelbrot. 1997. *Fractals and Scaling in Finance: Discontinuity, Concentration, Risk. Selecta Volume E*, 1997 edition edition. Springer, New York.

Benoit B. Mandelbrot and John W. Van Ness. 1968. Fractional Brownian Motions, Fractional Noises and Applications. *SIAM Review*, 10(4):422–437.

Saif Mohammad. 2018. Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 English words. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 174–184, Melbourne, Australia. Association for Computational Linguistics.

Saif Mohammad and Peter Turney. 2013. Nrc emotion lexicon. *National Research Council, Canada*, 2:1–234.

Mika V. Mäntylä, Daniel Graziotin, and Miikka Kuutila. 2018. The evolution of sentiment analysis—a review of research topics, venues, and top cited papers. 27:16–32.

Andrew J. Reagan, Brian Tivnan, Jake Ryland Williams, Christopher M. Danforth, and Peter Sheridan Dodds. 2016. Benchmarking sentiment analysis methods for large-scale texts: A case for using continuum-scored words and word shift graphs. ArXiv:1512.00531 [cs].

Filipe N. Ribeiro, Matheus Araújo, Pollyanna Gonçalves, Marcos André Gonçalves, and Fabrício Benevenuto. 2016. SentiBench - a benchmark comparison of state-of-the-practice sentiment analysis methods. *EPJ Data Science*, 5(1):1–29. Number: 1 Publisher: SpringerOpen.

Sean Shesgreen. 2009. Canonizing the canonizer: A short history of the norton anthology of english literature. *Critical Inquiry*, 35(2):293–318.

Annie Swafford. 2015. Problems with the Syuzhet Package.

Joan Torruella and Ramon Capsada. 2013. Lexical statistics and tipological structures: A measure of lexical richness. *Procedia - Social and Behavioral Sciences*, 95:447–454.

Ted Underwood. 2019. *Distant Horizons: Digital Evidence and Literary Change*. University of Chicago Press. Publication Title: Distant Horizons.

Andreas van Cranenburgh and Rens Bod. 2017. A data-oriented model of literary language. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1228–1238, Valencia, Spain. Association for Computational Linguistics.

Andreas van Cranenburgh and Corina Koolen. 2020. Results of a single blind literary taste test with short anonymized novel fragments. *arXiv preprint arXiv:2011.01624*.

Melanie Walsh and Maria Antoniak. 2021a. The goodreads 'classics': A computational study of readers, amazon, and crowdsourced amateur criticism. *Journal of Cultural Analytics*, 4:243–287.

Melanie Walsh and Maria Antoniak. 2021b. The Goodreads "Classics": A Computational Study of Readers, Amazon, and Crowdsourced Amateur Criticism. *Post45: Peer Reviewed*.

Matthew Wilkens. 2012. Canons, close reading, and the evolution of method. *Debates in the digital humanities*, pages 249–58.

Readability measures
Flesch reading ease
Flesch-Kincaid Grade Level
SMOG Readability Formula
Automated Readability Index
New Dale–Chall Readability Formula
Stylometric features
Lexical diversity
Text compressibility
Sentence length
Syntactic features
Verb frequency
Noun frequency
Adjective frequency
Adverb frequency
Pronoun frequency
Punctuation frequency
Stopword frequency
Nominal subject frequency
Auxilary frequency
Passive auxiliary frequency
Relative clause modifier frequency
Negation modifier frequency
Simple sentiment arc features
Mean sentiment
Std. deviation sentiment
Ending sentiment
Beginning sentiment
Difference between main and ending
Sentiment arc measures
Hurst
Approximate entropy

Table 6: Textual and arc-features considered

Linear regression
Ridge regression
Lasso
Elastic net regularization
Bayes ridge regression
Huber loss
Polynomial
PLS
TheilSen

Table 5: Complete list of models

	Linear	Ridge	Lasso	ElasticNet	BayesRidge	Huber	Polynomial	TheilSen
Std. Deviation	0.052	0.045	0.03	0.03	0.049	0.037	0.14	0.121

Table 7: Standard deviation per model on the filtered corpus, not considering sentiment-features.

Instruction Tuning for Few-Shot Aspect-Based Sentiment Analysis

Siddharth Varia[1*] **Shuai Wang**[1*] **Kishaloy Halder**[1*] **Robert Vacareanu**[2†*]
Miguel Ballesteros[1] **Yassine Benajiba**[1] **Neha Anna John**[1]
Rishita Anubhai[1] **Smaranda Muresan**[1] **Dan Roth**[1]
[1]AWS AI Labs
[2]University of Arizona, Tucson, AZ, USA
`{siddhvar, wshui, kishaloh, ballemig, benajiy, ranubhai, smaranm, drot}@amazon.com`
`rvacareanu@arizona.edu`

Abstract

Aspect-based Sentiment Analysis (ABSA) is a fine-grained sentiment analysis task which involves four elements from user-generated texts: aspect term, aspect category, opinion term, and sentiment polarity. Most computational approaches focus on some of the ABSA sub-tasks such as tuple (aspect term, sentiment polarity) or triplet (aspect term, opinion term, sentiment polarity) extraction using either pipeline or joint modeling approaches. Recently, generative approaches have been proposed to extract all four elements as (one or more) quadruplets from text as a single task. In this work, we take a step further and propose a unified framework for solving ABSA, and the associated sub-tasks to improve the performance in few-shot scenarios. To this end, we fine-tune a `T5` model with *instructional prompts* in a multi-task learning fashion covering all the sub-tasks, as well as the entire quadruple prediction task. In experiments with multiple benchmark datasets, we show that the proposed multi-task prompting approach brings performance boost (by absolute 8.29 F1) in the few-shot learning setting.

1 Introduction

Aspect-Based Sentiment Analysis (ABSA) is a fine-grained sentiment analysis task where the goal is to extract the sentiment associated with an entity and all its aspects (Liu, 2012; Pontiki et al., 2014, 2015, 2016; Schouten and Frasincar, 2015; Zhang et al., 2018; Nazir et al., 2020; Zhang et al., 2022). For example, in the context of Restaurant reviews the relevant aspects could be *food, ambience, location, service* with *general* used to represent the subject itself (i.e., restaurant). ABSA can provide valuable fine-grained information for businesses to analyze the aspects they care about. Annotated datasets have been released to foster research in this area (Pontiki et al., 2014, 2015, 2016).

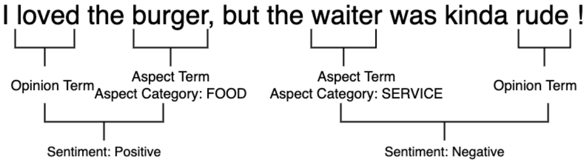


Figure 1: Illustrative orientation of four ABSA elements *i.e.,* Aspect Term, Aspect Category, Opinion Term, and Sentiment. The related tasks often involve predicting either everything together or a subset of them.

A full ABSA task aims to extract four elements from a user-generated text: aspect term, aspect category, opinion term and the sentiment polarity (see Figure 1 for an example). Most existing approaches have the focus on extracting some of these elements such as a single element (*e.g.,* aspect term), tuple (*e.g.,* aspect term, sentiment polarity), or triplet (*e.g.,* aspect term, aspect category, sentiment polarity) (Li et al., 2020; Hu et al., 2019; Xu et al., 2020a). Recently, Zhang et al. (2021a) tackled the full ABSA task, under the name of Aspect Sentiment Quadruple Prediction (ASQP). Technically, most existing computational approaches have used extractive and discriminative models either in a pipeline or in an end-to-end framework (Wang et al., 2016; Yu et al., 2019; Cai et al., 2021) to address ABSA. Generative approaches have been recently shown to be effective for the full ABSA task and its sub-tasks (Zhang et al., 2021a,b; Yan et al., 2021). Most notably, Zhang et al. (2021a) used a sequence-to-sequence (seq-to-seq) model to address ASQP as a paraphrase generation problem. One important consideration is that modeling ABSA in a generative

---

[*] Indicates equal contribution.

[†] Work done during internship at AWS.

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 19–27
July 14, 2023 ©2023 Association for Computational Linguistics

fashion allows for cross-task knowledge transfer.

We go a step further and propose a unified model that can tackle multiple ABSA sub-tasks, including the ASQP task, and explore its effectiveness for low data scenarios. Recent work on large language models relies on the intuition that most natural language processing tasks can be described via natural language instructions and that models trained on these instructions show strong zero-shot performance on several tasks (Wei et al., 2021; Sanh et al., 2022). Based on this success, we propose a unified model based on multi-task prompting with instructional prompts using T5 (Raffel et al., 2020) to solve the full ABSA task *i.e.,* ASQP (Zhang et al., 2021a) and several of its associated sub-tasks addressed in the literature: 1) Aspect term Extraction (AE) (Jakob and Gurevych, 2010); 2) Aspect term Extraction and Sentiment Classification (AESC) (Yan et al., 2021); 3) Target Aspect Sentiment Detection (TASD), which aims to extract the aspect term, aspect category, and sentiment polarity (Wan et al., 2020); 4) Aspect Sentiment Triplet Extraction (ASTE), which aims to extract the aspect term, opinion term, sentiment polarity (Peng et al., 2020). We conduct an extensive set of experiments with multiple review datasets. Experimental results show that our proposed model achieves substantial improvement (8.29 F1 on average) against the state-of-the-art in few-shot learning scenario[1].

2 Methods

The four elements of ABSA form a quadruple as the sentiments are associated with both the aspect, and the opinion terms (*cf* Figure 1). In this work, we hypothesize that it is important to capture the interaction between these components not only at the quadruple level, but also within a subset of these four elements.

We consider multiple factorized sub-tasks involving one or more of the four elements to be predicted. We pose it as a combination of five Question Answering (QA) tasks as illustrated in Figure 2. For each QA task, an instructional prompt is used to train a seq-to-seq model to learn one or more ABSA elements – referred to as Instruction Tuning (IT). Our formulation enables learning all sub-tasks via Multi-Task Learning (MTL).

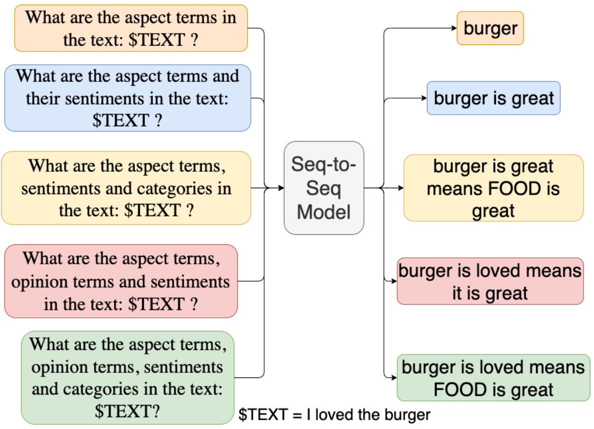

What are the aspect terms in the text: $TEXT ?
What are the aspect terms and their sentiments in the text: $TEXT ?
What are the aspect terms, sentiments and categories in the text: $TEXT ?
What are the aspect terms, opinion terms and sentiments in the text: $TEXT ?
What are the aspect terms, opinion terms, sentiments and categories in the text: $TEXT?
Seq-to-Seq Model
burger
burger is great
burger is great means FOOD is great
burger is loved means it is great
burger is loved means FOOD is great
$TEXT = I loved the burger

Figure 2: Instruction tuning to solve the sub-tasks related to ABSA. We devise multiple prompts to instruct a seq-to-seq model to learn in multi-task learning manner.

2.1 Input Transformation

First, we transform each sentence in the corpus using the instruction templates provided for each task as shown in Table 1. Furthermore, we use multiple paraphrased instruction templates as shown in Table 2 for a task, and sample randomly when preparing a batch during training (and evaluation) of the seq-to-seq model. However, the target output sequence remains unchanged irrespective of the template sampled for a task.

2.2 Model Training

Next, we perform IT with the seq-to-seq model. We train it in a MTL fashion where input-output combinations are sampled from all tasks simultaneously. We use the following loss for model training:

$$\mathcal{L} = -\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{n}\log p_{\theta}(y_i|y_1, ..., y_{i-1}, \mathbf{x_t}). \quad (1)$$

where $\mathbf{x_t}$ is the transformed input sequence ($\mathbf{x}$) for t^{th} task. θ is the set of model parameters. n is the length of output sequence. y_i is the ith token in output sequence. T is the number of tasks. The model parameters are updated using `Adam` optimizer with weight decay (Loshchilov and Hutter, 2019).

2.3 Output Transformation

Finally, we transform the output using the templates provided in the rightmost column in Table 1. In case there is more than one quadruple in the output, we use a special separation token `[SSEP]`. We

---

[1] Sources available at: `https://github.com/amazon-science/instruction-tuning-for-absa`

Task	$AT	$AC	$S	$OT	Input Instruction	Output
Aspect Extraction (AE)	✓				Given the text: $TEXT, what are the aspect terms in it ? What are the aspect terms in the text: $TEXT ?	Template: $AT Literal: *burger*
Aspect term Extraction and Sentiment Classification (AESC)	✓		✓		Given the text: $TEXT, what are the aspect terms and their sentiments ? What are the aspect terms and their sentiments in the text: $TEXT ?	Template: $AT is $S Literal: *burger is great*
Target Aspect Sentiment Detection (TASD)	✓	✓	✓		Given the text: $TEXT, what are the aspect terms, sentiments and categories ? What are the aspect terms, sentiments and categories in the text: $TEXT ?	Template: $AT is $S means $AC is $S Literal: *burger is great means food is great*
Aspect Sentiment Triplet Extraction (ASTE)	✓		✓	✓	Given the text: $TEXT, what are the aspect terms, opinion terms and sentiments ? What are the aspect terms, opinion terms and sentiments in the text: $TEXT ?	Template: $AT is $OT means it is $S Literal: *burger is loved means it is great*
Aspect Sentiment Quadruple Prediction (ASQP)	✓	✓	✓	✓	Given the text: $TEXT, what are the aspect terms, opinion terms, sentiments and categories ? What are the aspect terms, opinion terms, sentiments and categories in the text: $TEXT ?	Template: $AT is $OT means $AC is $S Literal: *burger is loved means food is great*

Table 1: The factorized sub-tasks in ABSA. Each of them covers a sub-set of all four prediction targets. $AT: Aspect Term; $AC: Aspect Category; $S: Sentiment; $OT: Opinion Term; $TEXT: input text. Both templates and literal values (for $TEXT = *I loved the burger*) are shown for Output against each task.

map sentiment classes positive, negative and neutral to *great, bad* and *ok* respectively in the output similar to (Zhang et al., 2021a). During inference, we apply the reverse transformations to recover the quadruples for evaluation.

3 Experiments

As this work is one of the first few attempts towards studying few-shot learning in ABSA context, unsurprisingly, there is a lack of standard few-shot datasets. We emulate few-shot data drawing inspiration from the literature (Halder et al., 2020; Ma et al., 2022) for our experiments.

3.1 Datasets: Few-shot Preparation

We use three datasets, Rest15, Rest16 from (Zhang et al., 2021a) and Laptop14 from (Xu et al., 2020b). For the first two, we shuffle the data with fixed random seed, and select first few samples so that there are at least k samples from each aspect category[2]. As Laptop14 does not have aspect category annotations, we select k examples per sentiment class instead, following the same principle (statistics in Table 5).

3.2 Baselines and Models for Comparison

As a strong baseline, we consider Paraphrase (or Para) model[3] – the current state-of-the-art for TASD, ASTE, and ASQP tasks (Zhang et al., 2021a). It uses the same backbone model as of ours, which ensures fair comparison. However, for the other two tasks Para is not applicable, hence we use a generative framework called BartAbsa as the baseline (Yan et al., 2021). All the Para numbers are obtained using our implementation for a fair comparison (*cf* Section A.5).

To understand the impact of all the components in our approach, we consider two model ablations:

1. **Text**: $TEXT is directly used as input
2. **IT**: $TEXT is transformed to instructions

We refer to our full proposed model as **IT-MTL**, it covers all the tasks. Table 3 provides illustrations of the input prompts for the ablations.

3.3 Experimental Setup

We use `t5-base` (Raffel et al., 2020) as the backbone for our models. Results are averaged over 5 runs with random seeds (*cf* Section A.2 for all details). Micro F1 is the evaluation metric following previous work (Zhang et al., 2021a).

---

[2] It is not feasible to guarantee exactly k samples since an example can have multiple aspect categories. (Ma et al., 2022)

[3] Other competitive models can be found in (Zhang et al., 2021a). Since Para has outperformed them, we focus on it.

Task	Input Prompts
AE	Given the text: $TEXT, what are the aspect terms in it ?
	What are the aspect terms in the text: $TEXT ?
ASE	Given the text: $TEXT, what are the aspect terms and their sentiments ?
	What are the aspect terms and their sentiments in the text: $TEXT ?
TASD	Given the text: $TEXT, what are the aspect terms, sentiments and categories ?
	What are the aspect terms, sentiments and categories in the text: $TEXT ?
	Given the text: $TEXT, what are the aspect terms, categories and sentiments ?
	What are the aspect terms, categories and sentiments in the text: $TEXT ?
ASTE	Given the text: $TEXT, what are the aspect terms, opinion terms and sentiments ?
	What are the aspect terms, opinion terms and sentiments in the text: $TEXT ?
	Given the text: $TEXT, what are the opinion terms, aspect terms and sentiments ?
	What are the opinion terms, aspect terms and sentiments in the text: $TEXT ?
ASQP	Given the text: $TEXT, what are the aspect terms, opinion terms, sentiments and categories ?
	What are the aspect terms, opinion terms, sentiments and categories in the text: $TEXT ?
	Given the text: $TEXT, what are the aspect terms, opinion terms, categories and sentiments ?
	What are the aspect terms, opinion terms, categories and sentiments in the text: $TEXT ?
	Given the text: $TEXT, what are the opinion terms, aspect terms, sentiments and categories ?
	What are the opinion terms, aspect terms, sentiments and categories in the text: $TEXT ?
	Given the text: $TEXT, what are the opinion terms, aspect terms, categories and sentiments ?
	What are the opinion terms, aspect terms, categories and sentiments in the text: $TEXT ?

Table 2: List of input instruction prompts for all the five sub-tasks. $TEXT is the place holder for actual text.

Ablation	**Input Prompt**
Text	$TEXT
IT	What are the aspect terms in the text: $TEXT?
IT-MTL	What are the aspect terms in the text: $TEXT?
	What are the aspect terms and their sentiments in the text: $TEXT?
	Given the text: $TEXT, what are the aspect terms, sentiments and categories?
	Given the text: $TEXT, what are the aspect terms, opinion terms and sentiments?
	What are the aspect terms, opinion terms, sentiments and categories in the text: $TEXT ?

Table 3: Illustration of input prompts to the seq-to-seq model for various ablations of our proposed approach.

3.4 Results

We present results for all the datasets in Table 4. Since, Laptop14 lacks aspect category annotations, TASD and ASQP are not applicable. We make four key observations from the results.

Ablation Study: First, IT beats Text in most settings proving effectiveness of our instructions. Second, we observe that IT-MTL outperforms others on Rest15, and Rest16 substantially in few-shot settings, except on Laptop14 as IT-MTL underperforms on AE task. This might be attributed to the absence of TASD, ASQP tasks. Overall, we observe the trend IT-MTL $>$ IT $>$ Text.

Baseline Comparison: Third, our proposed IT-MTL approach outperforms Para, and BartAbsa comfortably in most few-shot settings across all datasets with a performance boost of 8.29 F1 on average. We observe some exceptions in Laptop14, where Para outperforms IT-MTL slightly on ASTE – possibly due to the missing tasks that involve aspect category annotations. Fourth, we also experiment with the full training datasets and summarize them in Figure 3. In 4 out of 5 tasks, our IT-MTL model either outperforms or does at par with the SOTA baselines. Interestingly, in case of AE, it falls behind BartAbsa by 3.5 F1 scores. We attribute this difference to the advanced decoding strategies used in BartAbsa which are orthogonal to our work.

Regarding the randomness introduced by the seeds, we observe that the model training is reasonably stable across tasks (*cf* Table 6). Overall, we conclude that in few-shot settings, our proposed IT-MTL leverages the knowledge from multiple tasks, and improves the generalization of the underlying seq-to-seq model across all the ABSA tasks.

4 Conclusion

In this paper, we posed ABSA as an instruction tuning based seq-to-seq modeling task. We factorized the overall quadruple prediction task into five sub-tasks resembling Question Answering tasks. We proposed a multi-task learning based approach using a pre-trained seq-to-seq model. We experimented with customer reviews from two domains, showed that our approach gives superior performance compared to baseline models in few-shot, and stays comparable in full fine-tuning scenarios.

Task	**Model**	**K=5**	**K=10**	**K=20**	**K=50**
AE	BARTABSA	19.68	42.99	57.43	63.48
	Text	43.95	54.38	59.75	61.75
	IT	**45.24**	55.1	60.33	64.15
	IT-MTL	44.18	**56.57**	**62.65**	**67.22**
AESC	BARTABSA	10.77	27.38	42.23	52.55
	Text	37.33	47.68	50.6	56.69
	IT	**39.4**	**49.43**	52.06	58.4
	IT-MTL	38.99	47.62	**53.58**	**59.54**
TASD	PARA.	21.34	**37.39**	42.52	47.57
	Text	22.55	36.37	42.28	48.52
	IT	22.92	36.52	43.2	50.14
	IT-MTL	**27.05**	36.81	**43.56**	**50.24**
ASTE	PARA.	22.07	32.49	36.28	41.12
	Text	18.49	30.17	35.66	41.49
	IT	22.38	32.11	36.67	41.65
	IT-MTL	**22.7**	**33.52**	**37.78**	**43.84**
ASQP	PARA.	13.65	22.90	27.87	34.49
	Text	12.15	22.19	28.82	33.96
	IT	13.3	24.35	29.66	36.78
	IT-MTL	**15.54**	**25.46**	**31.47**	**37.72**

(a) REST15

Task	**Model**	**K=5**	**K=10**	**K=20**	**K=50**
AE	BARTABSA	31.48	55.90	62.96	71.06
	Text	52.7	58.5	61.49	67.21
	IT	55.64	59.36	63.75	68.14
	IT-MTL	**59.41**	**61.87**	**66.88**	**71.18**
AESC	BARTABSA	25.45	46.31	53.27	62.90
	Text	49.13	54.54	57.05	62.75
	IT	51.93	55.29	59.96	63.45
	IT-MTL	**52.42**	**55.37**	**60.22**	**65.14**
TASD	PARA.	28.93	38.99	48.29	54.89
	Text	30.65	38.39	46.72	54.04
	IT	34.38	38.58	47.66	55.16
	IT-MTL	**40.45**	**42.41**	**48.83**	**55.82**
ASTE	PARA.	32.48	38.90	43.51	51.47
	Text	28.44	38.23	42.12	50.9
	IT	33.08	**41.12**	44.08	51.69
	IT-MTL	**35.75**	38.95	**44.75**	**52.94**
ASQP	PARA.	20.02	28.58	36.26	43.50
	Text	20.98	28.06	35.04	45.26
	IT	23.86	30.02	37.20	46.9
	IT-MTL	**27.02**	**31.66**	**38.06**	**47.48**

(b) REST16

Task	**Model**	**K=5**	**K=10**	**K=20**	**K=50**
AE	BARTABSA	–	5.54	33.04	60.98
	Text	**34.64**	42.26	51.11	59.62
	IT	34.29	**47.4**	52.39	**63.86**
	IT-MTL	31.54	42.73	**53.08**	63.71
AESC	BARTABSA	–	4.75	24.92	50.01
	Text	21.68	30.7	37.74	50.39
	IT	23.28	**36.55**	43.39	52.92
	IT-MTL	**25.01**	34.44	**44.5**	**53.75**
ASTE	PARA.	**14.99**	23.87	30.12	**43.75**
	Text	10.10	16.27	26.37	39.65
	IT	12.60	21.31	30.03	41.91
	IT-MTL	14.18	**24.09**	**32.39**	42.62

(c) LAPTOP14

Table 4: Comparison of IT-MTL with baselines. **Bolded:** best, Underlined: second-best. '–' denotes the model failed to obtain a non-zero score.

5 Limitations

First, our work essentially relies upon a generative language model to understand the relationships between the sentiment elements in contrast to discriminative/extractive models which make structured predictions by design. As a result, our model is susceptible to usual anomalies suffered by generative models *e.g.,* malformed outputs. We recover the quadruples from the model's output sequence using regular expression based matching with fixed templates, as a result, an end-user will never receive any irrelevant text generated by the model. However, the accuracy will still be impacted in such cases nevertheless. Second, input sequences in user-generated content can be arbitrarily long and that might result in increased decoding time because of the underlying generative model. Last but not the least, all the instruction templates we provide in this work are designed solely for English. It would be interesting to explore systematic ways to be more language inclusive for instruction tuning based ABSA.

References

Hongjie Cai, Rui Xia, and Jianfei Yu. 2021. Aspect-category-opinion-sentiment quadruple extraction with implicit aspects and opinions. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 340–350, Online. Association for Computational Linguistics.

Kishaloy Halder, Alan Akbik, Josip Krapac, and Roland Vollgraf. 2020. Task aware representation of sentences for generic text classification. In *COLING 2020, 28th International Conference on Computational Linguistics*.

Minghao Hu, Yuxing Peng, Zhen Huang, Dongsheng Li, and Yiwei Lv. 2019. Open-domain targeted sentiment analysis via span-based extraction and classification. In *Proceedings of the 57th Annual Meet-*

ing of the Association for Computational Linguistics, pages 537–546.

Niklas Jakob and Iryna Gurevych. 2010. Extracting opinion targets in a single and cross-domain setting with conditional random fields. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 1035–1045.

Kun Li, Chengbo Chen, Xiaojun Quan, Qing Ling, and Yan Song. 2020. Conditional augmentation for aspect term extraction via masked sequence-to-sequence generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7056–7066.

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR*.

Jie Ma, Miguel Ballesteros, Srikanth Doss, Rishita Anubhai, Sunil Mallya, Yaser Al-Onaizan, and Dan Roth. 2022. Label semantics for few shot named entity recognition. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1956–1971.

Ambreen Nazir, Yuan Rao, Lianwei Wu, and Ling Sun. 2020. Issues and challenges of aspect-based sentiment analysis: a comprehensive survey. *IEEE Transactions on Affective Computing*.

Haiyun Peng, Lu Xu, Lidong Bing, Fei Huang, Wei Lu, and Luo Si. 2020. Knowing what, how and why: A near complete solution for aspect-based sentiment analysis. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8600–8607.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad AL-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, Véronique Hoste, Marianna Apidianaki, Xavier Tannier, Natalia Loukachevitch, Evgeniy Kotelnikov, Nuria Bel, Salud María Jiménez-Zafra, and Gülşen Eryiğit. 2016. SemEval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 19–30, San Diego, California. Association for Computational Linguistics.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. SemEval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 486–495, Denver, Colorado. Association for Computational Linguistics.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2022. Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference on Learning Representations*.

Kim Schouten and Flavius Frasincar. 2015. Survey on aspect-level sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering*, 28(3):813–830.

Hai Wan, Yufei Yang, Jianfeng Du, Yanan Liu, Kunxun Qi, and Jeff Z. Pan. 2020. Target-aspect-sentiment joint detection for aspect-based sentiment analysis. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9122–9129.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2016. Recursive neural conditional random fields for aspect-based sentiment analysis. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 616–626.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Lu Xu, Hao Li, Wei Lu, and Lidong Bing. 2020a. Position-aware tagging for aspect sentiment triplet extraction. *arXiv preprint arXiv:2010.02609*.

Lu Xu, Hao Li, Wei Lu, and Lidong Bing. 2020b. Position-aware tagging for aspect sentiment triplet extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2339–2349, Online. Association for Computational Linguistics.

Hang Yan, Junqi Dai, Tuo Ji, Xipeng Qiu, and Zheng Zhang. 2021. A unified generative framework for aspect-based sentiment analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International*

Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 2416–2429, Online. Association for Computational Linguistics.

Jianfei Yu, Jing Jiang, and Rui Xia. 2019. Global inference for aspect and opinion terms co-extraction based on multi-task neural networks. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 27(1):168–177.

Lei Zhang, Shuai Wang, and Bing Liu. 2018. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1253.

Wenxuan Zhang, Yang Deng, Xin Li, Yifei Yuan, Lidong Bing, and Wai Lam. 2021a. Aspect sentiment quad prediction as paraphrase generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9209–9219, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2021b. Towards generative aspect-based sentiment analysis. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 504–510, Online. Association for Computational Linguistics.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2022. A survey on aspect-based sentiment analysis: Tasks, methods, and challenges. *arXiv preprint arXiv:2203.01054*.

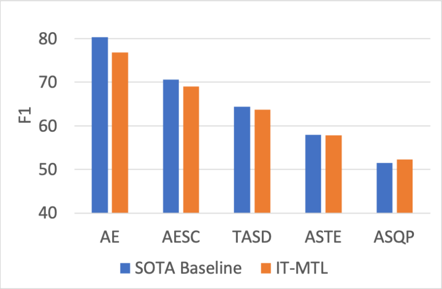


Figure 3: The average F1 scores achieved by our IT-MTL model and the relevant baseline. For AE, AESC the baseline is BARTABSA, and for others PARA. is the baseline.

A Appendix

A.1 List of input instruction prompts

A.2 Hyperparameters

We set the learning rate to 3e-4 for all the experiments in this paper. We train each model for a fixed number of 20 epochs similar to Zhang et al.. For full-shot experiments, we use a batch size of 16. For k=5, 10, 20 and 50 we use a batch size of 2, 2, 4 and 8 respectively. The maximum sequence length is set to 160. Longer sequences are truncated and shorter sequences are padded. Finally, we use Adam optimizer with weight decay.

A.3 Dataset Statistics

Table 5 presents the number of sentences in each dataset. Please note that for LAPTOP14 dataset, the few-shot data for different values of K was selected based on sentiment classes instead of Aspect category due to lack of category annotations.

A.4 Results on Full Datasets

The averaged results across full datasets (REST15, REST16 and LAPTOP14) are in Figure 3.

A.5 Implementation Issues

We extend Zhang et al. (2021a)'s library to implement our models. A careful reader might notice that the PARA and our text-only ablation should be similar as the only difference is in the output prompts. However, in practice we observe a large gap in few-shot performance between these two when we obtain the numbers for PARA with authors' published sources. Upon investigating, we discovered a few implementation issues in their sources. Our implementation improves PARA's F1 scores in few-shot settings and we report that to ensure a fair comparison. It brings the gap down from 6.75 to 2.32 in terms of absolute F1 scores between IT-MTL and PARA.

Evaluation Logic: We observe another critical issue in the evaluation logic in Zhang et al.'s sources[4]. It discounts the repetitions of the same tuple produced in the output. For illustration, let us assume for a review the target tuples for AE task are *burger, fries*. Now, if the seq-to-seq model outputs *burger, burger*, the logic in their sources computes the true positive count to be 2, whereas it should be only 1. This ultimately leads to an inflated F1 score. We fix this issue in our evaluation and comparisons with PARA. The reported F1 for PARA with the original logic was 61.13, after the fix it becomes 60.70 on full corpus of LAPTOP14. Overall, we observe that for few-shot cases, this issue becomes more apparent compared to the high-shot ones.

A.6 Stochasticity in Few-shot Data Sampling

So far, we keep the few-shot data fixed and vary the seed 5 times. To observe the effect of another form of stochasticity, in Table 7, we sample few-shot data 5 times for REST16 and keep the seed fixed. We observe that the trend remains the same.

Model	K=5	K=10	K=20	K=50
Text	21.99	29.3	37.92	46.83
IT	22.91	31.24	38.00	47.94
IT-MTL	**24.97**	**32.25**	**39.89**	**48.20**

Table 7: ASQP Results for REST16 averaged across 5 different k-shot samples.

---

[4] `https://github.com/IsakZhang/ABSA-QUAD/blob/master/eval_utils.py#L90`

	Rest15					Rest16					Laptop14				
	K=5	K=10	K=20	K=50	Full	K=5	K=10	K=20	K=50	Full	K=5	K=10	K=20	K=50	Full
Train	25	46	86	181	834	22	43	77	179	1264	11	19	40	106	906
Dev	21	35	68	140	209	26	42	73	159	316	8	16	34	86	219
Test	537					544					328				

Table 5: Number of sentences in each dataset. The same test set was used for few-shot and full-shot evaluation.

Dataset	Model	K=5	K=10	K=20	K=50
Rest15	Para.	13.65±0.92	22.90±0.50	27.87±1.64	34.49±0.64
	IT-MTL	15.54±1.61	25.46±1.09	31.47±0.58	37.72±0.76
Rest16	Para.	20.02±1.43	28.58±1.41	36.26±0.54	43.50±0.29
	IT-MTL	27.02±1.29	31.66±1.39	38.06±1.69	47.48±1.20

Table 6: Results (F1 ± standard deviation) for ASQP task. The F1 scores remain reasonably stable with the standard deviation being under $\sim$1.6 F1 points in all cases.

You Are What You Read: Inferring Personality From Consumed Textual Content

Adam Sutton and **Almog Simchon** and **Matthew Edwards** and **Stephan Lewandowsky**
University of Bristol
United Kingdom

Abstract

In this work we use consumed text to infer Big-5 personality inventories using data we have collected from the social media platform Reddit. We test our models on two datasets, sampled from participants who consumed either fiction content ($N = 913$) or news content ($N = 213$). We show that state-of-the-art models from a similar task using authored text do not translate well to this task, with average correlations of $r = .06$ between the model's predictions and ground-truth personality inventory dimensions. We propose an alternate method of generating average personality labels for each piece of text consumed, under which our model achieves correlations as high as $r = .34$ when predicting personality from the text being read.

1 Introduction

While authored text has previously been used for personality prediction (Eichstaedt et al., 2021), consumed text (the linguistic attributes of the text that people choose to read) has received no such attention. It is unclear if consumed text can be indicative of personality. Methods used in similar tasks may also not transfer to this domain, as a consumed piece of text is not unique to a single reader.

However, predicting reader personalities may help understand and reduce the impact of psychological micro-targeting, particularly in the domain of political advertising. Facebook has a psychological micro-targeting patent registered (Nowak and Eckles, 2014), and previous work indicates personality targeted messages increase desirable outcomes for advertisers (Matz et al., 2017).

As recent work has shown that targeted political advertising online has been more more effective than traditional methods (Zarouali et al., 2020; Goldberg et al., 2021; Tappin et al., 2022; Joyal-Desmarais et al., 2022), our work aims to reverse engineer the process of such psychological targeting, with the intention of developing countermeasures to remove or reduce the impact of this targeting. Here we first demonstrate that consumed text can be used to infer personality. This is significant, as we show that personality prediction of content consumers is possible even where there are not structural connections to known cultural touchpoints (which has been demonstrated previously by, e.g., Youyou et al. (2015)). Prediction of consumer personality from consumed text is highly tranferrable, being in principle applicable to any platform where users might read text. In the future we aim to develop tools for users that would flag articles or text that our model predicts could be congruent with their personality inventory.

In this paper we sample ∼1,100 participants from the social media website Reddit, using their public data and provided personality inventories to show that consumed text can also be indicative of the consumer's personality. Our models achieve Pearson's $r > 0.3$ between predicted personality dimension values and those provided using standard instruments. We also show that models that have achieved state-of-the-art performance when applied to produced text do not achieve suitable performance on consumed text.

2 Background

In the field of psychology, constructs such as personality are quantified using validated tools. One such tool is a personality inventory, where the outcome is usually represented as a numerical value for multiple personality dimensions. One example of this is the Big-5 model, which uses a questionnaire to capture people's personality along five dimensions (Soto and John, 2017; Goldberg, 1993). These scales enable measurement of personality, and in combination with access to large feature-rich datasets from social media they have enabled attempts at estimating people's personality from their behaviour (Bachrach et al., 2012; Schwartz et al., 2013). Machine learning has improved to

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 28–38
July 14, 2023 ©2023 Association for Computational Linguistics

the point where automated personality judgements can outperform humans at the same task (Youyou et al., 2015). This has also been found with textual content: various studies have shown that text produced by a user can be used to estimate their personality (Eichstaedt et al., 2021).

Language models have played a large part in the improvement of performance in many downstream natural language tasks in recent years (Pennington et al., 2014). The most recent development to have a substantial performance impact is attention (Vaswani et al., 2017; Bahdanau et al., 2014), which enables word representations that are dynamically generated based on surrounding text (i.e. "bark" will have different representations for a "dog's bark" and "tree bark"). This has resulted in a new generation of attention-based language models that reported state-of-the-art-performance for multiple NLP tasks (Devlin et al., 2018). These architectures are still being iterated on to improve performance (Zhong et al., 2022; Patra et al., 2022).

Attention has also been useful for personality modelling in the domain of produced text. Lynn et al. (2020) defines "message-level attention", which is based on the assumption that "not all documents are equally important". Models using this form of attention take multiple produced messages from an author and weigh the importance of each message according to a learned attention mechanism, in order to predict that author's personality. Lynn et al. (2020) represent the current state-of-the-art performance for this task, while also providing some interpretability of the model via message weights.

In this study we apply these message attention models to the domain of consumed text, alongside an alternative method that aims to predict the averaged personality profile of all known consumers of the article. We show that message attention models do not achieve desirable performance when applied to consumed text. Evaluation under averaged personality labelling shows promising performance in comparison. Our evaluation covers multiple datasets, spanning two different genres of text. We also trial the effectiveness of models predicting consumer personality using only article titles instead of the entire article. We find that the personalities of news readers are better predicted by our models than the consumers of fictional content, and predictions on the basis of news titles alone perform comparably to those informed by the content of the entire article.

3 Methods

3.1 Message Level Attention

For each personality dimension, given a set of N messages (or articles consumed) from a user u we encode each textual input ($article_i$) such that:

$$s_i = \Phi(article_i), \tag{1}$$

where Φ is the language model used to encode each consumed article. We then pass all vector representations (such that all $s_i \in S$) through another sequence model, multi-headed self-attention (MHA) (Vaswani et al., 2017):

$$S' = \text{MHA}(S). \tag{2}$$

We then apply the message attention mechanism to calculate articles that are most indicative of the personality of a given user, as proposed in Lynn et al. (2020):

$$h_i = \tanh(W_m s'_i + b_m) \tag{3}$$

$$m_i = \frac{\exp(h_i^\top h_m)}{\sum_{j=0}^{N} \exp(h_j^\top h_m)}, \tag{4}$$

where W_m and b_m are learned features for the encoders hidden state. h_m is a learned vector that judges how much attention should be paid to each article. Equation 4 is a softmax where all m_i will sum to 1.

Each value in m is a scalar that represents how important the attention mechanism considers its corresponding article vector s_i is, and scales it accordingly.

$$\hat{u} = \sum_{i=0}^{N} m_i s_i. \tag{5}$$

Equation 5 shows how the user representation is formulated using the weighted average summation of each article consumed by a user. The vector representation of the user is passed into a standard feed forward neural network such that:

$$f : \mathbb{R}^d \to \mathbb{R}, \tag{6}$$

which results in prediction of a single personality dimension for that user. Each personality dimension will have its own message attention and user representation weights calculated separately, to fine tune as accurately as possible.

3.2 Average Personality Per Article

With produced text it can be reasonably assumed that all messages produced are unique to that user. However that assumption does not hold for consumed text, as it is not intended to be unique to a single reader.

Our method assigns a single personality label for each article. The personality label for a given article is comprised of the average personality score of all participants who are known to have consumed that article. The underlying assumption of this method is that each article will target a large group of people that have an aggregate personality profile indirectly indicated in the text. We also assume that the average personality of known consumers is the likeliest approximation to the personality profile of the intended audience.

In contrast to our evaluations of message-level attention models, under this methodology only one article per training step is fed into the neural network. Labels in this model are the average personality of every user who has consumed the article.

For a given $article$ we calculate the language model representation as:

$$s = \Phi(article), \tag{7}$$

where Φ is the language model used to encode the article to a vector representation.

Unlike Section 3.1, no further processing is required to generate a vector used to predict the targeted personality, and a feed forward network is again used to estimate the personality of the average consumer of this article.

4 Dataset

Two datasets are used in our experiments, both sourced from the social media website Reddit between 2021-2022. Participants were invited to participate in a survey and gave permission for us to link their public post and comment history to their personality inventories as assessed via a Big-5 personality questionnaire (BFI-2) (Soto and John, 2017). We crawled the content of all posts our participants had commented upon, using commenting behaviour as an indication of text consumption. Our data collection and retention procedures were overseen by the relevant institutional ethics board.

Our two datasets cover different domains of content. Our news dataset contains news articles consumed by our participants from news-focused subreddits (communities dedicated to a specific topic), including *r/worldnews*, *r/politics* and *r/europe*. Our fiction dataset contains short fiction consumed by participants from subreddits devoted to sharing such content, such as *r/WritingPrompts*, *r/shortstories*, and *r/nosleep*.

Table 1: Number of users and articles that have been consumed for both datasets used in experiments. Note that these users may have consumed text from both domains.

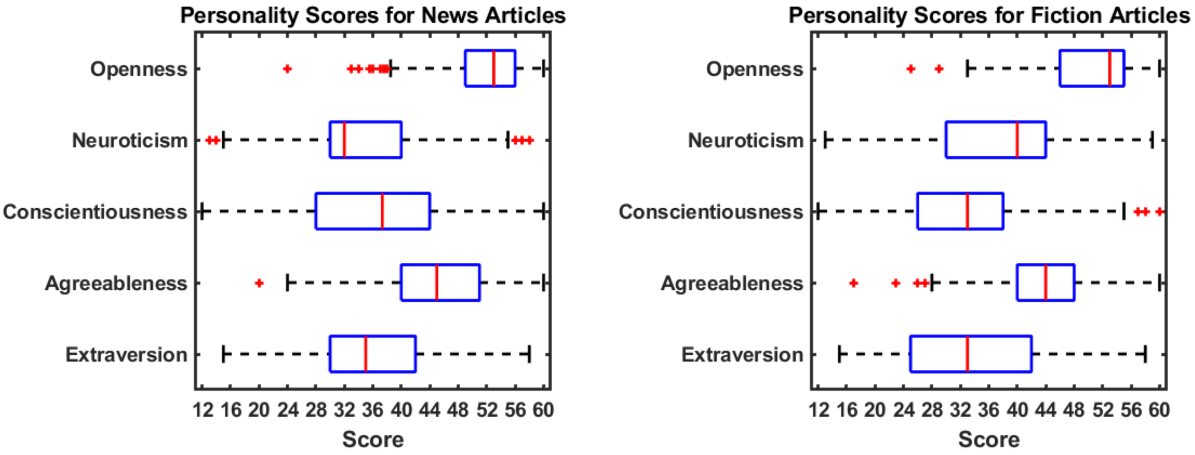
	News	Fiction
Users	213	953
Articles	19,609	4,000

News subreddits were chosen based on activity (number of users, and posts) and the majority of articles posted being URL submissions linking to news stories from external news sites. These subreddits are also moderated to remove unwanted content, such as spam or adverts. Fiction subreddits were also chosen based on activity, along with ease of crawling for the text content posted there. Text content is usually short stories which are submitted as a post, or in the case of *r/WritingPrompts* as top level comments.

Table 1 details the number of participants and articles that have been gathered through our sampling process. Active Reddit users engage with many articles, but engagement is not evenly distributed: some articles are consumed by only a single user, while other articles were consumed by hundreds of our participants. This leads to some imbalance and uncertainty in our average-personality labelling: it is possible that articles consumed by fewer of our participants give a single consumer's personality disproportionate weight.

Figure 1 shows the distribution of the Big-5 personality traits as aggregated by fiction and news articles as per our method described above. Personality labels at the article level show somewhat reduced variance compared to the per-user data (see Appendix C), but are by no means uniform. The personality distributions of news and fiction consumers are quite similar, seeming to reflect a common Reddit user personality type.

Figure 1: Box plots showing the distributions of personality scores per article. On each box, the central mark indicates the median, and the left and right edges of the box indicate the 25th and 75th percentiles, respectively. Outliers are plotted beyond the whiskers.



5 Results

In this work we primarily seek to answer the following questions:

- Can a person's personality inventory be inferred from the textual content they consume?
- Do state-of-the-art models for produced textual content achieve "good" performance when applied to consumed text?
- Does the domain of consumed text affect predictive performance in these tasks?

5.1 Message-Level Attention Results

Message-level attention is considered to achieve state-of-the-art performance when predicting personality based on text that is produced by users (Lynn et al., 2020; Eichstaedt et al., 2021). It may be reasonable to presume that these models would be good for the similar task of predicting personalities based on text consumed by users. In these experiments each article is passed through a Sentence-BERT language model (Reimers and Gurevych, 2019) to generate a vector representation for each article, which is then passed to the message-level attention model as described in Section 3.1. Appendix A provides more detail on our experimental setup for all models.

Three different models are trained; one using the fiction dataset, and two using the news dataset. The textual input for the two news dataset models differed, with one model trained using the article bodies in the same manner as for the fiction dataset, and one model using the new article titles alone. Fiction titles have not been considered as a textual input due to the format of titles in the chosen fiction subreddits; *r/WritingPrompts* titles are written as prompts for commenters to write their own fiction, while *r/shortstories* titles include authors, series tags, and other meta-text.

Figure 2: Histogram of all normalised message weights used in weighted sums to generate user vectors. A value being close to 1 represents an article that is weighed as important as it would be to a uniformly weighted mean. Higher weights represent more informative articles according to the message attention mechanism.

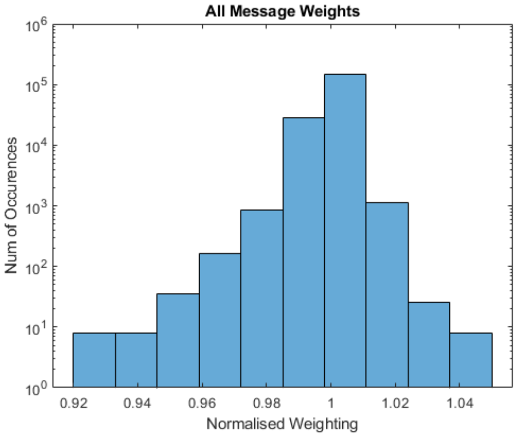


Table 2 shows the 5-fold cross-validation performance of all three models that are trained using message attention. The performance of all these models is degraded in comparison to the results obtained by Lynn et al. when using produced text (Lynn et al., 2020). The model trained on fiction articles produced the best performance of the three, and the lowest variance in performance.

Message attention models learn a weighting function that weighs the relative importance of each

Table 2: 5-fold cross-validation performance of message attention models attempting to predict personality inventories from consumed text. We show results for two different domains of text: fictional stories and news articles. We also show the performance of models trained just on news titles instead of the entire news article. We report the average Pearson's *r* across all 5 folds and intervals of one standard deviation.

Domain	Fiction	News	
Content	Articles	Articles	Titles
Extraversion	**0.08 [0, 0.16]**	0.05 [-0.06, 0.16]	0.06 [-0.06, 0.18]
Agreeableness	**0.1 [0.06, 0.14]**	0 [-0.2, 0.19]	0.01 [-0.11, 0.14]
Conscientiousness	0.07 [0.04, 0.11]	0.11 [-0.04, 0.26]	**0.13 [-0.05, 0.31]**
Neuroticism	**0.07 [-0.01, 0.16]**	0.04 [-0.05, 0.14]	0.01 [-0.14, 0.17]
Openness to Experience	0.04 [0, 0.09]	-0.02 [-0.2, 0.16]	**0.1 [-0.14, 0.33]**

article when generating the user vector. We can look at the distribution of these weights for each article to see if they are finding certain articles to be more informative than others.

We extract all message-level attention weights in order to examine the distribution. In the model these weights are used to create the user vector, with the weights contributing to the weighted sum of each article. If a user's personality is predicted given N arguments then a uniform weighting would be $1/N$ for each article. Each user may have consumed a different number of news articles, so to normalise this we divide each weight we obtain by $1/N$. If a normalised weight is < 1 then the model estimated that the article is less informative than average in predicting a user's personality. If a normalised weight is > 1 then the model has estimated that the article is more important to predicting that user's personality.

Figure 2 shows a histogram of all attention weights that are used in the validation steps of all three message attention models. All attention weights in these models are close to equally weighted for every article. This indicates that the model is predicting that none of these consumed articles are more or less informative than any other in predicting a user's personality.

5.2 Average Personality Per Article

For this experiment, all articles have a corresponding label that is the average personality score of all participants who have consumed the article. Our model of consumption is that an article has been consumed if the user has commented on a Reddit post that links to that article. In these experiments the language model used is the same as the previous models, with Sentence-BERT providing a vector representation for each article. The same input language model is used for a fair comparison between message attention and average personality models. The model is described in detail in Section 3.2.

Three models are trained again using the same fiction, news article and news title inputs as described in Section 5.1, this time with an average personality label for each article.

Table 3 shows the 5-fold cross-validation performance of all three models that are trained with averaged labels for each article. Model performance is much improved when compared to the results for the message attention approach. The variance within k-fold performance is also decreased, showing a more consistent performance between models. Models trained using news article text have generally better performance than models trained using fiction, with the exception of personality dimension of Openness. Our news titles model achieves similar performance to the model trained using the entire news article.

5.3 Visualisations

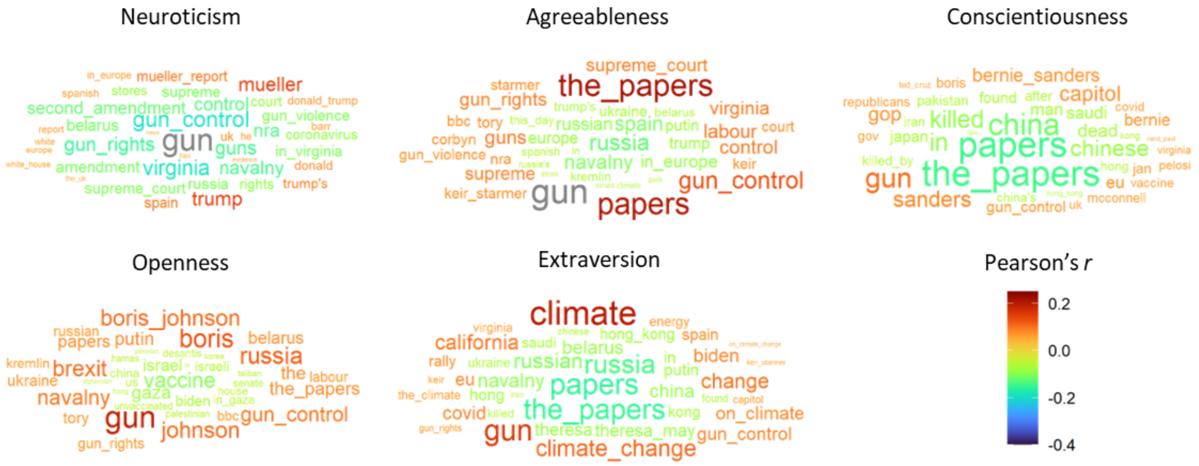
We generated word clouds to understand which words and phrases were most strongly correlated with each personality dimension. This was achieved by taking the validation set predictions from each fold and examining which n-gram phrases (1,2,3-gram) were most correlated with each personality dimension.

Figure 3 shows the word clouds for news titles when using the average-label method, and the n-grams that most correlate with each personality dimension. The word clouds show that words related to article content, rather than stylistic features, are most correlated with personality features of the text's consumer. The particular phrases visualised also represent major news stories that occurred during the period of data collection.

Table 3: 5-fold cross-validation performance of average-label models attempting to predict personality inventories. We show results for two different domains of text: fictional stories and news articles. We also show the performance of models trained just on news titles instead of the entire news article. We report the average Pearson's r across all 5 folds and and intervals of one standard deviation.

Domain	Fiction	News	
Content	Articles	Articles	Titles
Extraversion	0.29 [0.27, 0.30]	**0.3 [0.28, 0.31]**	0.27 [0.26, 0.28]
Agreeableness	0.12 [0.09, 0.14]	0.21 [0.2, 0.22]	**0.23 [0.22, 0.24]**
Conscientiousness	0.13 [0.09, 0.16]	**0.25 [0.23, 0.27]**	0.23 [0.21, 0.25]
Neuroticism	0.13 [0.11, 0.16]	0.33 [0.31, 0.35]	**0.34 [0.32, 0.35]**
Openness to Experience	**0.23 [0.2, 0.27]**	0.17 [0.15, 0.20]	0.15 [0.13, 0.17]

Figure 3: 1,2,3-gram word cloud of news titles. This shows top 20 positive and top 20 negative features that best predict each personality dimension. The colour of the n-grams indicate correlation to the dimension. The size of the n-gram is the absolute value of that correlation.



6 Discussion

Our experiments have tackled a novel personality prediction task using consumed rather than produced text to infer personality. We demonstrate that the structure of this task is important, showing a significant difference in performance between the state-of-the-art message attention model and average-labelling models. The message-level attention mechanism used by Lynn et al. (2020) does not perform well on this problem, and our examination of attention weights suggests this is because the model rarely finds any consumed text to be especially informative with regard to a participant's personality. We also find that the source of the corpus also has an impact on performance, with models trained upon news text better suited to predicting personality.

6.1 A failure of message attention

There may be multiple factors that explain the poor performance of message-level attention models on this task. The fundamental assumption of these models that is some pieces of text are more important than others when predicting personality. This assumption seems to hold true when using produced texts, given the performance of those models. However, this does not appear to have held for consumed text.

One factor that may be key to explaining this result is that produced text can usually be assumed to be somewhat unique. There may be some common questions or short phrases in reply to another user, but the majority of comments that are produced by a typical user would generally be unique text sequences. In contrast, news articles and pieces of fiction submitted to Reddit are intended to reach a wide audience, and so would take the role of common factors linked to the many unique personality profiles of all their consumers. Each article that is consumed by multiple people that have differing personality scores could confound the message attention mechanism, essentially providing the same input and expecting multiple different outputs.

The average-label approach to predicting the personality profile that consumes an article demonstrates encouraging predictive performance, accompanied by a reduction in variance between folds. These results instill confidence in the method's capability to infer the overall personality that a consumed article may elicit. Although our results may not match the outcomes achieved in other personality prediction tasks such as generated text, they serve as a solid foundation for further advancement.

Some consideration should be given to the difference between the tasks. Message attention models are modelling a user's personality given all of the text they have consumed. The average-label frame models the average personality of a single article. Are these similar enough tasks for a fair comparison of performance? Is it viable to use average label models as part of a model that would predict users?

6.2 Do we need more samples?

The different nature of the proposed models also leads to a large difference in the number of samples. We gathered 213 participants who consumed news articles, whereas we gathered 953 participants who read fiction. Contemporary work involving produced text generally has samples in the tens of thousands (Lynn et al., 2020; Eichstaedt et al., 2021).

To see if number of samples was the cause for the large gap in performance, we created a model that would predict personality from the produced text our participants posted on Reddit. We use the same message attention model as is used in previous work (Lynn et al., 2020). Our results (given in Appendix B) show performance much improved relative to that of our consumed text models, and with confidence intervals within range of state-of-the-art performance. This demonstrates that the number of samples alone does not explain the large decrease in performance between the produced and consumed text tasks.

6.3 Textual features

Average-label models also may be over-fitting to textual features that are repeated multiple times across each corpus. This may be particularly true with both article corpora. Efforts have been made to clean the text for repeating signals of this form (e.g., the author bylines for news articles) but we cannot be certain of removing all such indicators from our crawled article content. To mitigage this effect, we have trained models using L2 regularization. L2 regularization imposes a larger cost on the loss function for larger weights, thus decreasing the impact of over-fitting. Appendix D shows that L2 regularization on average-label models reduces performance, but these regularised models still outperform message-level attention models.

Our visualisations presented in Section 5.3 show how words and phrases correlate with personality dimensions. Content is picked up rather than writing style when looking at the word clouds, suggesting that consumption of particular topics may be more indicative of personality than the style in which the content is presented. These results may be seen as consistent with similar works involving user generated content and personality. Facebook likes of topics and media content have also been found to be congruent with personality (Youyou et al., 2015). Our visualisations of the news dataset also show that the model is correlating predictions with certain news topics dominant at the time of data collection. This may be an artefact of the small time period of data collection from users; while all articles that participants have consumed have been crawled, their activity is more likely to contain recent content.

N-grams that appear to be predictive of a high neuroticism score (such as 'gun control') have an inverse correlation with the other four dimensions. This is consistent with theoretical and other quantitative research into the general factors of personality, and the broader interrelation between those four dimensions when contrasted with neuroticism (Van der Linden et al., 2010; Musek, 2007).

6.4 Further pointers

News content in general appears to out-perform fictional content when used as a predictor of personality. Three personality dimensions appear to be less predictable from fictional content than from news, while extraversion remains predictable with good performance across all three datasets. Openness is however easier to predict when using fictional content as an input. Behavioural research may reveal if these patterns exist outside of these models.

To model consumption from observable posting behaviour, we assumed that if a participant had commented on a Reddit thread, that participant had read the article which began the thread. We cannot say with certainty that this is true, and especially cannot be confident that a user has read any specific part of an article, as commenting without reading is an unfortunately common behaviour on many social media platforms. To explore this, we compared predictions of personalities using news article text and just the title of the article (which is the first thing a user will see on entering a thread), finding that predictions using the titles alone were often as good as (and for some personality dimensions, better than) using the full article text. We tentatively conclude that when making predictions on the basis of text consumption, some scepticism may be warranted as to whether a user has fully consumed a given text.

7 Conclusions

In this paper we have shown that personality can be inferred based on the text that a user has consumed. To our knowledge, this is the first work using consumed textual content to model personality that reaches comparable performances to produced content. The performance achieved by average-label modelling can be seen as a baseline for personality modelling using consumed text.

Message attention models do not achieve acceptable levels of performance when applied to the domain of consumed text. We show that this may be due to the weighting function giving no especial weight to any consumed text, in combination with the lack of unique textual content for each user, which gives confounding feedback to the model during training.

We used three different datasets to train and evaluate our models: pieces of fiction, news articles, and news titles. Personality is shown to be more reliably inferred from news content than fiction content. Models trained upon news article titles, with less textual content, achieved similar performance to models trained upon whole news articles, which may reveal that a condensed set of features are most important for modelling personality.

Future work in this field should involve further investigation as to how message attention models may be adapted to this context, as well as establishing resources to enable new approaches to this problem in the form of a shared task. Due to participant privacy concerns, our datasets cannot be released, which forms a hurdle to reproduction and development. A publicly available dataset would be beneficial, so new work can be evaluated on a standardised dataset. An ideal dataset would also provide access to more training samples, along with greater assurance that the textual content has been consumed by the users.

Acknowledgements

This research was supported by funding from REPHRAIN: National Research Centre on Privacy, Harm Reduction and Adversarial Influence online (UKRI grant: `EP/V011189/1`). The authors were also supported by a large grant from the Volkswagen Foundation (“Reclaiming individual autonomy and democratic discourse online”). S.L. was also supported by funding from the Humboldt Foundation in Germany and by an ERC Advanced Grant (PRODEMINFO).

References

Yoram Bachrach, Michal Kosinski, Thore Graepel, Pushmeet Kohli, and David Stillwell. 2012. Personality and patterns of Facebook usage. In *Proceedings of the 4th Annual ACM Web Science Conference*, pages 24–32.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Johannes C Eichstaedt, Margaret L Kern, David B Yaden, HA Schwartz, Salvatore Giorgi, Gregory Park, Courtney A Hagan, Victoria A Tobolsky, Laura K Smith, Anneke Buffone, et al. 2021. Closed-and open-vocabulary approaches to text analysis: A review, quantitative comparison, and recommendations. *Psychological Methods*, 26(4):398.

Lewis R Goldberg. 1993. The structure of phenotypic personality traits. *American Psychologist*, 48(1):26.

Matthew H Goldberg, Abel Gustafson, Seth A Rosenthal, and Anthony Leiserowitz. 2021. Shifting republican views on climate change through targeted advertising. *Nature Climate Change*, 11(7):573–577.

Keven Joyal-Desmarais, Alexandra K Scharmer, Molly K Madzelan, Jolene V See, Alexander J Rothman, and Mark Snyder. 2022. Appealing to motivation to change attitudes, intentions, and behavior: A systematic review and meta-analysis of 702 experimental tests of the effects of motivational message matching on persuasion. *Psychological Bulletin*, 148(7-8):465.

Veronica Lynn, Niranjan Balasubramanian, and H Andrew Schwartz. 2020. Hierarchical modeling for user personality prediction: The role of message-level attention. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5306–5316.

S C Matz, M Kosinski, G Nave, and D J Stillwell. 2017. Psychological targeting as an effective approach to digital mass persuasion. *Proceedings of the National Academy of Sciences*, 114(48):12714–12719.

Janek Musek. 2007. A general factor of personality: Evidence for the Big One in the five-factor model. *Journal of Research in Personality*, 41(6):1213–1233.

Michael Nowak and Dean Eckles. 2014. Determining user personality characteristics from social networking system communications and characteristics. US Patent 8,825,764.

Barun Patra, Saksham Singhal, Shaohan Huang, Zewen Chi, Li Dong, Furu Wei, Vishrav Chaudhary, and Xia Song. 2022. Beyond English-centric bitexts for better multilingual language representation learning. *arXiv preprint arXiv:2210.14867*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. *arXiv preprint arXiv:1908.10084*.

H Andrew Schwartz, Johannes C Eichstaedt, Margaret L Kern, Lukasz Dziurzynski, Stephanie M Ramones, Megha Agrawal, Achal Shah, Michal Kosinski, David Stillwell, Martin EP Seligman, et al. 2013. Personality, gender, and age in the language of social media: The open-vocabulary approach. *PloS ONE*, 8(9):e73791.

Christopher J Soto and Oliver P John. 2017. The next big five inventory (BFI-2): Developing and assessing a hierarchical model with 15 facets to enhance bandwidth, fidelity, and predictive power. *Journal of Personality and Social Psychology*, 113(1):117–143.

Ben M Tappin, Chloe Wittenberg, Luke Hewitt, David Rand, et al. 2022. Quantifying the persuasive returns to political microtargeting. *(Working Paper)*.

Dimitri Van der Linden, Jan te Nijenhuis, and Arnold B Bakker. 2010. The general factor of personality: A meta-analysis of Big Five intercorrelations and a criterion-related validity study. *Journal of Research in Personality*, 44(3):315–327.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Wu Youyou, Michal Kosinski, and David Stillwell. 2015. Computer-based personality judgments are more accurate than those made by humans. *Proceedings of the National Academy of Sciences*, 112(4):1036–1040.

Brahim Zarouali, Tom Dobber, Guy De Pauw, and Claes de Vreese. 2020. Using a personality-profiling algorithm to investigate political microtargeting: Assessing the persuasion effects of personality-tailored ads on social media. *Communication Research*, 49(8):1066–1091.

Qihuang Zhong, Liang Ding, Yibing Zhan, Yu Qiao, Yonggang Wen, Li Shen, Juhua Liu, Baosheng Yu, Bo Du, Yixin Chen, et al. 2022. Toward efficient language model pretraining and downstream adaptation via self-evolution: A case study on SuperGLUE. *arXiv preprint arXiv:2212.01853*.

Appendix

A Experimental Setup

All models were implemented using PyTorch and PyTorch Lightning. One model was trained for each personality dimension. Language modelling was performed using the 'sentence-transformers/all-distilroberta-v1' Sentence-BERT model, which provides a 768-dimensional representation for each piece of text. The learning rate for each model was selected using PyTorch Lightning. Training lasted for 8 epochs, although most stopped after 4 epochs, due to early stopping. Samples were uniformly sampled and split into 5 folds.

A.1 Users

The batch size was set to 1, due to hardware constraints. The maximum number of articles per user was also limited to 100. 512 input tokens were taken from each article. If an article was shorter than 512 tokens, it would be padded. If the article was longer, it would be truncated to the first 512 tokens. All articles were passed to the language model, and messages were split between 5 Titan-X GPUs for parallel computation.

After generating the embeddings, mean pooling was applied to the token embeddings to obtain sentence vectors. These sentence vectors were then processed through a multi-headed attention mechanism, followed by dot product attention on the outputs. This produced scalar values corresponding to each message. These scalars were used as weights for the weighted sum of the token embeddings. Finally, two feed forward layers used the user vector to generate the personality prediction.

A.2 Articles

The batch size for articles was set to 16. Since the model is relatively small, only a single GPU was required for processing. Each article was represented with 512 tokens and undergoes the same truncation or padding process as the users' articles. Once the token embeddings were computed, sentence vectors were generated using mean pooling. These sentence vectors were then passed through two feed-forward layers to generate a personality prediction. When training models with L2 regularization (as specified in Appendix D) the weight decay parameter (λ) was set to 0.001.

A.3 Titles

The batch size for titles was also set to 16 to ensure comparable training with models that use articles as input text. However, each title was limited to 128 tokens in length. The titles are padded or truncated as necessary during pre-processing. As with articles, when doing L2 regularization the weight decay was set to 0.001.

B Produced Text Models

Models trained from produced text have the same model as titles, where the length of the text is limited to 128 tokens due to comments being shorter. Approximately 10,000 comments were used in training these models.

Table 4: 5-fold cross-validated prediction performance when using message attention to predict users' personality scores from the text they have produced. The dataset used here is sampled from the same 1,116 participants used in our consumed text models, but with predictions made using text they produced via their comments. We report the average performance across each fold as well as 95% confidence intervals.

Personality Dimension	Pearson's r [95% CI]
Extraversion	0.32 [0.22, 0.43]
Agreeableness	0.31 [0.20, 0.42]
Conscientiousness	0.33 [0.27, 0.38]
Neuroticism	0.33 [0.21, 0.45]
Openness to Experience	0.32 [0.20, 0.44]

Table 4 shows the 5-fold cross-validation performance of a message-level attention model, using our participants' produced text to predict their personality inventories rather than the text they have consumed. The model used in this experiment is the same as the model described in Section 3.1, which achieved underwhelming performance when using consumed text.

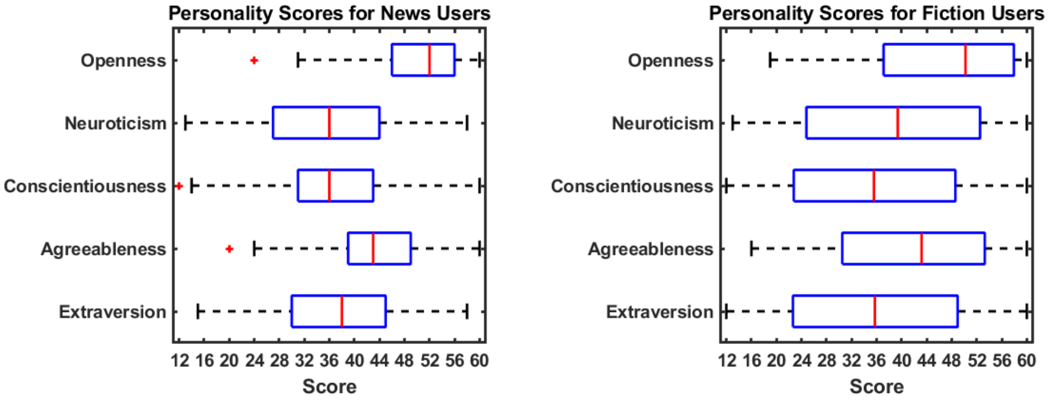
This level of performance more closely resembles state-of-the-art metrics that have been documented when using produced text for personality prediction, with state of the art performance within the confidence intervals for each dimension (Eichstaedt et al., 2021).

This shows that message-level attention models can perform well with a lower number of samples when using produced text, however consumed text may not be an ideal medium for this model architecture.

Table 5: 5-fold cross-validation performance of average-label models attempting to predict personality inventories when using L2 regularization. These experiments were intended to probe whether over-fitting is evident in our average-label models. Bold indicates which models performed the best for each dimension.

Domain	Fiction	News	
Content	Articles	Articles	Titles
Extraversion	0.17 [0.15, 0.2]	0.23 [0.21, 0.25]	**0.28 [0.27, 0.29]**
Agreeableness	0.1 [0.05, 0.14]	0.17 [0.15, 0.18]	**0.21 [0.19, 0.22]**
Conscientiousness	0.11 [0.08, 0.14]	**0.2 [0.19, 0.21]**	**0.2 [0.19, 0.22]**
Neuroticism	0.08 [0.06, 0.1]	0.29 [0.28, 0.31]	**0.31 [0.3, 0.32]**
Openness to Experience	0.09 [0.08, 0.1]	**0.12 [0.12, 0.13]**	**0.12 [0.11, 0.13]**

Figure 4: Box plots showing the distributions of personality scores per user. On each box, the central mark indicates the median, and the left and right edges of the box indicate the 25th and 75th percentiles, respectively. Outliers are plotted beyond the whiskers.



C User Personality Distributions

Figure 4 shows the distribution of the Big-5 personality traits for fiction and news consumers amongst our participants. Figure 1 from the main body displays scores as aggregated on a per-article basis.

D Regularized Models

Table 5 presents the 5-fold cross-validation performance of average-label models when using L2 regularization. The decreases in performance may be explained by over-fitting in the original models without L2 regularization.

The models using the news title dataset are generally now the better-performing models and also see the lowest performance impact from regularisation. This may indicate that our news article representations contain noisy features as a byproduct of crawling, and models without regularization over-fit to those features.

UNIDECOR:
A Unified Deception Corpus for Cross-Corpus Deception Detection

Aswathy Velutharambath[1,2] and **Roman Klinger**[1]
[1]Institut für Maschinelle Sprachverarbeitung, University of Stuttgart, Germany
[2]100 Worte Sprachanalyse GmbH, Heilbronn, Germany
`aswathy.velutharambath@100Worte.de`
`roman.klinger@ims.uni-stuttgart.de`

Abstract

Verbal deception has been studied in psychology, forensics, and computational linguistics for a variety of reasons, like understanding behaviour patterns, identifying false testimonies, and detecting deception in online communication. Varying motivations across research fields lead to differences in the domain choices to study and in the conceptualization of deception, making it hard to compare models and build robust deception detection systems for a given language. With this paper, we improve this situation by surveying available English deception datasets which include domains like social media reviews, court testimonials, opinion statements on specific topics, and deceptive dialogues from online strategy games. We consolidate these datasets into a single unified corpus. Based on this resource, we conduct a correlation analysis of linguistic cues of deception across datasets to understand the differences and perform cross-corpus modeling experiments which show that a cross-domain generalization is challenging to achieve. The unified deception corpus (UNIDECOR) can be obtained from `https://www.ims.uni-stuttgart.de/data/unidecor`.

1 Introduction

Deception detection has remained an area of vested interest in fields like psychology, forensics, law, and computational linguistics for a myriad of reasons like understanding behavioral patterns of lying (Newman et al., 2003; DePaulo and Morris, 2004), identifying fabricated information (Conroy et al., 2015), distinguishing false statements or testimonies (Şen et al., 2022) and detecting deception in online communication (Hancock, 2009). These are relevant tasks because of the truth bias, which is the inherent inclination of humans to actively believe or passively presume that a statement made by another person is true and accurate by default, without the need for evidence to substantiate this belief (Levine, 2014). While this facilitates efficient communication, it also makes people susceptible to deception, especially in online media where digital deception (Hancock, 2009) manifests in many forms like fake news, misleading advertisements, impersonation and scams. This warrants automatic deception detection systems that can accurately distinguish between truthful and deceptive discourse solely from textual data.

The task of automatic deception detection comes with several challenges. Deception or lying is a complex human behavior and its signals are faint in text. Moreover, it is sensitive to the communication context, interlocutors, and the stake involved (Ten Brinke and Porter, 2012; Salvetti et al., 2016). Most importantly, acquiring annotated data proves to be one of the major hurdles for deception studies. Traditional data annotation methods cannot be employed because human performance is shown to be worse than machines in differentiating truths and lies (Bond Jr. and DePaulo, 2006; Vrij, 2014). One way to collect accurate data is to get the labels at source by the person producing the text. Alternatively, they can be collected using the acquired knowledge that certain types of contents are deceptive. Across the literature, different strategies like crawling fake reviews (Yao et al., 2017), collecting text from users identified as suspicious (Fornaciari et al., 2020), using non-linguistic deception cues (Fornaciari and Poesio, 2014) and soliciting through crowd-sourcing (Ott et al., 2011, 2013; Salvetti et al., 2016) have been employed to obtain reliable annotations.

The diversity in the domains of interest, the medium of deceptive communication (spoken vs. written) and dataset creation procedures make it difficult to compare cues of deception across datasets and to understand their generalizability across different domains. With this study, we aim at mitigating this situation by conducting a comparative survey of publicly available textual deception datasets.

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 39–51
July 14, 2023 ©2023 Association for Computational Linguistics

We contribute (1) a consolidated corpus in a unified format and (2) conduct experiments in which we evaluate models trained on one data set on all others. Our (3) results show that cross-corpus, particularly cross-domain, generalizability is limited, which motivates future work to develop robust deception detectors. We lay the foundation for such work with (4) additional correlation analyses of the linguistic cues of deception across datasets and verify their generalizability across domains.

2 Background & Related Work

Deception in communication is the act of intentionally causing another person to have a false belief that the deceiver knows or believes to be false (Zuckerman et al., 1981; Mahon, 2007; Hancock, 2009). Lies, exaggerations, omissions, and distortions are all different forms of deception (Turner et al., 1975; Metts, 1989). While the definition of deception varies across literature, they concur that it is intentional or deliberate (Mahon, 2007; Gupta et al., 2013).

2.1 Domains and Ground Truth

Deception research is spread across disciplines which contributed to a variety of domains and consequentially to a number of data collection methods. Domains include opinions statements on a specific topic (Pérez-Rosas and Mihalcea, 2014; Capuozzo et al., 2020; Lloyd et al., 2019), open domain statements (Pérez-Rosas and Mihalcea, 2015), online reviews (Ott et al., 2011, 2013; Fornaciari and Poesio, 2014; Yao et al., 2017), deceptive dialogues in strategic games like Mafiascum[1], Box of Lies and Diplomacy (de Ruiter and Kachergis, 2018; Soldner et al., 2019; Peskov et al., 2020; Skalicky et al., 2020) and court trials (Şen et al., 2022).

The ground truth generation strategies differ across datasets. While datasets of opinion statements (Pérez-Rosas and Mihalcea, 2014; Capuozzo et al., 2020; Lloyd et al., 2019), and online reviews (Ott et al., 2011, 2013; Fornaciari and Poesio, 2014; Yao et al., 2017) are collected in written form, interviews include both verbal and non-verbal content (Şen et al., 2022). Game-based corpora contain monologue (Skalicky et al., 2020) or dialogue data (Soldner et al., 2019; Peskov et al., 2020).

All of these resources contain instances that are labeled as truthful or deceptive. Only few studies employ the same procedure to generate both truthful and deceptive content (Salvetti et al., 2016; Skalicky et al., 2020); most resort to separate strategies for collecting them (Ott et al., 2011, 2013; Fornaciari et al., 2020). Instances labeled as deceptive are either solicited content or collected from a source identified as deceptive. Ott et al. (2011, 2013) crawled the truthful reviews from websites of interest and the deceptive ones were crowd-sourced through AMT[2], while Salvetti et al. (2016) solicited both via AMT. Yao et al. (2017) tracked fake review generation tasks from crowd-sourcing platforms to identify deceptive reviews and reviewers. For the datasets based on strategic games, the labels are assigned based on game rules. Opinion domain datasets contain stances on topics, like gay marriage and abortion, written by the same person, where the truthful labeled opinions align with the author's true opinion and deceptive ones align with the opposite (Pérez-Rosas and Mihalcea, 2014; Capuozzo et al., 2020).

2.2 Automatic Deception Detection Methods

Several studies have explored the effectiveness of automatic methods to detect deception from textual data. These include feature-based classification methods with support vector machines (Ott et al., 2011; Pérez-Rosas and Mihalcea, 2014; Fornaciari and Poesio, 2014), logistic regression (de Ruiter and Kachergis, 2018), decision trees (Pérez-Rosas and Mihalcea, 2015), and random forests (Soldner et al., 2019; Pérez-Rosas and Mihalcea, 2015). Some studies also consider contextual information by using recurrent neural networks (Peskov et al., 2020) and transformer-based models (Capuozzo et al., 2020; Peskov et al., 2020; Fornaciari et al., 2021). Transformers are not always better – Peskov et al. (2020) show that BERT is en par with LSTMs while Fornaciari et al. (2021) showed that adding extra attention layers help to improve upon the previous state of the art.

Most works focused on modeling the concept of deception in one domain. An exception is Hernández-Castañeda et al. (2016) who report cross-domain classification results on OpSpam, DeRev2014, and CrossCultDe, but in an all-against-one setting, not in a one-against-one setup.

2.3 Linguistic Cues of Deception

To understand the phenomenon of deception better, previous studies have analyzed the linguistic

---

[1] `https://www.mafiascum.net/`

[2] Amazon's Mechanical Turk, `https://www.mturk.com/`

cues that characterize deceptive language in written statements, spoken conversations, and online communication (Newman et al., 2003; Bond and Lee, 2005) and demonstrated that a systematic analysis of these cues can prove valuable in automated deception detection specifically in computer-mediated communication (Zhou et al., 2004). Newman et al. (2003) noted that the use of fewer self-references in deceptive statements indicate that the liars are attempting to distance themselves from the lies. The use of exclusive words (e.g., *but*, *rather*) allow deceivers to introduce communicative ambiguity into the discourse. Hancock et al. (2007) noted that these cues are broadly associated with the number of words, use of pronouns, use of emotion words, and presence of markers of cognitive complexity. They also pointed out that these cues can manifest differently based on the type and medium of discourse; real-world vs. online or monologue vs. dialogue.

While these analyses have found application in machine learning models, there are more sets of features that have been used to automatically detect deception. These include n-grams (Fornaciari and Poesio, 2014; Fornaciari et al., 2020; Ott et al., 2011), part-of-speech tags (Lloyd et al., 2019; Fornaciari et al., 2020; Pérez-Rosas and Mihalcea, 2015), lexicon-based features, including the Linguistic Inquiry and Word Count (LIWC, Pennebaker et al., 2015) psychological categories, (Pérez-Rosas and Mihalcea, 2014; Yao et al., 2017) and production rules derived from syntactic context free grammar trees (Yao et al., 2017; Pérez-Rosas and Mihalcea, 2015). Duran et al. (2010), Swol et al. (2012) and Hauch et al. (2015) conducted extensive surveys and analyses of different linguistic cues of deception.

3 Unified Deception Dataset

As preparation for cross-corpus analysis of the concept of deception, we consolidate publicly available textual deception datasets into a unified format.[3] We now describe the included datasets.

Deceptive Opinion Spam (OPSPAM). Ott et al. (2011) describes *deceptive opinion spam* as fraudulent reviews written to sound authentic with the goal to deceive the reader. To study the nature of such reviews, they collected truthful reviews by crawling online review platforms like TripAdvisor[4] and crowd-sourced deceptive reviews via Amazon's Mechanical Turk (AMT). The initial OPSPAM dataset published by Ott et al. (2011) contains 400 truthful and 400 deceptive reviews with positive sentiments. Ott et al. (2013) extended the dataset to include reviews with negative sentiments. The complete OPSPAM dataset contains 1600 instances labeled for veracity and sentiment. It is available publicly with a Creative Commons Attribution-NonCommercial-ShareAlike license.[5]

Cross-cultural Deception (CROSSCULTDE). Pérez-Rosas and Mihalcea (2014) collected the CROSSCULTDE dataset to investigate deception in a cross-cultural setting. It consists of short essays on the topics of abortion, death penalty, and feelings about a best friend, collected from the United States, India, and Mexico. We take into account the data collected from the United States and India which are in English and consist of 100 deceptive and 100 truthful essays per topic per geographical region adding up to 1200 labeled instances. The dataset is available for download without mentioning any usage restrictions.[6]

Deception in Reviews (DEREV2014/2018). To investigate the phenomenon of sock puppetry, Fornaciari and Poesio (2014) collected DEREV2014, containing book reviews from *amazon.com* that were identified as authentic or fake using predefined linguistic cues. To overcome the shortcoming that these cues cannot be used while developing a deception classifier, Fornaciari et al. (2020) released the DEREV2018 dataset, in which they collect deceptive reviews based on *a priori* knowledge about authors who solicited fake reviews. Additionally, the authors crowd-sourced both truthful and deceptive reviews for the same books. The DEREV2014 dataset contains 118 reviews each with a truthful label and a deceptive label, while the DEREV2018 dataset includes 1552 reviews each collected from *amazon.com* and through crowd-sourcing with a balanced distribution of truthful and deceptive reviews. The datasets overlap by 62 reviews. Both corpora are available for download.[7]

Open Domain Deception (OPENDOMAIN). Pérez-Rosas and Mihalcea (2015) study deception, gender, and age detection with an open domain

---

[3] We refer to our corpus as UNIDECOR: "Unified Deception Corpus". The scripts to download and convert the dataset can be found in the following repository: `https://www.ims.uni-stuttgart.de/data/unidecor`

[4] `https://www.tripadvisor.com/`

[5] `https://myleott.com/op-spam.html`

[6] `https://web.eecs.umich.edu/~mihalcea/downloads.html`

[7] `https://fornaciari.netlify.app/`

Dataset	Domain	Truthful	Deceptive	Total	TC	SC
Bluff the listener (Bluff)	game	251 (33.3%)	502 (66.7%)	753	241.66	11.5
Diplomacy dataset (Diplomacy)	game	16402 (94.9%)	887 (5.1%)	17289	24.53	1.7
Mafiascum dataset (Mafiascum)	game	7439 (76.9%)	2237 (23.1%)	9676	4690.69	362.8
Multimodal Decep. in Dialogues (BoxOfLies)	game	101 (20.2%)	400 (79.8%)	501	12.2	1.6
Miami University Decep. Detection Db. (MU3D)	interview	160 (50.0%)	160 (50.0%)	320	131.7	5.7
Real-life trial data (Trial)	interview	60 (49.6%)	61 (50.4%)	121	79.85	3.9
Cross-cultural deception (CrossCultDe)	opinion	600 (50.0%)	600 (50.0%)	1200	80.0	4.5
Deceptive Opinion (DecOp)	opinion	1250 (50.0%)	1250 (50.0%)	2500	65.56	4.0
Boulder Lies and Truth Corpus (BLTC)	review	1041 (69.8%)	451 (30.2%)	1492	116.92	6.5
Deception in reviews (DeRev2014)	review	118 (50.0%)	118 (50.0%)	236	145.22	6.7
Deception in reviews (DeRev2018)	review	1552 (50.0%)	1552 (50.0%)	3104	176.6	8.1
Deceptive opinion spam (OpSpam)	review	800 (50.0%)	800 (50.0%)	1600	170.5	9.5
Online deceptive reviews (OnlineDe)	review	101431 (85.9%)	16694 (14.1%)	118125	171.5	7.2
Open Domain Deception (OpenDomain)	statement	3584 (50.0%)	3584 (50.0%)	7168	9.33	1.0
		134789 (82.1%)	29296 (17.9%)	164085	436.88	31.05

Table 1: Datasets included in our unified corpus (UniDecor), together with statistical information. TC: average token count; SC: average sentence count.

dataset acquired via AMT. Workers were asked to contribute seven true and seven plausible deceptive statements without a restriction of domain, each in a single sentence. The balanced dataset consists of 7168 annotated instances with additional demographic information. The data set is made available without specifying usage restrictions.[6]

Real-life Trial Data (Trial). To study real-life high-stake deception scenarios, Pérez-Rosas et al. (2015) collected videos of trial hearings from publicly available sources like "The Innocence Project" website[8]. The dataset contains multimodal information with annotations for non-verbal behavior like facial displays and gestures in addition to crowd-sourced transcriptions. It contains 60 truthful and 61 deceptive reviews.This corpus is made available without specifying any usage restrictions.[6]

Boulder Lies and Truth Corpus (BLTC). Salvetti et al. (2016) built a balanced dataset containing reviews elicited via AMT for the domains of electronic appliances and hotels. The crowdworkers were instructed to write fake or real reviews, with positive or negative sentiment, about objects that they were familiar with or not. Unlike other datasets which limited the labeling to truthful vs. deceptive, this dataset distinguished between fake and deceptive reviews, where the former are fabricated opinions about an unknown object while the latter was a false review of a known object. The corpus contains 1492 reviews, out of which 451 are truthful and the rest is labeled as fake or deceptive. It is available through the LDC.[9]

Online Deceptive Reviews (OnlineDe). To address the bottleneck that large realistic data for deception detection do not exist, Yao et al. (2017) created the OnlineDe corpus containing manipulated reviews posted online. They employed the automatic deception detection framework outlined by Fayazi et al. (2015) to identify deceptive reviewers and reviews from social media manipulation campaigns. It contains more than 100K labeled reviews with ≈10000 deceptive instances, covering more than 30 domains. The dataset is available for research purposes from the authors.

Mafiascum Dataset (Mafiascum). This dataset published by de Ruiter and Kachergis (2018) contains a collection of more than 700 games of Mafia, an online strategy game played on the Internet forum Mafiascum[10]. Here, players are assigned deceptive or non-deceptive roles randomly, which serve as annotations of the instances. Each of the 9000 documents contain all messages written by a single user in a specific game. The average token count in the instances (4690.69) is therefore considerably higher than in other corpora. The authors have made the dataset publicly available along with the code used for analyses.[11]

Miami University Deception Detection Database (MU3D). To investigate the role of gender and race in deception studies, Lloyd et al. (2019) created MU3D. It is a collection of interview videos where participants were instructed to talk truthfully or deceptively about their relationship with a person

---

[8] `http://www.innocenceproject.org/`

[9] Linguistic Data Consortium, `https://catalog.ldc.upenn.edu/LDC2014T24`

[10] `https://www.mafiascum.net/`

[11] `https://bitbucket.org/bopjesvla/thesis/src/master/`

whom they liked or disliked. The 80 participants, each belonging to a different gender and ethical background contributed to a positive truth, a negative truth, a positive lie and a negative lie, counting to 160 truthful and 160 deceptive interview content. The transcriptions of these videos along with demographic information, valency, and veracity annotations are made available for research purposes with a Creative Commons Attribution-NonCommercial-NoDerivs license.[12]

Multimodal Deception in Dialogues (BoxOfLies). To explore deception in conversational dialogue, Soldner et al. (2019) collected the BoxOfLies dataset which is based on the "Box of Lies" game, a segment on "The Tonight Show Starring Jimmy Fallon" where two celebrity guests take turns describing the contents of a box but are allowed to lie. The opposing player must decide if they believe the description or not. The collected dataset contained 25 videos of the game, transcribed and annotated for non-verbal cues of deception and the veracity of the describer. We exported the statements containing veracity label from the dataset using ELAN[13], a tool used to create and modify annotations for audio and video data. The dataset is available for download without specifying any usage restrictions.[6]

Diplomacy Dataset (Diplomacy). To study deception in a conversational context specifically in long-lasting relationships, Peskov et al. (2020) employed the negotiation-based online game Diplomacy. The players use deception as a strategy to convince other players to form alliances, for which they use a chat interface. Contrary to other deception datasets, Diplomacy contains an additional label for perceived truthfulness of an instance. The intended and perceived truthfulness of each message was annotated by the sender and the receiver respectively. Out of more than 13k messages less than 5% are labeled as intended or perceived lie, resulting in an imbalanced dataset. We use the dataset made available through ConvoKit.[14]

Deceptive Opinion (DecOp). To study deception in multi-domain and multi-lingual settings, Capuozzo et al. (2020), following the method described by Pérez-Rosas and Mihalcea (2014), collected truthful and deceptive opinion statements on five different topics, namely abortion, cannabis legalization, euthanasia, gay marriage, and policies on migrants. The experiment was conducted for English and Italian, from which we include the English instances in UniDecor. They consist of 2500 opinions statements with balanced labels. This dataset can be obtained from the authors.

Bluff the Listener (Bluff). To study humorous deception with no malicious intent, Skalicky et al. (2020) compiled the Bluff dataset. It contains data from the "Bluff the Listener" game which is part of the radio show "Wait…Don't Tell Me". It is a variation of the game "Two Truths and a Lie" in which a panelist tells three stories, two of which are true, and one of which is false. This corpus published by Skalicky et al. (2020) contains 753 humorous stories collected from 251 episodes broadcast from 2010 to 2019. The authors downloaded the transcripts from News-Bank[15], a curated repository containing current and archived media. One-third of the stories are truthful while two-thirds are fabricated, counting to 251 truthful and 502 deceptive stories. The dataset is publicly available and can be downloaded via the OSF platform.[16]

Aggregation. We consolidate the datasets into one unified corpus in which each instance is assigned a binary label indicating if it is truthful or deceptive. We retain annotation dimensions that are available for more than one dataset (age, gender, country, and sentiment). More details on the aggregation process and a sample entry from the corpus are available in Appendix A. Table 1 provides an overview of the corpora, including size, label distribution, token and sentence counts[17], along with the domain. The datasets vary greatly in its size, but the distribution of labels is mostly comparable, except for BLTC, OnlineDe and Diplomacy with comparably high counts for truthful instances.

4 Similarity Analysis

The datasets included in UniDecor come from a variety of domains and differ markedly in terms of the method of collection. At the same time, datasets from the same domains also have differences (e.g., solicited reviews vs. actual reviews). To understand the differences of datasets better, we explore the similarity values between these datasets

---

[12] `https://sc.lib.miamioh.edu/handle/2374.MIA/6067`

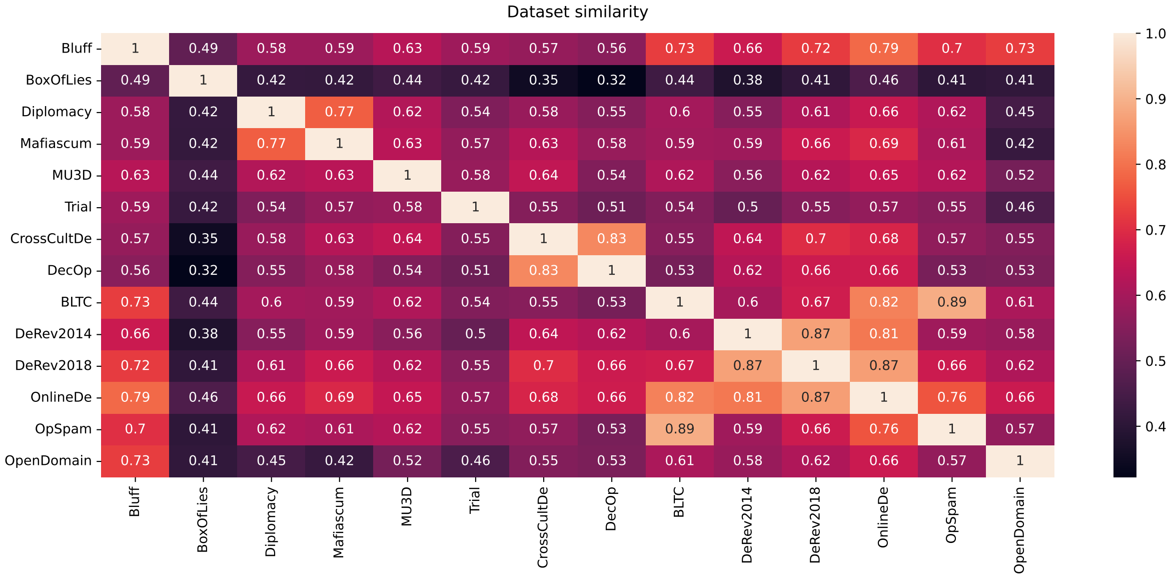
[13] `https://tla.mpi.nl/tools/tla-tools/elan/download/`

[14] `https://convokit.cornell.edu/documentation/diplomacy.html`

[15] `www.newsbank.com`

[16] `https://osf.io/download/mupd9`

[17] Using NLTK's wordpunct_tokenize and sent_tokenize



	Bluff	BoxOfLies	Diplomacy	Mafiascum	MU3D	Trial	CrossCultDe	DecOp	BLTC	DeRev2014	DeRev2018	OnlineDe	OpSpam	OpenDomain
Bluff	1	0.49	0.58	0.59	0.63	0.59	0.57	0.56	0.73	0.66	0.72	0.79	0.7	0.73
BoxOfLies	0.49	1	0.42	0.42	0.44	0.42	0.35	0.32	0.44	0.38	0.41	0.46	0.41	0.41
Diplomacy	0.58	0.42	1	0.77	0.62	0.54	0.58	0.55	0.6	0.55	0.61	0.66	0.62	0.45
Mafiascum	0.59	0.42	0.77	1	0.63	0.57	0.63	0.58	0.59	0.59	0.66	0.69	0.61	0.42
MU3D	0.63	0.44	0.62	0.63	1	0.58	0.64	0.54	0.62	0.56	0.62	0.65	0.62	0.52
Trial	0.59	0.42	0.54	0.57	0.58	1	0.55	0.51	0.54	0.5	0.55	0.57	0.55	0.46
CrossCultDe	0.57	0.35	0.58	0.63	0.64	0.55	1	0.83	0.55	0.64	0.7	0.68	0.57	0.55
DecOp	0.56	0.32	0.55	0.58	0.54	0.51	0.83	1	0.53	0.62	0.66	0.66	0.53	0.53
BLTC	0.73	0.44	0.6	0.59	0.62	0.54	0.55	0.53	1	0.6	0.67	0.82	0.89	0.61
DeRev2014	0.66	0.38	0.55	0.59	0.56	0.5	0.64	0.62	0.6	1	0.87	0.81	0.59	0.58
DeRev2018	0.72	0.41	0.61	0.66	0.62	0.55	0.7	0.66	0.67	0.87	1	0.87	0.66	0.62
OnlineDe	0.79	0.46	0.66	0.69	0.65	0.57	0.68	0.66	0.82	0.81	0.87	1	0.76	0.66
OpSpam	0.7	0.41	0.62	0.61	0.62	0.55	0.57	0.53	0.89	0.59	0.66	0.76	1	0.57
OpenDomain	0.73	0.41	0.45	0.42	0.52	0.46	0.55	0.53	0.61	0.58	0.62	0.66	0.57	1

Figure 1: Similarity values, according to the measure proposed by Li and Dunn (2022), between all pairs of datasets.

using the corpus-similarity measure defined by Li and Dunn (2022), which uses word unigram frequencies and character trigram frequencies of the datasets to calculate the Spearman's $\rho \in [-1; 1]$.[18]

Figure 1 shows a symmetrical matrix of similarity scores for dataset pairs. The correlation values could in principle be negative, but we do not observe any such values because all corpora are in the same language and have a high degree of term and character frequency overlap.

The heatmap reflects the domains of datasets. For instance, BLTC, OpSpam, as well as DeRev2014 and DeRev2018 from the review domain have similarity scores of 0.89 and .87, respectively. The opinion statement datasets CrossCultDe and DecOp exhibit a high similarity score of 0.83. Similarly, Mafiascum and Diplomacy show relatively high similarity (0.77), despite differences in the game rules.

Datasets obtained under specific conditions within the same domain are assigned a lower similarity score. For instance, BoxOfLies, which is a game that takes place in an in-person setting, differs from the online game datasets (.42 with Diplomacy and Mafiascum). We also observe similarity across domains, e.g., Bluff is more similar to reviews than games, presumably due to its monologue setting instead of dialogue.

5 Linguistic Correlation Analysis

To understand the generalizability of linguistic cues across different dataset, we conduct a correlation analysis, similar to previous studies that focused on isolated or smaller numbers of corpora (Pérez-Rosas and Mihalcea, 2015; Skalicky et al., 2020)

5.1 Method

We aim at identifying frequently used features which are general across domains. We build our analysis on the "Linguistic Inquiry and Word Count" (LIWC22[19], Pennebaker et al., 2015) and Flesch-Kincaid (Kincaid et al., 1975) and Gunning Fog (Robert, 1968) readability scores as measures of complexity or sophistication of language.[20]

We use point-biserial correlation[21] (Glass and Hopkins, 1996) to measure the relation between deception labels (discrete) and a score assigned by LIWC or readability measurement (continuous). The correlation value ranges from -1 to $+1$.

5.2 Results

Table 2 lists the features which show at least a weak correlation (> 0.15) with $p \leq 0.05$ for at least three datasets. The positive and negative correlation values correspond to the strength of association with truth and deception respectively.

---

[18] We use the Python implementation `https://github.com/jonathandunn/corpus_similarity`

[19] `https://www.liwc.app/`

[20] `https://pypi.org/project/readability/`.

[21] `https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.pointbiserialr.html`

	Datasets													
Features	BLTC	Bluff	BoxOfLies	CrossCultDe	DecOp	DeRev2014	DeRev2018	Diplomacy	Mafiascum	MU3D	OnlineDe	OpenDomain	OpSpam	Trial
Analytic	**.13**	−.04	**.12**	.01	.02	**−.25**	**.23**	**.02**	**−.02**	**.14**	**.10**	**.05**	**.15**	**.25**
Authentic	.03	−.05	.00	**.28**	**.22**	**.28**	**−.05**	**−.03**	−.02	.07	.00	**−.04**	**−.09**	−.09
BigWords	.02	.00	**.18**	.04	**.05**	**−.21**	**.24**	.01	−.01	**.18**	**−.01**	**.03**	**−.08**	.09
Clout	.00	.00	.02	**−.11**	**−.28**	**−.45**	.00	**.02**	**.02**	.03	**−.05**	.01	**.10**	**.26**
Cognition	**−.08**	**.17**	−.05	.02	**.07**	−.06	**−.13**	−.01	−.01	**−.17**	.00	**−.09**	**−.06**	**−.28**
GunningFog	**.18**	**−.21**	**.12**	**.21**	**.25**	.01	**.13**	**−.09**	**−.03**	−.04	**.13**	.02	.02	.06
Kincaid	**.18**	**−.21**	**.14**	**.2**	**.24**	.01	**.13**	**−.08**	**−.03**	−.04	**.13**	**.03**	.02	.06
Linguistic	**−.07**	**.10**	**−.15**	.04	**.10**	**.29**	**−.14**	**−.02**	**−.03**	**−.16**	**−.05**	**−.05**	**−.18**	−.08
Period	.01	**−.07**	.02	**−.11**	**−.18**	**.26**	**−.07**	.00	.00	.03	**.01**	**.03**	**.24**	−.06
Physical	.02	.03	**.15**	−.04	**−.16**	**−.25**	**.06**	.00	**.03**	.04	**−.15**	−.01	−.01	.06
WC	**.18**	**−.21**	.04	**.22**	**.25**	.02	**.13**	**−.10**	.01	−.04	**.13**	−.02	.02	.06
auxverb	**−.08**	**.12**	−.06	**−.08**	**−.09**	**.22**	**−.12**	−.01	**.02**	**−.15**	.00	**.03**	**−.08**	**−.21**
focusfuture	**−.09**	**.09**	−.02	−.04	**−.08**	**−.17**	**−.2**	−.01	.02	−.04	**.01**	**−.04**	**−.16**	.08
function	−.05	**.13**	−.03	.00	**.10**	**.25**	**−.06**	**−.04**	**−.03**	**−.15**	**−.03**	**−.05**	**−.23**	**−.23**
i	**−.06**	**−.15**	−.07	**.13**	**−.3**	**.39**	**−.16**	**−.05**	.02	−.01	**−.12**	**−.04**	**−.33**	−.13
shehe	.01	**−.11**	−.03	**−.15**	.00	**−.17**	**−.07**	.00	**−.04**	**−.14**	**.04**	**−.04**	−.01	**−.18**
verb	**−.11**	.07	**−.09**	**−.06**	**−.07**	**.16**	**−.26**	**−.02**	.00	**−.14**	**−.07**	−.01	**−.16**	−.14
you	**−.10**	**.17**	−.03	−.05	**−.07**	**−.19**	**−.23**	.01	**.03**	−.08	**−.05**	**−.05**	.01	−.05

Table 2: Point-biserial correlation between the deception labels and linguistic features (LIWC categories + readability). We only show features with a correlation coefficient of $\geq .15$ and $p \leq .05$ for at least three datasets. Correlation scores with $p \leq .05$ are shown in bold.

Deceptive language is argued to have fewer self-references ("i") and more references to others ("shehe", "you"), as liars attempt to distance themselves from their lies (Newman et al., 2003; DePaulo et al., 2003). Our analysis supports this hypothesis in the categories "shehe" and "you" for a substantial number of data sets. Contrary to our expectation, however, in 8 out of 14 datasets the category "i" is seen to correlate with deception and not with truth, with an exception of CrossCultDe ($\rho = .13$) and DeRev2018 (.39).

Studies have attributed less cognitive complexity in language to deceptive communication (Newman et al., 2003; DePaulo et al., 2003). Liars use fewer words related to cognitive concepts (e.g., *think*, *believe*), which should correspond to a positive correlation value for the category "Cognition" in LIWC. However, our analysis corroborates this observation only in Bluff ($\rho = .17$) and DecOp ($\rho = .07$).

In general, we found no consistent linguistic cues across domains and datasets in our analysis. This might be because deception is highly sensitive to the goal of a lie and the stakes involved, which is not consistent across the domains under consideration.

6 Deception Detection Experiments

The correlation analysis in the previous section showed that deception cues do barely generalize across domains. This analysis might be limited by the choice of categories, which motivates us to conduct cross-corpus modeling experiments.

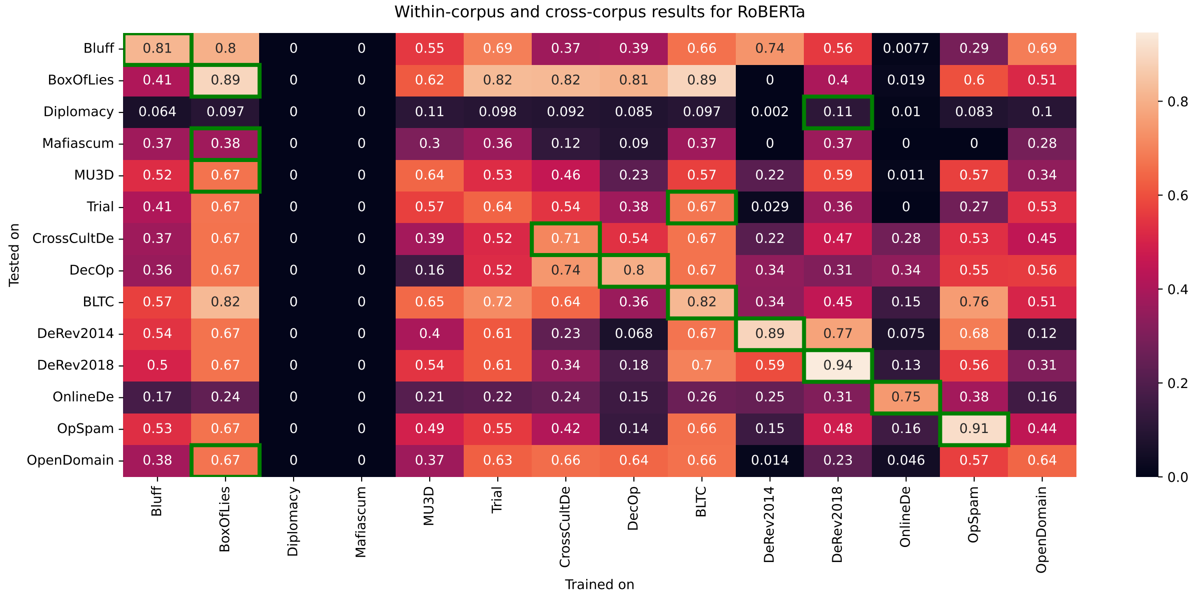
6.1 Experimental Setup

In the within-corpus setup, we fine-tune and evaluate RoBERTa models (Liu et al., 2019) on the same dataset via 10-fold cross-validation. In the cross-corpus setting, we train on one corpus and test on the other. To ensure comparability between these experiments, we perform 10-fold cross-validation in both settings: we also evaluate 10 times on the same corpus subsets in the cross-corpus setup. This is not strictly required but ensures comparability.

We use the English RoBERTa-base, with 12 layers, 768 hidden-states, 12 heads and 125M parameters as available in the HuggingFace implementation (Wolf et al., 2020). We finetune with default hyperparameters for 6 epochs using the Auto Model for Sequence Classification. [22]

---

[22] `https://huggingface.co/transformers/v3.0.2/model_doc/auto.html`


Within-corpus and cross-corpus results for RoBERTa

Tested on \ Trained on	Bluff	BoxOfLies	Diplomacy	Mafiascum	MU3D	Trial	CrossCultDe	DecOp	BLTC	DeRev2014	DeRev2018	OnlineDe	OpSpam	OpenDomain
Bluff	0.81	0.8	0	0	0.55	0.69	0.37	0.39	0.66	0.74	0.56	0.0077	0.29	0.69
BoxOfLies	0.41	0.89	0	0	0.62	0.82	0.82	0.81	0.89	0	0.4	0.019	0.6	0.51
Diplomacy	0.064	0.097	0	0	0.11	0.098	0.092	0.085	0.097	0.002	0.11	0.01	0.083	0.1
Mafiascum	0.37	0.38	0	0	0.3	0.36	0.12	0.09	0.37	0	0.37	0	0	0.28
MU3D	0.52	0.67	0	0	0.64	0.53	0.46	0.23	0.57	0.22	0.59	0.011	0.57	0.34
Trial	0.41	0.67	0	0	0.57	0.64	0.54	0.38	0.67	0.029	0.36	0	0.27	0.53
CrossCultDe	0.37	0.67	0	0	0.39	0.52	0.71	0.54	0.67	0.22	0.47	0.28	0.53	0.45
DecOp	0.36	0.67	0	0	0.16	0.52	0.74	0.8	0.67	0.34	0.31	0.34	0.55	0.56
BLTC	0.57	0.82	0	0	0.65	0.72	0.64	0.36	0.82	0.34	0.45	0.15	0.76	0.51
DeRev2014	0.54	0.67	0	0	0.4	0.61	0.23	0.068	0.67	0.89	0.77	0.075	0.68	0.12
DeRev2018	0.5	0.67	0	0	0.54	0.61	0.34	0.18	0.7	0.59	0.94	0.13	0.56	0.31
OnlineDe	0.17	0.24	0	0	0.21	0.22	0.24	0.15	0.26	0.25	0.31	0.75	0.38	0.16
OpSpam	0.53	0.67	0	0	0.49	0.55	0.42	0.14	0.66	0.15	0.48	0.16	0.91	0.44
OpenDomain	0.38	0.67	0	0	0.37	0.63	0.66	0.64	0.66	0.014	0.23	0.046	0.57	0.64

Figure 2: Performance of RoBERTa models with F_1 measure on the deception label. The best model on each test set is highlighted with a green box.

6.2 Results

The heatmap in Figure 2 shows the results as F_1 measure for the deception label (Appendix B shows results for both labels). The diagonal corresponds to within-corpus experiments. For most datasets, the model shows better performance in the within-corpus setting than in the cross-corpus evaluation. This is not the case for MU3D, TRIAL, and OPENDOMAIN, but the difference is negligible (0.04).

Models on datasets from the same domain or which are otherwise similar (§ 4) show comparably better results in the cross-corpus setting. For instance, training on OPSPAM and testing on BLTC achieves an F_1 score of 0.76 on the deception label. Training on BLTC and testing on OPSPAM is however not as good (0.66). Similar observations can be made for DEREV2014 and DEREV2018, and CROSSCULTDE and DECOP.

The heatmap shows the lowest performance for MAFIASCUM and DIPLOMACY, with an F_1=0. We assume that this is a result of the imbalanced label distribution in DIPLOMACY and the long documents in MAFIASCUM (see Table 1). Similarly, the exceptionally good results on the BOXOFLIES test set are due to the bias towards the deceptive label (see appendix for F_1 score on truth label).

Note that previous work reported other evaluation measures than F_1, which makes this dramatically low performance difficult to compare. Our evaluation with accuracy (shown in the appendix in Figure 4) appears to be more positive with .77 and .95.

From the sub-par results on cross-corpus experiments, we conclude that generalization across domains and dissimilar datasets is challenging, even with pre-trained language models with rich contextual information. In our future work, we plan to use this dataset to train models that can capture domain-independent cues of deception, which can presumably generalize better across datasets.

7 Conclusion & Future Work

Different scientific disciplines have contributed to the creation of deception datasets for textual communication in a variety of domains. In this study, we present a comprehensive survey of deception datasets in English available for research and compile them into a unified deception dataset. We are not aware of any previous work that considered a comparably large amount of corpora and evaluated models between all of them. Some of the evaluation results are encouraging, but particularly between dissimilar domains, the generalization is limited and requires future research.

The RoBERTa-based classification experiments and linguistic correlation analysis of deception cues demonstrate that it is indeed challenging to generalize the concept of deception across datasets, or domains. In the classification experiment results, the wildly diverging F_1 scores can be attributed to the complexity of the task as well as to the limitations of the approach employed. In future work, we plan to explore the reasons for this variability across datasets further.

Additionally, we acknowledge the need to address the issue of biased models, such as the ones trained on MAFIASCUM, ONLINEDE, and DIPLOMACY, which tends to favor truthful labels owing to the label imbalance in these datasets, resulting in an F_1 score of 0. To overcome this challenge, we could employ techniques like oversampling to rectify the class imbalance and improve the reliability and effectiveness of our approach.

The goal of our future work is to create robust deception detection models that work reliably across corpora and domains. This includes understanding differences in the concept as it represents itself in these data and understanding differences in linguistic realization.

Our UNIDECOR dataset serves as a valuable resource for future research enabling standardized data comparison, transfer learning, and domain adaptation experiments.

Acknowledgments

We thank Kai Sassenberg for fruitful discussions regarding the concept of deception. Roman Klinger's research is partially funded by the German Research Council (DFG), projects KL 2869/1-2 and KL 2869/5-1.

Limitations

The goal of the current study was to unify the resources available for deception and report observations on cross-corpus and within-corpus analyses. While reporting the baseline performance using RoBERTa, we did not perform any optimization specific to the datasets. Hence, better results might be reported in the papers which handle the datasets or domains in isolation.

Ethical Considerations

The datasets used in this research are publicly available resources from previous studies. We have taken appropriate steps to ensure that we do not violate any license terms or intellectual property rights. Also, proper attribution is given to the original sources of the data. Deception is a sensitive topic, and non-anonymous data should not be used. To the best of our knowledge, all data sets that we considered have been compiled or collected according to such standards.

The performance of deception detection systems is not perfect, making them unsuitable for examining the utterances of individuals due to the threat of incorrect predictions. Even if automatic systems might reach a close-to-perfect performance, we consider their practical application to analyze and profile people unethical. However, there might be use cases, for instance in forensics, that can be considered ethical from a utilitaristic perspective.

Given the ethical implications of employing automated deception detection systems on individual, non-anonymous statements, we propose utilizing the resources collected and models developed on anonymous data. Any data analysis that could lead back to its origin must only be conducted with the data creator's informed consent and knowledge of potential consequences.

We consider the research in this paper to be fundamental, with the goal of better understanding human communication.

References

Gary D. Bond and Adrienne Y. Lee. 2005. Language of lies in prison: linguistic classification of prisoners' truthful and deceptive natural language. *Applied Cognitive Psychology*, 19(3):313–329.

Charles F. Bond Jr. and Bella M. DePaulo. 2006. Accuracy of deception judgments. *Personality and Social Psychology Review*, 10(3):214–234.

Pasquale Capuozzo, Ivano Lauriola, Carlo Strapparava, Fabio Aiolli, and Giuseppe Sartori. 2020. DecOp: A multilingual and multi-domain corpus for detecting deception in typed text. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1423–1430, Marseille, France. European Language Resources Association.

Nadia K. Conroy, Victoria L. Rubin, and Yimin Chen. 2015. Automatic deception detection: Methods for finding fake news. *Proceedings of the Association for Information Science and Technology*, 52(1):1–4.

Bob de Ruiter and George Kachergis. 2018. The mafiascum dataset: A large text corpus for deception detection. *ArXiv*, abs/1811.07851.

Bella M. DePaulo, James J. Lindsay, Brian E. Malone, Laura Muhlenbruck, Kelly Charlton, and Harris Cooper. 2003. Cues to deception. *Psychological bulletin*, 129(1):74–118.

Bella M. DePaulo and Wendy L. Morris. 2004. *Discerning lies from truths: behavioural cues to deception and the indirect pathway of intuition*, page 15–40. Cambridge University Press.

Nicholas D. Duran, Charles Hall, Philip M. McCarthy, and Danielle S. McNamara. 2010. The linguistic correlates of conversational deception: Comparing natural language processing technologies. *Applied Psycholinguistics*, 31(3):439–462.

Amir Fayazi, Kyumin Lee, James Caverlee, and Anna Squicciarini. 2015. Uncovering crowdsourced manipulation of online reviews. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, page 233–242, New York, NY, USA. Association for Computing Machinery.

Tommaso Fornaciari, Federico Bianchi, Massimo Poesio, and Dirk Hovy. 2021. BERTective: Language models and contextual information for deception detection. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2699–2708, Online. Association for Computational Linguistics.

Tommaso Fornaciari, Leticia Cagnina, Paolo Rosso, and Massimo Poesio. 2020. Fake opinion detection: how similar are crowdsourced datasets to real data? *Language Resources and Evaluation*, pages 1–40.

Tommaso Fornaciari and Massimo Poesio. 2014. Identifying fake Amazon reviews as learning from crowds. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 279–287, Gothenburg, Sweden. Association for Computational Linguistics.

G.V. Glass and K.D. Hopkins. 1996. *Statistical Methods in Education and Psychology*. Allyn and Bacon.

Swati Gupta, Kayo Sakamoto, and Andrew Ortony. 2013. Telling it like it isn't: A comprehensive approach to analyzing verbal deception. Online.

Jeffrey T. Hancock. 2009. Digital deception: Why, when and how people lie online. In *Oxford Handbook of Internet Psychology*. Oxford University Press.

Jeffrey T. Hancock, Lauren E. Curry, Saurabh Goorha, and Michael Woodworth. 2007. On lying and being lied to: A linguistic analysis of deception in computer-mediated communication. *Discourse Processes*, 45(1):1–23.

Valerie Hauch, Iris Blandón-Gitlin, Jaume Masip, and Siegfried L. Sporer. 2015. Are computers effective lie detectors? a meta-analysis of linguistic cues to deception. *Personality and Social Psychology Review*, 19(4):307–342.

Ángel Hernández-Castañeda, Hiram Calvo, Alexander Gelbukh, and Jorge J. García Flores. 2016. Cross-domain deception detection using support vector networks. *Soft Computing*, 21(3):585–595.

J. Peter Kincaid, Robert P. Fishburne Jr, Richard L. Rogers, and Brad S. Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical Report 8-75, University of Central Florida, Institute for Simulation and Training.

Timothy R. Levine. 2014. Truth-default theory (tdt): A theory of human deception and deception detection. *Journal of Language and Social Psychology*, 33(4):378–392.

Haipeng Li and Jonathan Dunn. 2022. Corpus similarity measures remain robust across diverse languages. *Lingua*, 275:103377.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized bert pretraining approach.

Paige E. Lloyd, Jason C. Deska, Kurt Hugenberg, Allen R. McConnell, Brandon T. Humphrey, and Jonathan W. Kunstman. 2019. Miami university deception detection database. *Behavior Research Methods*, 51:429–439.

James Edwin Mahon. 2007. A definition of deceiving. *International Journal of Applied Philosophy*, 21(2):181–194.

Sandra Metts. 1989. An exploratory investigation of deception in close relationships. *Journal of Social and Personal Relationships*, 6(2):159–179.

Matthew L. Newman, James W. Pennebaker, Diane S. Berry, and Jane M. Richards. 2003. Lying words: Predicting deception from linguistic styles. *Personality and Social Psychology Bulletin*, 29(5):665–675.

Myle Ott, Claire Cardie, and Jeffrey T. Hancock. 2013. Negative deceptive opinion spam. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 497–501, Atlanta, Georgia. Association for Computational Linguistics.

Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T. Hancock. 2011. Finding deceptive opinion spam by any stretch of the imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 309–319, Portland, Oregon, USA. Association for Computational Linguistics.

James W. Pennebaker, Ryan Boyd, Kayla Jordan, and Kate Blackburn. 2015. *The development and psychometric properties of LIWC2015*. University of Texas at Austin.

Verónica Pérez-Rosas, Mohamed Abouelenien, Rada Mihalcea, and Mihai Burzo. 2015. Deception detection using real-life trial data. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ICMI '15, page 59–66, New York, NY, USA. Association for Computing Machinery.

Verónica Pérez-Rosas and Rada Mihalcea. 2014. Cross-cultural deception detection. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 440–445, Baltimore, Maryland. Association for Computational Linguistics.

Verónica Pérez-Rosas and Rada Mihalcea. 2015. Experiments in open domain deception detection. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1120–1125, Lisbon, Portugal. Association for Computational Linguistics.

Denis Peskov, Benny Cheng, Ahmed Elgohary, Joe Barrow, Cristian Danescu-Niculescu-Mizil, and Jordan Boyd-Graber. 2020. It takes two to lie: One to lie, and one to listen. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3811–3854, Online. Association for Computational Linguistics.

Gunning Robert. 1968. *The Technique of Clear Writing*. McGraw-Hill, New York.

Franco Salvetti, John B. Lowe, and James H. Martin. 2016. A tangled web: The faint signals of deception in text - boulder lies and truth corpus (BLT-C). In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3510–3517, Portorož, Slovenia. European Language Resources Association (ELRA).

Stephen Cameron Skalicky, Nicholas D. Duran, and Scott Andrew Crossley. 2020. Please, please, just tell me: The linguistic features of humorous deception. *Dialogue Discourse*, 11:128–149.

Felix Soldner, Verónica Pérez-Rosas, and Rada Mihalcea. 2019. Box of lies: Multimodal deception detection in dialogues. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1768–1777, Minneapolis, Minnesota. Association for Computational Linguistics.

Lyn M. Van Swol, Michael T. Braun, and Deepak Malhotra. 2012. Evidence for the pinocchio effect: Linguistic differences between lies, deception by omissions, and truths. *Discourse Processes*, 49(2):79–106.

Leanne Ten Brinke and Stephen Porter. 2012. Cry me a river: identifying the behavioral consequences of extremely high-stakes interpersonal deception. *Law and Human Behavior*, 36(6):469–477.

Ronny E. Turner, Charles Edgley, and Glen Olmstead. 1975. Information control in conversations: Honesty is not always the best policy. *The Kansas Journal of Sociology*, 11(1):69–89.

Aldert Vrij. 2014. *14. Detecting lies and deceit: Pitfalls and opportunities in nonverbal and verbal lie detection*, pages 321–346. De Gruyter Mouton, Berlin, Boston.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Wenlin Yao, Zeyu Dai, Ruihong Huang, and James Caverlee. 2017. Online deception detection refueled by real world data collection. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 793–802, Varna, Bulgaria. INCOMA Ltd.

Lina Zhou, Judee K. Burgoon, Jay F. Nunamaker, and Doug Twitchell. 2004. Automating linguistics-based cues for detecting deception in text-based asynchronous computer-mediated communications. *Group Decision and Negotiation*, 13(1):81–106.

Miron Zuckerman, Bella M. DePaulo, and Robert Rosenthal. 1981. Verbal and nonverbal communication of deception. In Leonard Berkowitz, editor, *Advances in Experimental Social Psychology*, volume 14, pages 1–59. Academic Press.

M. Umut Şen, Verónica Pérez-Rosas, Berrin Yanikoglu, Mohamed Abouelenien, Mihai Burzo, and Rada Mihalcea. 2022. Multimodal deception detection using real-life trial data. *IEEE Transactions on Affective Computing*, 13(1):306–319.

Appendix

A Details on the Aggregated Dataset

All datasets included in the unified collection contains one binary label indicating whether an instance is truthful or deceptive, the naming convention for which has been normalized retaining the original label for backward compatibility. However, some datasets like Salvetti et al. (2016) and Peskov et al. (2020) include an additional dimension for deception, where the former differentiates between lying about a known object and lying about an unknown object, and the latter contains annotations on the *perceived truthfulness* of the statement in addition to the actual intention. For providing a unified format, we map both these deceptive instances in Salvetti et al. (2016) to one label and since the perceived truthfulness is an independent annotation, we do not take this label into account.

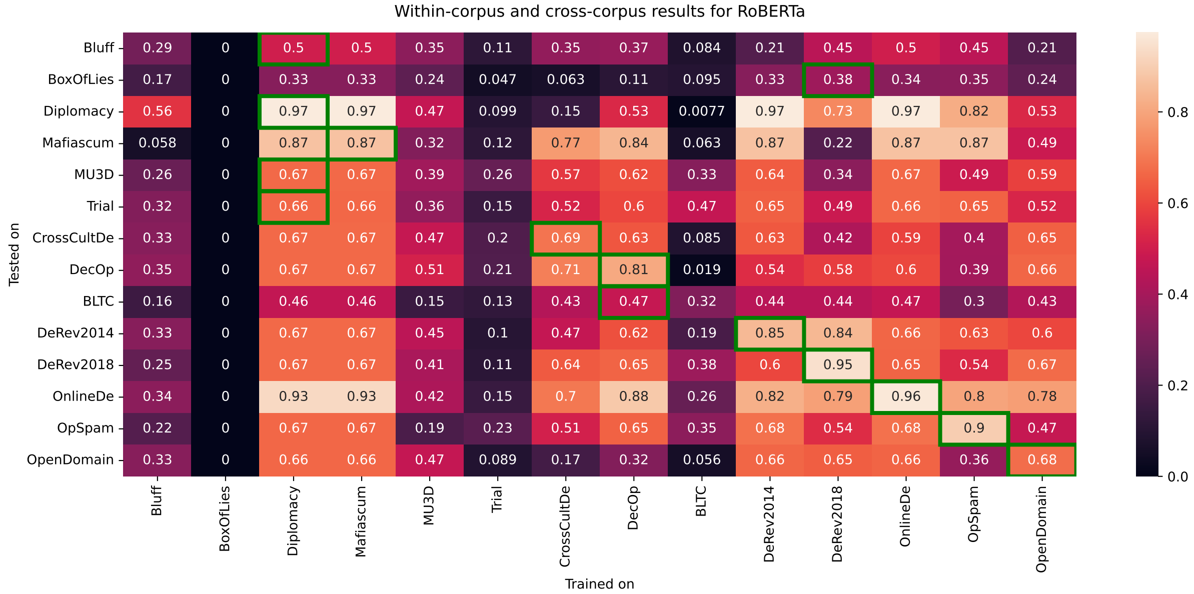
In addition to truth labels, datasets also contain additional annotations like demographic information related to the author, sentiment, valency of the instance and perceived truthfulness. We retain only those annotation dimensions which are available for more than one dataset which are age, gender, country, and sentiment

The unified dataset includes corpora that are available for research purposes which are downloadable from source, made available directly by the creators, or obtained from a consortium like the Linguistic Data Consortium. We provide a script to automatically download all datasets if they are available for download, which otherwise provides instructions on how to obtain them. Once all datasets are populated in their respective folders, a second script is used to generate the unified dataset in json format. You can find the repository with instructions to obtain the aggregated UNIDECOR, Unified Deception Corpus, at `https://www.ims.uni-stuttgart.de/data/unidecor`.

The following entry shows an example instance from the corpus.

```
{
        "source": "OPEN_DOMAIN",
        "text_ID": "119_f_t_1",
        "text": "Thad cochran has been in the us senate since before the internet was
        invented.",
        "participant_ID": "NA",
        "age": "20",
        "sentiment": "NA",
        "language": "EN",
        "gender": "Female",
        "country": "United States",
        "original_label": "truth",
        "truth_label": "T",
        "topic_name": "statement",
        "domain": "opinion",
        "mode": "written",
        "split": null,
        "fold": null
}
```

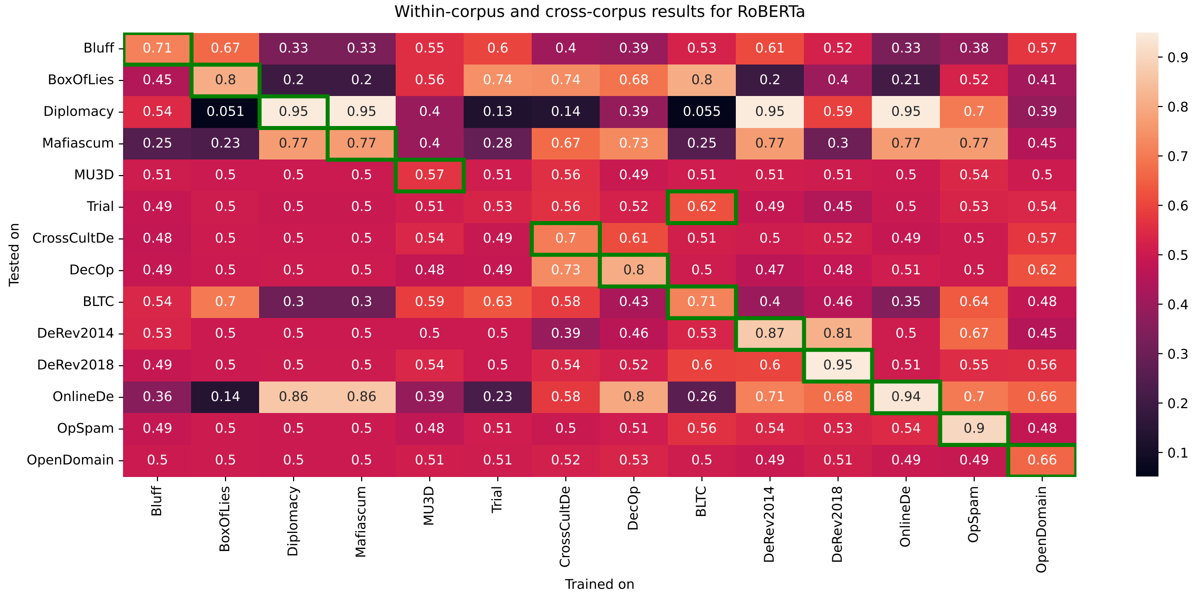
B Additional Experimental Results



Within-corpus and cross-corpus results for RoBERTa

	Bluff	BoxOfLies	Diplomacy	Mafiascum	MU3D	Trial	CrossCultDe	DecOp	BLTC	DeRev2014	DeRev2018	OnlineDe	OpSpam	OpenDomain
Bluff	0.29	0	0.5	0.5	0.35	0.11	0.35	0.37	0.084	0.21	0.45	0.5	0.45	0.21
BoxOfLies	0.17	0	0.33	0.33	0.24	0.047	0.063	0.11	0.095	0.33	0.38	0.34	0.35	0.24
Diplomacy	0.56	0	0.97	0.97	0.47	0.099	0.15	0.53	0.0077	0.97	0.73	0.97	0.82	0.53
Mafiascum	0.058	0	0.87	0.87	0.32	0.12	0.77	0.84	0.063	0.87	0.22	0.87	0.87	0.49
MU3D	0.26	0	0.67	0.67	0.39	0.26	0.57	0.62	0.33	0.64	0.34	0.67	0.49	0.59
Trial	0.32	0	0.66	0.66	0.36	0.15	0.52	0.6	0.47	0.65	0.49	0.66	0.65	0.52
CrossCultDe	0.33	0	0.67	0.67	0.47	0.2	0.69	0.63	0.085	0.63	0.42	0.59	0.4	0.65
DecOp	0.35	0	0.67	0.67	0.51	0.21	0.71	0.81	0.019	0.54	0.58	0.6	0.39	0.66
BLTC	0.16	0	0.46	0.46	0.15	0.13	0.43	0.47	0.32	0.44	0.44	0.47	0.3	0.43
DeRev2014	0.33	0	0.67	0.67	0.45	0.1	0.47	0.62	0.19	0.85	0.84	0.66	0.63	0.6
DeRev2018	0.25	0	0.67	0.67	0.41	0.11	0.64	0.65	0.38	0.6	0.95	0.65	0.54	0.67
OnlineDe	0.34	0	0.93	0.93	0.42	0.15	0.7	0.88	0.26	0.82	0.79	0.96	0.8	0.78
OpSpam	0.22	0	0.67	0.67	0.19	0.23	0.51	0.65	0.35	0.68	0.54	0.68	0.9	0.47
OpenDomain	0.33	0	0.66	0.66	0.47	0.089	0.17	0.32	0.056	0.66	0.65	0.66	0.36	0.68

Figure 3: A heatmap representing the performance of RoBERTa model with the F_1 measure on the truth label across different datasets. Figure 2 in the main paper analogously shows the results for the deception category.



Within-corpus and cross-corpus results for RoBERTa

	Bluff	BoxOfLies	Diplomacy	Mafiascum	MU3D	Trial	CrossCultDe	DecOp	BLTC	DeRev2014	DeRev2018	OnlineDe	OpSpam	OpenDomain
Bluff	0.71	0.67	0.33	0.33	0.55	0.6	0.4	0.39	0.53	0.61	0.52	0.33	0.38	0.57
BoxOfLies	0.45	0.8	0.2	0.2	0.56	0.74	0.74	0.68	0.8	0.2	0.4	0.21	0.52	0.41
Diplomacy	0.54	0.051	0.95	0.95	0.4	0.13	0.14	0.39	0.055	0.95	0.59	0.95	0.7	0.39
Mafiascum	0.25	0.23	0.77	0.77	0.4	0.28	0.67	0.73	0.25	0.77	0.3	0.77	0.77	0.45
MU3D	0.51	0.5	0.5	0.5	0.57	0.51	0.56	0.49	0.51	0.51	0.51	0.5	0.54	0.5
Trial	0.49	0.5	0.5	0.5	0.51	0.53	0.56	0.52	0.62	0.49	0.45	0.5	0.53	0.54
CrossCultDe	0.48	0.5	0.5	0.5	0.54	0.49	0.7	0.61	0.51	0.5	0.52	0.49	0.5	0.57
DecOp	0.49	0.5	0.5	0.5	0.48	0.49	0.73	0.8	0.5	0.47	0.48	0.51	0.5	0.62
BLTC	0.54	0.7	0.3	0.3	0.59	0.63	0.58	0.43	0.71	0.4	0.46	0.35	0.64	0.48
DeRev2014	0.53	0.5	0.5	0.5	0.5	0.5	0.39	0.46	0.53	0.87	0.81	0.5	0.67	0.45
DeRev2018	0.49	0.5	0.5	0.5	0.54	0.5	0.54	0.52	0.6	0.6	0.95	0.51	0.55	0.56
OnlineDe	0.36	0.14	0.86	0.86	0.39	0.23	0.58	0.8	0.26	0.71	0.68	0.94	0.7	0.66
OpSpam	0.49	0.5	0.5	0.5	0.48	0.51	0.5	0.51	0.56	0.54	0.53	0.54	0.9	0.48
OpenDomain	0.5	0.5	0.5	0.5	0.51	0.51	0.52	0.53	0.5	0.49	0.51	0.49	0.49	0.66

Figure 4: A heatmap representing the accuracy of RoBERTa model different datasets. As the categories of truth and deception and mutual exclusive in all our datasets, this corresponds to a micro-average of the results shown in Figure 2 and 3.

Discourse Mode Categorization of Bengali Social Media Health Text

Salim Sazzed
Department of Computer Science
Old Dominion University
Norfolk, VA, 23529
salim.sazzed@gmail.com

Abstract

The scarcity of annotated data is a major impediment to natural language processing (NLP) research in Bengali, a language that is considered low-resource. In particular, the health and medical domains suffer from a severe paucity of annotated data. Thus, this study aims to introduce BanglaSocialHealth, an annotated social media health corpus that provides sentence-level annotations of four distinct types of expression modes, namely narrative (NAR), informative (INF), suggestive (SUG), and inquiring (INQ) modes in Bengali. We provide details regarding the annotation procedures and report various statistics, such as the median and mean length of words in different sentence modes. Additionally, we apply classical machine learning (CML) classifiers and transformer-based language models to classify sentence modes. We find that most of the statistical properties are similar in different types of sentence modes. To determine the sentence mode, the transformer-based M-BERT model provides slightly better efficacy than the CML classifiers. Our developed corpus and analysis represent a much-needed contribution to Bengali NLP research in medical and health domains and have the potential to facilitate a range of downstream tasks, including question-answering, misinformation detection, and information retrieval.

1 Introduction

With the increasing popularity of social media, various types of online content generated by vast numbers of people have become available. The health and medicine-related data are no exception, accumulating at a high pace as more and more people are using social media for health-related queries and discussions (Andy et al., 2021; Ganti et al., 2022). In fact, nowadays, by possessing ample amounts of health-related information, social media has become one of the prominent data sources for health-related research. People all over the world use online health forums to acquire medical information. Besides, people share their experiences regarding diseases, symptoms, and related matters to help other patients. Due to the importance of medical and health text mining, the NLP community has organized a series of open challenges focusing on biomedical entity extraction and classification (Weissenbacher et al., 2019).

The importance of social support in online health forums has been discussed in many earlier studies (Wang et al., 2012; Yang et al., 2017). As individuals seek support and information regarding various health-related issues in health and well-being forums, it is imperative to analyze them for a better understanding of user needs and to provide required support (Andy et al., 2021; Moorhead et al., 2013). For example, Andy et al. (2021), in their study, classified COVID-related health text into the following four categories: i) Emotional Support Given, ii) Emotional Support Sought, iii) Informational Support Given, and iv) Informational Support Sought. Another health text classification task was performed by Ganti et al. (2022), where the authors classified health-related text into narratives and non-narrative categories. A study related to the identification of informative health posts was conducted by Olsen and Plank (2021).

Categorizing health-related text on social media into distinct discourse modes can be beneficial for a range of downstream natural language processing (NLP) tasks, as each mode has specific roles in health support and discussion. For instance, user-generated questions can provide insights into the outbreak of the disease over time (Wen and Rosé, 2012) and can facilitate the development of social support chatbots that cater to the needs of individuals seeking healthcare-related assistance (Wang et al., 2021). User narratives or experiences can reveal valuable information on disease symptoms and severity. In addition, it can help to find peers with similar experiences (Levonian et al., 2021). It is imperative to analyze the suggestions or information-

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 52–57
July 14, 2023 ©2023 Association for Computational Linguistics

related discourse shared by peer users to detect the dissemination of disinformation and misinformation (Wang et al., 2019).

In English and a few other major languages, various health-related corpora are publicly available (Kolárik et al., 2008). However, although Bengali (also known as Bangla) is one of the most spoken languages in the world [1], such resources are typically not available (Sazzed, 2022). With the growing popularity of telemedicine and the availability of health and medical-related data written in Bengali, creating resources for developing an NLP-based health system in Bengali is a pressing necessity. To comprehend and automatically categorize health-related text, it is essential to have at least a moderate amount of annotated data.

Hence, in this study, we introduce a discourse mode annotated health corpus, the first of its kind, for the low-resource Bengali language. The dataset is created by retrieving publicly available health-related texts from a number of social media health forums. The retrieved text data are tokenized into sentence-level and annotated with four types of discourse modes: narrative (NAR), informative (INF), suggestive (SUG), and inquiring (INQ). The final corpus contains around 2000 sentences annotated by one of the four types of sentence modes. The details of the annotation procedure and various statistics of the sentence modes are provided. In addition, we present a baseline evaluation by employing multiple ML classifiers for the automatic categorization of the sentence modes. We observe that top classical ML classifiers and deep learning-based models demonstrate similar efficacy for the classification tasks.

1.1 Contributions

The main contributions of this study can be summarized as follows:

- To address the lack of annotated health-related text data in Bengali, we present a health corpus, BanglaSocialHealth, by collecting health data from various Bengali health forums.

- We manually annotate around 2000 sentences into four types of modes: narrative, informative, suggestive, and inquiring. The dataset is publicly available in the following link [2].

- We provide various statistics, such as frequency and attributes of discourse mode annotated sentences in the corpus.

- Finally, we provide the baseline evaluations of the classification tasks utilizing both classical machine learning classifiers and multi-lingual BERT.

2 Creation of BanglaSocialHealth

2.1 Data Collection

We obtain health-related textual data from multiple Bengali Facebook pages where individuals actively engage in discussions related to health. These discussions involve inquiries about health issues, recommendations, and information sharing concerning symptoms and disease prevention. While the majority of posts consist of interactive discussions, such as questions, answers, and suggestions from individuals, we also encounter health-related articles authored by healthcare professionals that provide informative content. To maintain the anonymity of the users, we do not collect any user information; only user-written texts are extracted. Therefore, the dataset is anonymous. The data collection period spans from May 2022 to July 2022.

The posts are manually retrieved from the Facebook pages for annotation. We find the textual contents in the posts represent three different forms of languages: Bengali, English, and transliterated Bengali. Since we are only interested in Bengali text, we collected only the posts written primarily in Bengali. The excerpted texts are automatically segmented into sentence-level tokens based on the Bengali *dari* (i.e., '।') delimiter, which is equivalent to the English full stop ('.') delimiter. The sentence-level tokens are then manually reviewed to ensure each represents a contextually meaningful single sentence. As social media data are noisy, it is not uncommon to have sentences with missing delimiters. Again, some sentences may end with different types of delimiters. The manual examination assures each instance represents a complete sentence. Any sentence written in English or in transliterated Bengali in a post is excluded from annotation.

2.2 Discourse Modes

The following four types of discourse modes are considered during the annotation process.

---

[1] `https://www.berlitz.com/blog/most-spoken-languages-world`

[2] `https://github.com/sazzadcsedu/BanglaHealthText.git`

- **Narrative (NAR)**: This discourse mode is related to the written narratives. Narration is the use of a written or spoken commentary to convey a story, such as a particular event or scenario, to an audience [3]. For example, an individual may tell about the experience and suffering regarding a particular disease or symptoms.

- **Inquiring (INQ)**: This discourse mode pertains to sentences that embody user inquiries and requests for information and recommendations. For example, in a health forum, a user may ask questions or seek suggestions concerning disease/symptoms, or request information regarding other health-related concerns.

- **Informative (INF)**: This discourse mode comprises informative sentences, which primarily convey factual information. For instance, sentences that encompass information about disease attributes and preventive measures are categorized within this mode.

- **Suggestive (SUG)**:The suggestive sentence primarily encompasses suggestions, advice, or recommendations offered in response to an individual's request for guidance. For instance, when a user seeks recommendations for a specialized doctor, another user may respond by providing specific suggestions.

2.3 Data Annotation Guidelines

In order to assign the discourse mode at the sentence level, annotators are provided with the aforementioned definitions and corresponding examples as guidelines. Initially, two annotators label all the sentences, and a third annotator intervenes only in cases where there is a disagreement between the first two annotators. We observed an annotator agreement of 0.80, calculated using Cohen's kappa (Cohen, 1960), for the label assignment between the first two annotators.

2.4 Corpus Statistics and Examples

Table 1 shows the frequency and word-length distributions of various discourse modes in the corpus. As we can see that the corpus exhibits an imbalanced distribution across various discourse modes. The most prevalent mode is NAR, which accounts for approximately 42% of the 2000 sentences in the corpus, while the INQ mode has the lowest representation among the sentences.

Mode	**#Frequency (%)**	**Length (word)** (median/mean/std.)
NAR	840 (42.00%)	9/11.47/9.07
INQ	296 (14.93%)	8/9.47/5.42
INF	425 (21.20%)	12/12.50/6.04
SUG	405 (20.36%)	10/11.35/6.80

Table 1: Statistics of various discourse modes in the annotated corpus

3 Classification

3.1 Classical ML Classifier

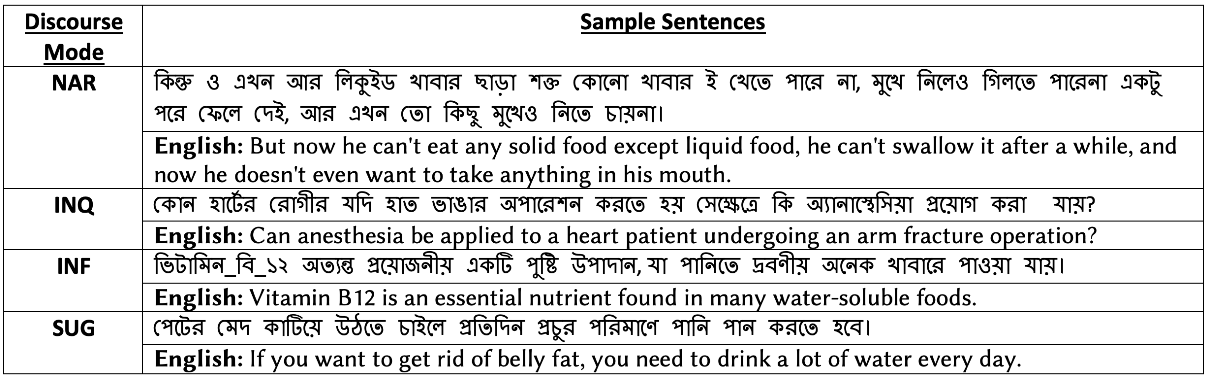
We employ four classical supervised ML classifiers: Logistic Regression (LR), Support Vector Machine (SVM), Random Forest (RF), and Gradient Boosting Tree (GBT) for determining the discourse modes of sentences. For classical ML classifiers, we select word unigrams and bigrams features from the corpus and compute corresponding term frequency-inverse document frequency (tf-idf) scores that act as inputs for the classifiers. For all classifiers, the default parameter settings of the scikit-learn (Pedregosa et al., 2011) library are used with class weight set to *balanced* (when applicable).

3.2 Transformer-based Model

We fine-tune M-BERT (Devlin et al., 2018) language model for categorizing sentences into four classes (i.e., discourse modes): NAR, INQ, INF, and SUG. Since this is a classification task, we utilize the classification module of the M-BERT. The HuggingFace library (Wolf et al., 2019) is used to fine-tune M-BERT. Since the initial layers of M-BERT only learn very general features, we keep them unchanged. Only the last layer of the M-BERT is fine-tuned for our classification task. We tokenize and feed our input training data to fine-tune the M-BERT model; Afterward, the fine-tuned model is used for classifying the testing data. A mini-batch size of 8 and a learning rate of 4×10^{-5} are used. The validation and training split ratio is set to 80% and 20%. The model is optimized using the Adam optimizer (Kingma and Ba, 2014) based on cross-entropy loss. The model is trained for 3 epochs with early stopping criteria set.

---

[3] `https://en.wikipedia.org/wiki/Narration`



Discourse Mode	**Sample Sentences**
NAR	কিন্তু ও এখন আর লিকুইড খাবার ছাড়া শক্ত কোনো খাবার ই খেতে পারে না, মুখে নিলেও গিলতে পারেনা একটু পরে ফেলে দেই, আর এখন তো কিছু মুখেও নিতে চায়না।
	English: But now he can't eat any solid food except liquid food, he can't swallow it after a while, and now he doesn't even want to take anything in his mouth.
INQ	কোন হার্টের রোগীর যদি হাত ভাঙার অপারেশন করতে হয় সেক্ষেত্রে কি অ্যানাস্থেসিয়া প্রয়োগ করা যায়?
	English: Can anesthesia be applied to a heart patient undergoing an arm fracture operation?
INF	ভিটামিন_বি_১২ অত্যন্ত প্রয়োজনীয় একটি পুষ্টি উপাদান, যা পানিতে দ্রবণীয় অনেক খাবারে পাওয়া যায়।
	English: Vitamin B12 is an essential nutrient found in many water-soluble foods.
SUG	পেটের মেদ কাটিয়ে উঠতে চাইলে প্রতিদিন প্রচুর পরিমাণে পানি পান করতে হবে।
	English: If you want to get rid of belly fat, you need to drink a lot of water every day.

Figure 1: Samples sentences representing various discourse modes

Classifier	**Discourse Mode**				
	NAR	**INQ**	**INF**	**SUG**	**Overall**
	P/R/F1	P/R/F1	P/R/F1	P/R/F1	P/R/F1
LR	0.77/0.86/0.81	0.89/0.7/0.78	0.74/0.77/0.76	0.79/0.71/0.75	0.80/0.76/0.78
SVM	0.71/0.91/0.80	0.93/0.60/0.73	0.80/0.70/0.75	0.83/0.66/0.73	0.82/0.72/0.76
RF	0.61/0.94/0.74	0.92/0.55/0.69	0.79/0.41/0.54	0.85/0.52/0.65	0.79/0.61/0.69
GBT	0.71/0.84/0.77	0.83/0.65/0.73	0.72/0.62/0.67	0.80/0.66/0.72	0.76/0.70/0.74
M-BERT	0.77/0.84/0.80	0.94/0.70/0.78	0.77/0.78/0.77	0.78/0.72/0.76	0.82/0.78/0.80

Table 2: Performance of various classifiers for discourse mode classification

Class	NAR	INQ	INF	SUG
NAR	724	19	64	34
INQ	72	202	8	14
INF	73	0	326	26
SUG	76	5	35	289

Table 3: Confusion matrix of LR classifier

4 Results and Discussion

We evaluate the performances of various classifiers based on 5-fold cross-validation and report F1 scores. Table 2 presents F1 scores and accuracies of CML classifiers and transformers-based M-BERT models for sentence mode identification. The results indicate that the LR classifier yields the best performance among the four traditional ML classifiers by achieving an F1 score of around 0.78. SVM performs similarly and achieves an F1 score of about 0.76. The tree-based methods show comparatively inferior performances.

We observe that the performance of CML classifiers is affected by the class distribution of the dataset. All classifiers yield better results for the narrative (NAR) mode as NAR mode represents the highest number of samples (42%) in the dataset. Although the transformer-based multilingual language model yields slightly better performance than the CML classifiers, the improvement is not significant compared to CML classifiers, which can be attributed to the limited amount of labeled data. With more labeled data incorporated, the improvement may be higher as transformer-based models have shown state-of-the-art performances for various NLP tasks across languages.

Table 3 portrays the confusion matrix of the LR classifier from a sample run. We observe that misclassification is affected by the distribution of classes in most cases. Since NAR contains the highest number of samples in the dataset, we notice false negative (FN) predictions of other modes mainly refer to NAR. Nevertheless, for the NAR class, the FN classifications are mostly predicted as INF, even though INF and SUG have a similar number of instances.

5 Summary and Future Work

Developing an effective framework for analyzing social media health data has substantial practical applications. However, such tools require annotated data which is hardly available in a low-resource language like Bengali. Therefore, we introduce a Bengali health corpus created from several Bengali social media health pages. We report detailed annotation guidelines and procedures for the annotation. Moreover, we provide various statistics of four types of discourse modes in the annotated corpus. We make the corpus publicly available for the researchers. Future work will focus on enlarging the size of the annotated corpus. Besides, we will investigate how to leverage cross-lingual resources from other languages, such as English, to improve the performance of this classification task.

6 Ethical statement

Research is based on publicly available data on Facebook. No user personal information is included in the analysis, and no user identity is disclosed.

References

Anietie Andy, Brian Chu, Ramie Fathy, Barrington Bennett, Daniel Stokes, and Sharath Chandra Guntuku. 2021. Understanding social support expressed in a covid-19 online forum. In *Proceedings of the 12th International Workshop on Health Text Mining and Information Analysis*, pages 19–27.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Achyutarama Ganti, Steven Wilson, Zexin Ma, Xinyan Zhao, and Rong Ma. 2022. Narrative detection and feature analysis in online health communities. In *Proceedings of the 4th Workshop of Narrative Understanding (WNU2022)*, pages 57–65.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Corinna Kolárik, Roman Klinger, Christoph M Friedrich, Martin Hofmann-Apitius, and Juliane Fluck. 2008. Chemical names: terminological resources and corpora annotation. In *Workshop on Building and evaluating resources for biomedical text mining (6th edition of the Language Resources and Evaluation Conference)*.

Zachary Levonian, Marco Dow, Drew Erikson, Sourojit Ghosh, Hannah Miller Hillberg, Saumik Narayanan, Loren Terveen, and Svetlana Yarosh. 2021. Patterns of patient and caregiver mutual support connections in an online health community. *Proceedings of the ACM on Human-Computer Interaction*, 4(CSCW3):1–46.

S Anne Moorhead, Diane E Hazlett, Laura Harrison, Jennifer K Carroll, Anthea Irwin, and Ciska Hoving. 2013. A new dimension of health care: systematic review of the uses, benefits, and limitations of social media for health communication. *Journal of medical Internet research*, 15(4):e1933.

Benjamin Olsen and Barbara Plank. 2021. Finding the needle in a haystack: Extraction of informative covid-19 danish tweets. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 11–19.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Salim Sazzed. 2022. Banglabiomed: A biomedical named-entity annotated corpus for bangla (bengali). In *Proceedings of the 21st Workshop on Biomedical Language Processing*, pages 323–329.

Liuping Wang, Dakuo Wang, Feng Tian, Zhenhui Peng, Xiangmin Fan, Zhan Zhang, Mo Yu, Xiaojuan Ma, and Hongan Wang. 2021. Cass: Towards building a social-support chatbot for online health community. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW1):1–31.

Yi-Chia Wang, Robert Kraut, and John M Levine. 2012. To stay or leave? the relationship of emotional and informational support to commitment in online health support groups. In *Proceedings of the ACM 2012 conference on computer supported cooperative work*, pages 833–842.

Yuxi Wang, Martin McKee, Aleksandra Torbica, and David Stuckler. 2019. Systematic literature review on the spread of health-related misinformation on social media. *Social science & medicine*, 240:112552.

Davy Weissenbacher, Abeed Sarker, Arjun Magge, Ashlynn Daughton, Karen O'Connor, Michael Paul, and Graciela Gonzalez. 2019. Overview of the fourth social media mining for health (smm4h) shared tasks at acl 2019. In *Proceedings of the fourth social media mining for health applications (# SMM4H) workshop & shared task*, pages 21–30.

Miaomiao Wen and Carolyn Penstein Rosé. 2012. Understanding participant behavior trajectories in online health support groups using automatic extraction methods. In *Proceedings of the 2012 ACM International Conference on Supporting Group Work*, pages 179–188.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Diyi Yang, Robert Kraut, and John M Levine. 2017. Commitment of newcomers and old-timers to online health support communities. In *Proceedings of the 2017 CHI conference on human factors in computing systems*, pages 6363–6375.

Emotion and Sentiment Guided Paraphrasing

Justin J. Xie[*]
Westview High School
Portland, OR, USA
`justinjxie@gmail.com`

Ameeta Agrawal
Portland State University
Portland, OR, USA
`ameeta@pdx.edu`

Abstract

Paraphrase generation, a.k.a. paraphrasing, is a common and important task in natural language processing. Emotional paraphrasing, which changes the emotion embodied in a piece of text while preserving its meaning, has many potential applications, including moderating online dialogues and preventing cyberbullying. We introduce a new task of fine-grained emotional paraphrasing along emotion gradients, that is, altering the emotional intensities of the paraphrases in fine-grained settings following smooth variations in affective dimensions while preserving the meaning of the original text. We reconstruct several widely used paraphrasing datasets by augmenting the input and target texts with their fine-grained emotion labels. Then, we propose a framework for emotion and sentiment guided paraphrasing by leveraging pre-trained language models for conditioned text generation. Extensive evaluation of the fine-tuned models suggests that including fine-grained emotion labels in the paraphrase task significantly improves the likelihood of obtaining high-quality paraphrases that reflect the desired emotions while achieving consistently better scores in paraphrase metrics such as BLEU, ROUGE, and METEOR.

1 Introduction

With the rise of social media and online chat rooms, the textual aspect of language is often found to be the only aspect of communication that is transferred over the Internet. Devoid of any intonations or accompanying facial movements, it is more challenging for people to decipher the true meaning and underlying emotion that a message is intended to convey, especially if that message incorporates the more complex aspects of speech. This could lead to negative social consequences. For example, political tweets from prominent figures without careful consideration can lead to political radicalization and conflicts. Furthermore, on messaging apps such as Discord, cyberbullies attack others with emotion-ladened words while innocent people send unnecessarily emotional messages in the heat of the moment. Emotional paraphrasing could be an important solution to overly intense emotions expressed on social media (Seehausen et al., 2012) and provide support toward moderation of hate speech (Tontodimamma et al., 2021; Altarawneh et al., 2023).

Paraphrase generation (a.k.a. paraphrasing), a key task in natural language processing, involves generating an output text that preserves the meanings of the input text while including variations in words and grammars. The refined task of emotional paraphrasing has garnered much recent attention (Casas et al., 2021). Its goal is to alter the underlying emotion associated with a sentence while maintaining its meaning.

In this paper, we introduce a new task of *fine-grained* emotional paraphrasing along emotion gradients, i.e., altering emotional intensities in fine grain following smooth variations in affective dimensions (e.g., from anger to annoyance) while preserving the overall meaning. First, we analyze and reconstruct existing paraphrasing datasets to adapt them for the current task. Next, we propose the concept of an emotion-transition graph where transitions are based on the fine-grained emotions and their emotion gradients as identified by GoEmotions (Demszky et al., 2020), and are constrained by specific goals of emotion transition. Then, we develop a framework for emotion and sentiment guided paraphrasing by leveraging several pretrained language models for conditioned text generation under zero-shot, few-shot, and fully supervised settings. Lastly, we conduct a comprehensive evaluation of the proposed framework with several datasets using metrics pertaining to both paraphrasing and emotion transition.

---

[*] Work done as a research intern at Portland State University.

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 58–70
July 14, 2023 ©2023 Association for Computational Linguistics

Dataset	Transition	Input Text	Paraphrased Text
`Google`	anger → disappointment	*He is angry to learn that in June Ethan Lovett (Nathan Parsons) is his half brother.*	*He is upset to learn in June that Nathan Parsons (Ethan Lovett) is his half brother.*
`MRPC`	approval → realization	*The decision was among the most significant steps toward deregulation undertaken during the Bush administration.*	*The decision is among the far-reaching deregulatory actions made during the Bush administration.*
`Quora`	fear → nervousness	*My boyfriend wants to kiss me and I kind of want to kiss him, but I've never kissed anyone and I'm scared I'll be terrible at it. What should I do?*	*My boyfriend is wanting to kiss me and I want to kiss him too, but I've never kissed anyone, and I'm nervous. What do I do?*

Table 1: Some sample instances of emotion paraphrasing from our reconstructed datasets.

In all settings, the fully supervised and few-shot fine-tuned models showed significant improvements over the zero-shot base models, i.e., doubling the number of exact matches of desired fine-grained emotions while achieving consistently better scores in paraphrase metrics such as BLEU, ROUGE, and METEOR. Few-shot learning delivered competitive performances in all categories compared with fully-supervised. This study indicates that our fine-grained emotional paraphrasing framework has potentials in applications to specific scenarios, e.g., chat rooms, forums, and public online spaces.

Specifically, our contributions include:

- Reconstructed Emotion Paraphrase Datasets: Given existing paraphrase datasets, we apply a fine-grained emotion classification model to label the input text and target text of each paraphrase pair with their emotions (see examples in Table 1). A similar procedure is also applied to label each paraphrase pair with their sentiment intensities: neutral, low, or high.
- Emotional Paraphrasing Models: Leveraging pre-trained language models, we propose a paraphrasing framework guided by emotion and sentiment transitions.
- Evaluation: We conduct an extensive set of experiments to verify the effectiveness of the proposed approach.

2 Related Work

This section discusses two main threads of related work: emotion classification and paraphrasing.

2.1 Emotion Psychology and Classification

Emotions are a key component of human psychology, playing a role in many cognitive processes including learning, memory, decision making, and interpersonal communication (Oatley and Duncan, 1994; Tyng et al., 2017). Equally important is the role that emotions play in human-to-human interactions. Words can trigger emotional responses, both negative and positive. Without facial expressions, vocal intonations, or hand gestures, it is harder to communicate one's emotions online. The intensities of words can be higher than what someone wants them to communicate. For example, someone could want to communicate frustration, but instead could come off as furious. Rooted in the psychology of communication and emotion, the need for lowering intensity of online communications inspires the task of fine-grained emotional paraphrasing.

In 1890, James et al. proposed fear, grief, love, and rage as a set of the most basic emotions. Then, Plutchik (1980) introduced eight categories of emotions, which was followed by Ekman (1992) who introduced his famous set of six basic emotions: fear, anger, joy, sadness, disgust, and surprise. These taxonomies form the basis of many early NLP experiments pertaining to emotions (Mohammad and Turney, 2010; Agrawal and An, 2012). Another classification produced by Lazarus and Lazarus (1994) included a list of 15 emotions. Recently a study done by Cowen and Keltner (2017) expanded on these classifications. By having human test subjects report on the emotions they felt while viewing videos, the study found that there were 27 emotion categories, in addition to a neutral emotion. This study also grouped these emotions into “clusters.” Demszky et al. (2020) produced a similar set of 28 emotions that was used in the GoEmotions project. This project provided a labeled dataset of 58K texts and a model based on BERT (Devlin et al., 2018) capable of classifying inputs into one of the 28 emotions. In addition, the GoEmotions project provided a heatmap showing the adjacency between emotions by continuous gradients as well as including a stratification of the emotions into groups (see Appendix A). While the proposed approach can adopt any emotion taxonomy, our work follows the GoEmotions groups as guidance for structuring the proposed emotion transition graph.

2.2 Paraphrasing

Paraphrasing involves changing the wording of an input text while preserving its original meaning. Several stuides combine deep generative models with other modeling and training techniques: e.g., variations using reinforcement learning (Li et al., 2017), long short-term memory or LSTM (Gupta et al., 2018), and stacked residual LSTM (Prakash et al., 2016). Transformer-based text-to-text models such as BART (Lewis et al., 2019) and T5 (Raffel et al., 2020) have become more popular for paraphrasing. Several studies have been conducted to improve these models’ paraphrasing performance through combining Transformers and sequence-to-sequence models (Egonmwan and Chali, 2019) and joint paraphrase learning (Min et al., 2020).

Emotional paraphrasing, a task that alters the underlying emotion associated with the input sentence while maintaining its meaning, has been closely studied. Casas et al. (2021) fine-tuned six GPT models (one for each emotion) for emotional paraphrasing, where the input text was paraphrased to fit one of Ekman’s six emotional categories. Our new task, instead, stipulates a more fine-grained emotion categorization and paraphrasing. Our fine-tuned language models conduct emotional transitions based on the emotion of the input text, and is capable of transitioning to various emotions along emotion gradients on a transition graph.

Our task is also related to emotion or sentiment text style transfer. Sundararaman et al. (2020) proposed an unsupervised aspect-level approach to sentiment controllable style transfer. Other studies include a delete-retrieve-generate approach (Li et al., 2018) and a mask-infill approach (Wu et al., 2019) to sentiment style transfer. Through masked language modeling and transfer learning, MohammadiBaghmolaei and Ahmadi (2023) adapted style transfer to transform texts into one of four emotions: anger, fear, sadness, and joy. While these tasks transfer text following certain emotion or sentiment styles, our task focuses on more flexible fine-grained emotion and sentiment transitions.

As our task lowers emotion intensity of input texts, thereby lowering the strong psychological effects that intense emotional interactions can bring, it also relates to the task of positive reframing (Ziems et al., 2022). Both focus on altering the emotions of texts, while preserving its underlying connotations. However, the task of positive reframing emphasizes altering the input text into a positive emotion while our task does not transit every emotion into a positive one, but rather lowers the intensities of emotions, which allows negative and positive emotions alike. Our goal of lowering the intensity of emotion in text is related to, but different from the task of neutralizing bias (Pryzant et al., 2020). Neutralizing bias strives to eliminate all bias, which results in most paraphrased texts being classified as *neutral*. Our task aims to preserve the base meaning and tone while lowering the *intensity* of the emotion in the input text. Thus, the paraphrase still expresses its original view or belief, but in a less provocative or intense manner.

3 Fine-Grained Emotional Paraphrasing

3.1 Problem Description

Given an input text t_i with emotion e_i where e_i belongs to an emotion adjacency group $\mathcal{E}$: $e_i \in \mathcal{E}$, the task of fine-grained emotional paraphrasing along emotion gradients is to paraphrase t_i into t_f where the emotion of t_f is e_f and $(e_f \in \mathcal{E}) \cap (e_i! = e_f)$. Further constraints help to guide the emotion transitions along a specific affective dimension, e.g., lowering the sentiment intensity. If the intensity of e_i is s_i and that of e_f is s_f, the refined condition is $(e_f \in \mathcal{E}) \cap (e_i! = e_f) \cap (s_f < s_i)$.

To tackle the task of fine-grained emotional paraphrasing along emotion gradients, we propose a novel framework as illustrated in Figure 1. The top part of this workflow fine-tunes pre-trained language models into fine-grained emotional paraphrasers. First, it labels the emotions of input and target texts of each paraphrase pair in both train and test sets. Then for each pair, a prefix of the

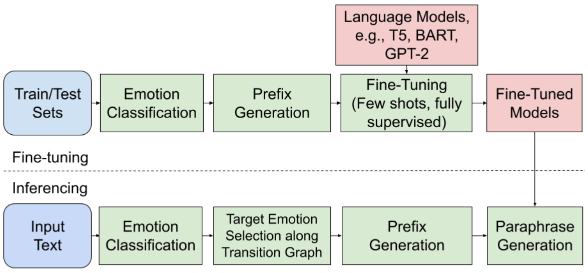


Figure 1: Workflow of Fine-Grained Emotional Paraphrasing along Emotion Gradients

form "(input emotion) to (target emotion)" is generated. Finally, the train/test sets augmented with emotion transition prefixes are utilized to fine-tune language models, e.g., T5, BART, and GPT-2, under three settings: zero-shot, few-shot, and fully supervised. The bottom of this workflow utilizes the fine-tuned paraphrasing models in inference applications. Given an input text t_i, it first identifies the emotion e_i of t_i. Then it selects a target emotion e_f for paraphrasing, utilizing an emotion transition graph that is based on emotion gradients. After that, it generates a prefix for the selected emotion transition "e_i to e_f". Finally, it sends the query, "e_i to e_f: t_i" to our fine-tuned paraphraser to generate the target paraphrase t_f.

3.2 Emotion Classification

The first step in our workflow is to identify the emotion (e_i) of the input text (t_i). This is done through our enhanced version of the GoEmotions model: we modified the model to only report the dominant emotion that is above a certain threshold. If no emotion meets the threshold, the model reports no emotion label. Given the input text t_i, this classification model identifies the most compatible of the 28 emotions (e_f) to feed into the transition graph. The GoEmotions model has a wider variety and more detailed array of emotions compared to emotion classifications such as Ekman's. This allows for more precise emotion classifications that enable fine-grained adjustment of paraphrase emotions.

3.3 Target Emotion Selection Using Emotion Transition Graph

The second step in our workflow is target emotion selection using an emotion transition graph such as the one shown in Figure 2. This particular transition graph is intended for lowering sentiment intensity. It is based on the GoEmotions emotion heatmap created by Demszky et al., which shows emotions as grouped by continuous gradients. Each group of emotions (as shown in Appendix A), although close in sentiments, exhibits different levels of intensities. To measure the sentiment intensities of different emotions, we have applied NLTK's Vader Score (Hutto and Gilbert, 2014) function to all emotion-labeled texts from the GoEmotions dataset and computed the median score for each emotion (which can be found in Appendix A). Based on the median Vader scores of the 28 emotions, we are able to group them into five groups: high negative, low negative, neutral, low positive, and high positive as shown in Table 2.

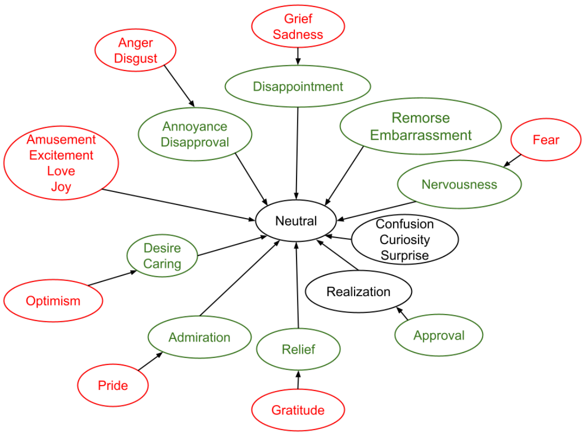


Figure 2: Sentiment Intensity Lowering Emotion Transition Graph: From High (Red) to Low (Green) to Neutral

Group	Emotions
`high negative`	anger, disgust, grief, fear, sadness
`low negative`	nervousness, annoyance, disappointment, embarrassment, remorse, disapproval
`neutral`	confusion, curiosity, realization, surprise, neutral
`low positive`	approval, caring, desire, relief
`high positive`	amusement, excitement, pride, optimism, gratitude, joy, admiration, love

Table 2: Emotion Grouping by Sentiment Intensity

The emotion transition graph in Figure 2 is derived by combining the two groupings found in GoEmotions and Table 2. The emotions in red are emotions of high sentiment intensities, positive or negative, those in green are of low sentiment intensities, and those in black have neutral sentiment intensities. The arrows between ovals indicate the emotions in these ovals belong to the same GoEmotions emotion clusters, i.e., they are adjacent and connected with continuous gradients. The arrows to the neutral oval indicate that all emotions can transit to the neutral emotion. By following the transition graph, we can adjust emotion intensity. For example, if the GoEmotions model identifies the input emotion as "anger," the transition graph may recommend a transition to "annoyance."

3.4 Prefix Generation

The third step in our workflow is prefix generation. We adopt the multi-task design for text-to-text generation, i.e., many NLP tasks can be cast as text-to-text tasks and a prefix can be added to the input text to indicate the task at hand. Our prefix generator utilizes this design and generates the prefix for the task of fined-grained emotional paraphrasing. Given the source emotion e_i identified in the emotion classification step and the target emotion e_f selected in the target emotion selection step, the prefix is generated in the format of "e_i to e_f" and placed in front of the input text t_i. It guides the fine-tuned language models to paraphrase along the selected emotion transition. An example of such a prefix would be: *"anger to disappointment: He is angry to learn that in June Ethan Lovett (Nathan Parsons) is his half brother."*

In addition, we also explore the use of sentiment ranges (i.e., high positive, low positive, neutral, low negative, and high negative) in place of fine-grained emotion labels as alternative fine-grained prefixes. Such a prefix would look like: *"high_neg to low_neg: He is angry to learn that in June Ethan Lovett (Nathan Parsons) is his half brother."*

3.5 Paraphrase Generation

The final step of our workflow is paraphrase generation which utilizes a fine-tuned language model to complete the task of fine-grained emotion paraphrasing along emotion gradients. Such a model is fine-tuned with a dataset of paraphrase pairs that exemplify the transitions along the continuous gradients that connect the emotions. The fine-tuned model allows for precise emotional paraphrasing by inputting the emotion transition prefix and the original text, paraphrasing it, and outputting the paraphrase that best fits the target emotion.

4 Experiments

Figure 3 illustrates the workflow of our experiments on preparing the train/test datasets for fine-grained emotional paraphrasing, conducting fine-tuning on various language models, and evaluating the emotional paraphrasing performance of these models.

- Given a paraphrase dataset, we first label the input text and target text of each paraphrase pair with fine-grained emotions by using our modified verison of GoEmotions model.

- Second, we remove the paraphrase pairs that have the same input/target emotions and those pairs whose input or target emotions are labeled as neutral, as we are focused on the paraphraser's ability to lower the emotional intensity instead of neutralizing it.

- Third, we select the paraphrase pairs with decreasing intensity and if a pair has increasing intensity, we flip its input/target texts and emotions, so it can be used in our experiment.

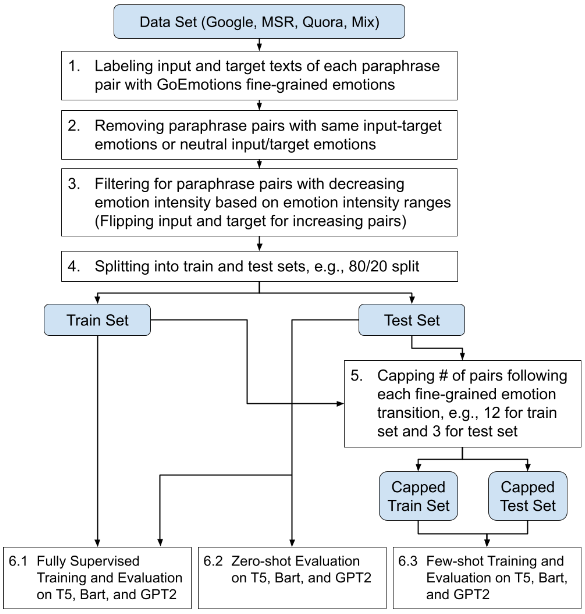

Data Set (Google, MSR, Quora, Mix)
1. Labeling input and target texts of each paraphrase pair with GoEmotions fine-grained emotions
2. Removing paraphrase pairs with same input-target emotions or neutral input/target emotions
3. Filtering for paraphrase pairs with decreasing emotion intensity based on emotion intensity ranges (Flipping input and target for increasing pairs)
4. Splitting into train and test sets, e.g., 80/20 split
Train Set
Test Set
5. Capping # of pairs following each fine-grained emotion transition, e.g., 12 for train set and 3 for test set
Capped Train Set
Capped Test Set
6.1 Fully Supervised Training and Evaluation on T5, Bart, and GPT2
6.2 Zero-shot Evaluation on T5, Bart, and GPT2
6.3 Few-shot Training and Evaluation on T5, Bart, and GPT2

Figure 3: Experiment Workflow

- Fourth, we split the dataset into train/test sets, e.g., with a 80/20 split.

- Fifth, an optional step for few-shot training, we cap the number of instances of the same emotion transition, e.g., 12 in the train set and 3 in the test set following the 80/20 split.

Dataset	Total # of Pairs	Emotion Transiting w/ Neutral	Emotion Transiting w/o Neutral	Sentiment Intensity Lowering
PAWS	57401	3593	432	395
MRPC	3728	508	53	32
Quora	149263	32866	16935	2401
Mix	210392	36967	17420	2828

Table 3: Dataset Statistics

- Sixth, we conduct three types of fine-tuning: fully supervised (or full), zero-shot, and few-shot and compare the performances of each type of fine-tuned model. In the zero-shot case, we directly evaluate the original model without fine-tuning and in the few-shot case, we fine-tune the model with the capped datasets as in Step 5 and evaluate with the full test set.

4.1 Datasets

Three publicly available paraphrasing datasets were used in our experiments after reconstruction. These include **Google PAWS-Wiki** (PAWS) (Zhang et al., 2019), **Microsoft Research Paraphrase Corpus** (MRPC) (Dolan and Brockett, 2005), and **Quora Questions Pairs** (Iyer et al., 2017).

The Google PAWS project produced multiple sets of paraphrasing input-output pairs. We chose to use to PAWS-Wiki Labeled (Final) data because they were generated by translation methods and human verified for accuracy. The MRPC corpus was a compilation of human-annotated data from the news. The Quora corpus has the goal of aiding the training of "semantic equivalence" models, similar to the goals of paraphrasing models. Some sample instances are presented in Table 1.

To make these datasets suitable for our emotional paraphrasing task, we reconstructed them by following Steps 1-4 in Figure 3. The statistics of the filtered datasets are shown in Table 3, and these datasets are also combined into a Mix dataset for the study of overall performance.

4.2 Evaluation Metrics

The emotional paraphrasing capabilities of the models are evaluated from two aspects: **emotion transition** and **paraphrasing**.

To evaluate the emotion transition performance of the models, we utilize the *Exact* metric to compute two scores: *Exact-SR* and *Exact-FE*. The *Exact-SR* score measures the percentage of the emotion sentiment ranges (i.e., high positive, low positive, neutral, low negative, and high negative) of the generated paraphrases that match the target sentiment ranges. The *Exact-FE* score measures the percentage of the fine-grained emotions of the generated paraphrases that match the target emotions. By comparing the sentiment ranges and specific emotions of the target texts and the predictions of each model, the *Exact* scores indicate how capable a model is at emotion transitioin.

To evaluate the paraphrasing capabilities of the models, we utilize several metrics: *BLEU* (Papineni et al., 2002), *ROUGE-L* (Lin, 2004), and *METEOR* (Banerjee and Lavie, 2005). They evaluate the similarities of target texts and model predictions.

4.3 Models

Below we discuss our models and training settings.

Emotion Labeling. The original GoEmotions model, for each input text, outputs a list of emotions that it identified as being "possible" candidates for the emotion of the input text and a confidence score for each candidate. In our experiments, we modified the model to only report the dominant emotion with a confidence score over 0.5.

Paraphrasing. For paraphrasing, we fine-tuned 3 pre-trained language models, T5, BART, and GPT-2. We adopted multi-task training. Let t_i be the input text and e_i be its emotion. Let e_f be the target emotion, and t_f be the emotional paraphrased output of t_i. In the task of fine-grained emotional paraphrasing along emotion gradients, t_i, e_i, and e_f are given to the language model in the query format: "e_i to e_f: t_i". The fine-tuned model will output t_f, a paraphrased version of t_i where the underlying semantics of t_i is kept and the intensity of emotion is changed. Each model is trained under 3 settings: fully supervised, few-shot, and zero-shot.

4.4 Implementation

We utilized the Simple Transformers package (Rajapakse, 2023) Version 0.63.6 to fine-tune T5 and BART models. For GPT-2, we utilized Huggingface's transformers implementation (HuggingFace, 2023) Version 4.25.1. We conducted fine-tuning and evaluation on a desktop with an AMD Ryzen 7 5800x, 32GB RAM, and RTX 3080TI GPU. Due to a limited amount of GPU memory, 12GB precisely, we had to adopt a smaller batch size of 6. Each model was fine-tuned over 3 epochs.

			Emotion-Transition		**Paraphrasing**		
	Training	**Prefix Type**	Exact-SR	Exact-FE	BLEU	R-L	METEOR
T5	Full	Sentiment Ranges	**0.796**	**0.632**	0.314	0.557	0.571
		Fine-grained Emotions	**0.801**	**0.604**	0.316	0.555	0.572
	Few-Shot	Sentiment Ranges	0.791	0.620	0.298	0.528	0.547
		Fine-grained Emotions	0.698	0.534	0.301	0.538	0.561
	Zero-Shot	Sentiment Ranges	0.450	0.349	0.248	0.484	0.515
		Fine-grained Emotions	0.468	0.307	0.244	0.488	0.513
BART	Full	Sentiment Ranges	0.719	0.606	**0.408**	**0.626**	**0.663**
		Fine-grained Emotions	0.706	0.578	**0.409**	**0.619**	**0.665**
	Few-Shot	Sentiment Ranges	0.719	0.606	0.408	0.626	0.663
		Fine-grained Emotions	0.706	0.578	0.409	0.619	0.665
	Zero-Shot	Sentiment Ranges	0.291	0.339	0.335	0.588	0.633
		Fine-grained Emotions	0.290	0.237	0.335	0.588	0.633
GPT-2	Full	Sentiment Ranges	0.691	0.494	0.168	0.381	0.399
		Fine-grained Emotions	0.649	0.471	0.164	0.387	0.407
	Few-Shot	Sentiment Ranges	0.668	0.461	0.150	0.371	0.391
		Fine-grained Emotions	0.639	0.452	0.178	0.389	0.408
	Zero-Shot	Sentiment Ranges	0.632	0.113	0.004	0.094	0.124
		Fine-grained Emotions	0.593	0.080	0.005	0.091	0.117

Table 4: Evaluations of T5, BART, and GPT-2 for Fine-Grained Emotional Paraphrasing

5 Results and Discussions

Table 4 summarizes the results from our experiments using T5, BART, and GPT-2 models for the fine-grained emotional paraphrasing task. It can be observed for all three models, fully supervised fine-tuning significantly outperformed the zero-shot setting in every category in both emotion-transition and paraphrasing metrics. Few-shot fine-tuning delivered competitive performances in all categories compared with the fully supervised setting.

When comparing model performance, it can be observed that T5 outperforms BART and GPT-2 on emotion-transition. This may be attributed to T5's design as a multi-task model meant to accept the prefixes we utilized. For paraphrasing, BART outclassed both T5 and GPT-2 models in text similarity and consistency. We speculate that designing more appropriate prompts might benefit GPT-2.

In few-shot fine-tuning, we experimented with different limits for the numbers of text pairs following each fine-grained emotion transition in the train/test sets, 4/1, 8/2, 12/3, 16/4, and 20/5 per the 80/20 split. All few-shot train/test sets delivered better emotional transition performance than zero-shot and their paraphrasing performance became consistently better with 12/3 split and above.

One important takeaway from the results is the similarity in performance of using sentiment ranges or fine-grained emotions as part of the prefix prompt to the models. We noticed that there was an insignificant difference in both the emotion-transition and paraphrasing performances of the two prefix types. An explanation for this behavior in the fine-tuned models may be that emotion transitions largely follow continuous gradients among emotions along certain affective dimensions and, therefore, lowering the sentiment intensity from an emotion often transitions to a same target emotion. This means that although the prefixes are different, the models learn the same emotion transitions that are embodied in the paraphrase pairs.

Figure 4 illustrates the success rates of T5 in transitioning texts between different sentiment intensity levels under different fine-tuning settings. We observe that fully supervised and few-shot fine-tuning both outperform zero-shot significantly in all sentiment intensity lowering transitions. Fully supervised seems to perform better in emotion transitions lowering positive sentiments while few-shot better in lowering negative sentiments. Importantly, we also observe that lowering from high positive or negative to low positive or negative is more challenging for the model than lowering to neutral level.

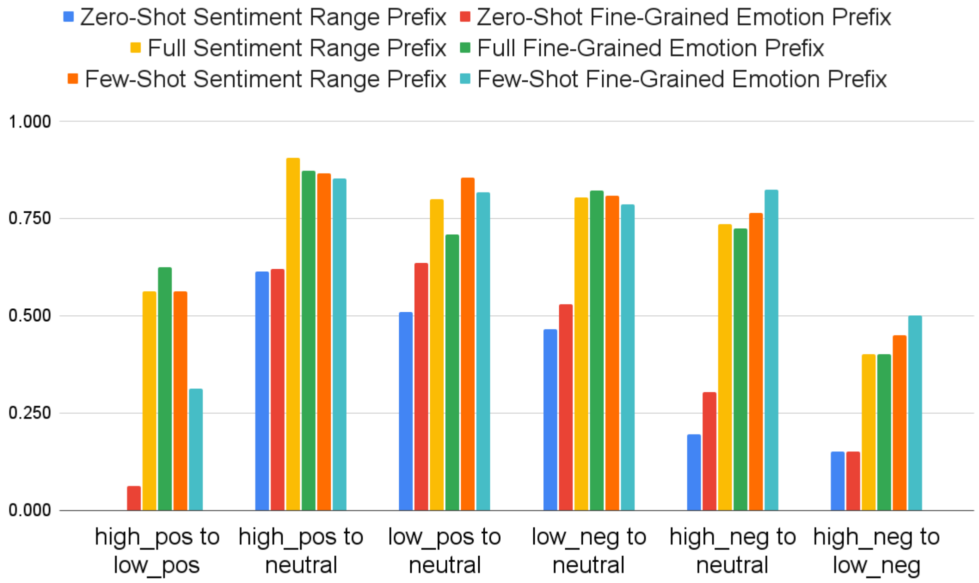


Figure 4: Success Rates of T5 in Transitioning Sentiment Intensity Levels on Mix Dataset

6 Case Study on Transition Graph Guided Target Emotion Selection

We created a new test dataset from the original Mix test dataset by leveraging transition-graph-guided emotion selection. Instead of utilizing the target emotion provided by the original test dataset, the transition-graph was used to randomly select a new target emotion that would maintain the emotion proximity while lowering the emotional intensity. However, if the neutral emotion was selected, the original target emotion was kept. In doing so, 35 percent of the dataset was given a larger variety of transition types between the high, low, and neutral emotion groups, while the size of the dataset was maintained. The emotion of the model prediction was compared to the desired target emotion to evaluate emotion-transition performance. The model prediction was compared to the original target text for measuring paraphrasing performance.

Figure 5 shows the performances of zero-shot and fully supervised fine-tuned T5 models on this new test dataset. They continue to reflect the observation from Table 4 that the fine-tuned models show major improvements in emotion transition, while maintaining a slight gain in paraphrasing performance. With the increased variety of target emotions, the success rate of the models does decrease as indicated by the lower *Exact* metrics. This points to the neccesity of paraphrase datasets that provide better coverage of the emotion transition graph which helps automate the target emotion selection for practical emotion moderation applications.

7 Conclusions and Future Work

In this paper, we introduced a new task of fine-grained emotional paraphrasing along emotion gradients. We developed a workflow for addressing this task by fine-tuning pre-trained language models under multi-task learning framework. Our experiments have demonstrated that fine-tuned models perform significantly better than baseline models in both emotion transition and paraphrasing.

For future work, there is still much to improve for fine-grained emotional paraphrasing. We will pursue better datasets for emotional fine-tuning or even develop new datasets for this purpose. We will further develop our approach on top of the state-of-the-art large language models, e.g., GPT-4. We will also investigate more customized models beyond the baseline language models. For evaluation, we plan to conduct human studies as appropriate.

Limitations

There is no dataset currently available specific for fine-grained emotional paraphrasing. For our study, we have to utilize publicly available paraphrase datasets, Google PAWS, MRPC, and Quora and augment their text pairs with emotions labels. These datasets may not be best suited for study-

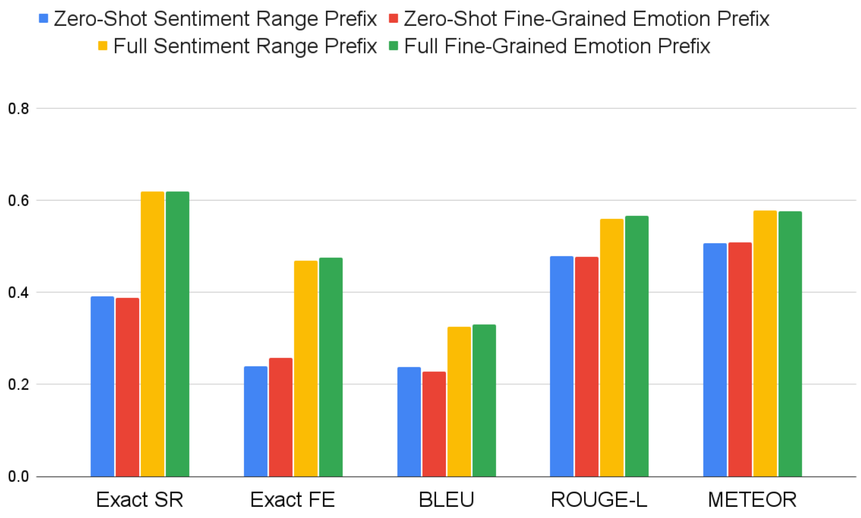


Figure 5: Fine-Tuned T5 Models on Test Dataset Enhanced by Transition-Graph-Guided Emotion Selection

ing this new task. Therefore, new datasets that are particularly developed for fine-grained emotional paraphrasing are needed. Furthermore, it is also desirable to evaluate the proposed methods in alternative application scenarios other than lowering sentiment intensity.

When using GoEmotions as our fine-grained emotion classifier, we selected the emotion with the dominant confidence score above the threshold of 0.5. As the authors of GoEmotions have pointed out, there is still much room to improve on the classification accuracy of GoEmotions. Although the confidence score threshold of 0.5 worked well in our experiments, how to set this threshold still requires more studies. Similarly we utilized NLTK's Vader scores to place emotions into high, low, and neutral intensity groups. The Vader score thresholds for this grouping were selected empirically. Further studies are needed for setting the thresholds or developing better ways for intensity grouping.

In the evaluation of our fine-grained emotional paraphrasing models, we utilized two sets of metrics for emotion transition and paraphrasing respectively. It is desirable to jointly evaluate these two aspects, which we believe would be best done by well-designed human studies in future work.

Ethics Statement

Our study is based on publicly available datasets from reputable sources. The augmented datasets will be made available with open-source code release. The fine-grained emotional paraphraser obtained through our study is based on existing pre-trained language models and paraphrase datasets; therefore, it may inherit their drawbacks such as undesirable social biases. As an unintended use, the methods proposed by this paper can be utilized or modified to produce paraphrasers that increase the emotional intensities of texts, leading to texts with extreme emotions that can be potentially harmful. While we advocate for voluntary adoption of emotion moderation to achieve more peaceful cyberspaces, we do realize that the proposed methods can be abused as emotion moderation tools for censorship. We strongly oppose such applications.

Acknowledgments

We thank the anonymous reviewers for their helpful feedback.

References

Ameeta Agrawal and Aijun An. 2012. Unsupervised emotion detection from text using semantic and syntactic relations. In *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, volume 1, pages 346–353.

Enas Altarawneh, Ameeta Agrawal, Michael Jenkin, and Manos Papagelis. 2023. Conversation derailment forecasting with graph convolutional networks. In *The 7th Workshop on Online Abuse and Harms (WOAH)*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Jacky Casas, Samuel Torche, Karl Daher, Elena Mugellini, and Omar Abou Khaled. 2021. Emotional paraphrasing using pre-trained language models. In *2021 9th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*, pages 01–07. IEEE.

Alan S Cowen and Dacher Keltner. 2017. Self-report captures 27 distinct categories of emotion bridged by continuous gradients. *Proceedings of the national academy of sciences*, 114(38):E7900–E7909.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. Goemotions: A dataset of fine-grained emotions. *arXiv preprint arXiv:2005.00547*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Bill Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Third International Workshop on Paraphrasing (IWP2005)*.

Elozino Egonmwan and Yllias Chali. 2019. Transformer and seq2seq model for paraphrase generation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 249–255.

Paul Ekman. 1992. An argument for basic emotions. *Cognition & emotion*, 6(3-4):169–200.

Ankush Gupta, Arvind Agarwal, Prawaan Singh, and Piyush Rai. 2018. A deep generative framework for paraphrase generation. In *Proceedings of the aaai conference on artificial intelligence*, volume 32.

HuggingFace. 2023. GPT-2 Implementation.

C. Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. *Proceedings of the International AAAI Conference on Web and Social Media*, 8(1):216–225.

Shankar Iyer, Nikhil Dandekar, and Kornél Csernai. 2017. First quora dataset release: Question pairs.

William James, Frederick Burkhardt, Fredson Bowers, and Ignas K Skrupskelis. 1890. *The principles of psychology*, volume 1. Macmillan London.

Richard S Lazarus and Bernice N Lazarus. 1994. *Passion and reason: Making sense of our emotions*. Oxford University Press, USA.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: a simple approach to sentiment and style transfer. *arXiv preprint arXiv:1804.06437*.

Zichao Li, Xin Jiang, Lifeng Shang, and Hang Li. 2017. Paraphrase generation with deep reinforcement learning. *arXiv preprint arXiv:1711.00279*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

So Yeon Min, Preethi Raghavan, and Peter Szolovits. 2020. Advancing seq2seq with joint paraphrase learning. In *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, pages 269–279.

Saif Mohammad and Peter Turney. 2010. Emotions evoked by common words and phrases: Using mechanical turk to create an emotion lexicon. In *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*, pages 26–34.

Rezvan MohammadiBaghmolaei and Ali Ahmadi. 2023. Tet: Text emotion transfer. *Knowledge-Based Systems*, 262:110236.

Keith Oatley and Elaine Duncan. 1994. The experience of emotions in everyday life. *Cognition & emotion*, 8(4):369–381.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Robert Plutchik. 1980. A general psychoevolutionary theory of emotion. In *Theories of emotion*, pages 3–33. Elsevier.

Aaditya Prakash, Sadid A Hasan, Kathy Lee, Vivek Datla, Ashequl Qadir, Joey Liu, and Oladimeji Farri. 2016. Neural paraphrase generation with stacked residual lstm networks. *arXiv preprint arXiv:1610.03098*.

Reid Pryzant, Richard Diehl Martinez, Nathan Dass, Sadao Kurohashi, Dan Jurafsky, and Diyi Yang. 2020. Automatically neutralizing subjective bias in text. In *Proceedings of the aaai conference on artificial intelligence*, volume 34, pages 480–489.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Thilina Rajapakse. 2023. Simple transformers.

Maria Seehausen, Philipp Kazzer, Malek Bajbouj, and Kristin Prehn. 2012. Effects of empathic paraphrasing – extrinsic emotion regulation in social conflict. *Frontiers in Psychology*, 3.

Mukuntha Narayanan Sundararaman, Zishan Ahmad, Asif Ekbal, and Pushpak Bhattacharyya. 2020. Unsupervised aspect-level sentiment controllable style transfer. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 303–312.

Alice Tontodimamma, Eugenia Nissi, Annalina Sarra, and Lara Fontanella. 2021. Thirty years of research into hate speech: topics of interest and their evolution. *Scientometrics*, 126:157–179.

Chai M Tyng, Hafeez U Amin, Mohamad NM Saad, and Aamir S Malik. 2017. The influences of emotion on learning and memory. *Frontiers in psychology*, 8:1454.

Xing Wu, Tao Zhang, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. " mask and infill": Applying masked language model to sentiment transfer. *arXiv preprint arXiv:1908.08039*.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. Paws: Paraphrase adversaries from word scrambling. *arXiv preprint arXiv:1904.01130*.

Caleb Ziems, Minzhi Li, Anthony Zhang, and Diyi Yang. 2022. Inducing positive perspectives with text reframing. *arXiv preprint arXiv:2204.02952*.

A Appendix

Group	Emotions
1	neutral
2	amusement, excitement, joy, love
3	optimism, desire, caring
4	pride, admiration
5	gratitude, relief
6	approval, realization
7	surprise, curiosity, confusion
8	fear, nervousness
9	remorse, embarrassment
10	disappointment, sadness, grief
11	disgust, anger, annoyance, disapproval

Table 5: Emotion Grouping by Demszky et al. (2020)

Emotions	Median Vader Score
grief	-0.5423
anger	-0.5234
disgust	-0.51805
fear	-0.4404
sadness	-0.4404
nervousness	-0.3597
disappointment	-0.3059
annoyance	-0.296
embarrassment	-0.26655
remorse	-0.0772
disapproval	-0.0644
confusion	0
curiosity	0
realization	0
surprise	0
neutral	0
approval	0.296
caring	0.3412
desire	0.4019
relief	0.4391
amusement	0.4404
excitement	0.4404
pride	0.4767
optimism	0.5081
gratitude	0.5574
joy	0.6008
admiration	0.6249
love	0.6369

Table 6: Sentiment Intensities of Emotions by NLTK Vader Scores Computed on GoEmotions Dataset

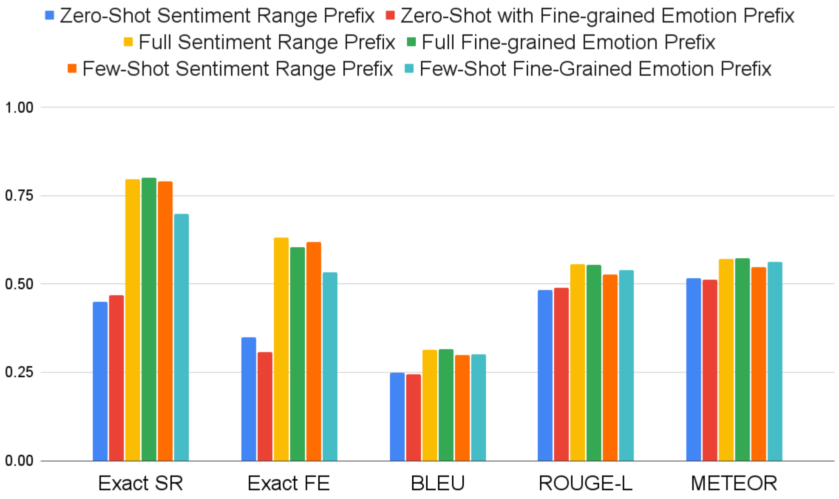


Figure 6: Evaluation Results of Mix Dataset on T5

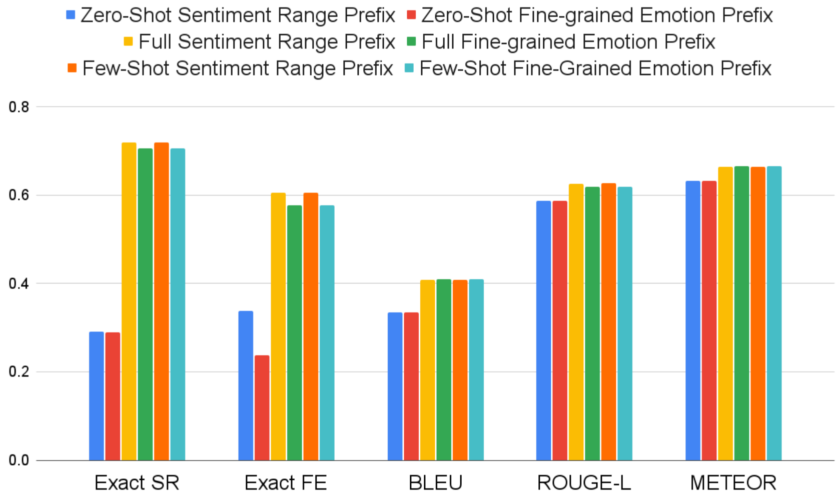


Figure 7: Evaluation Results of Mix Dataset on BART

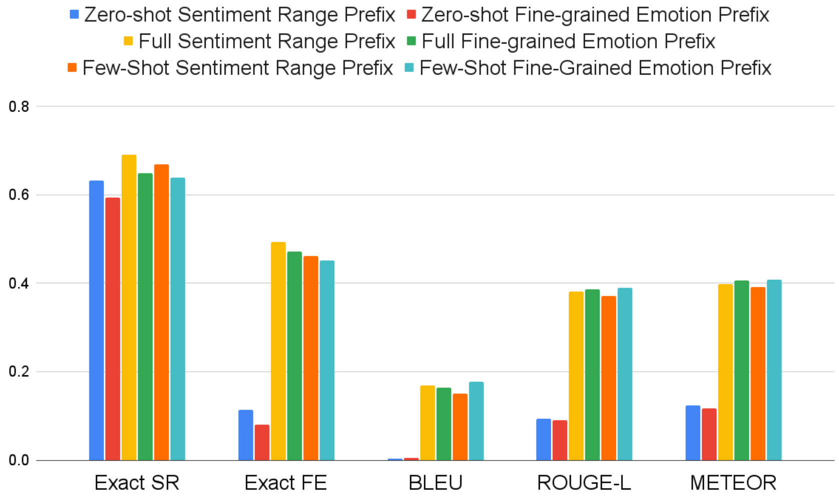


Figure 8: Evaluation Results of Mix Dataset on GPT2

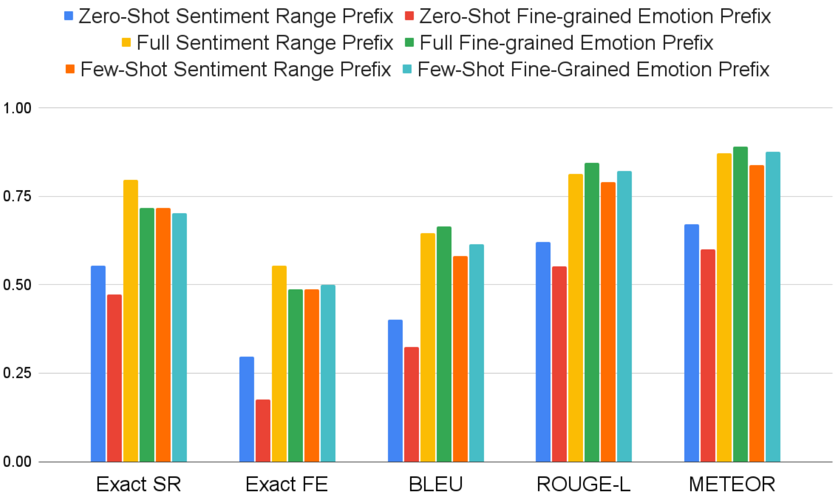


Figure 9: Evaluation Results of Google Dataset on T5

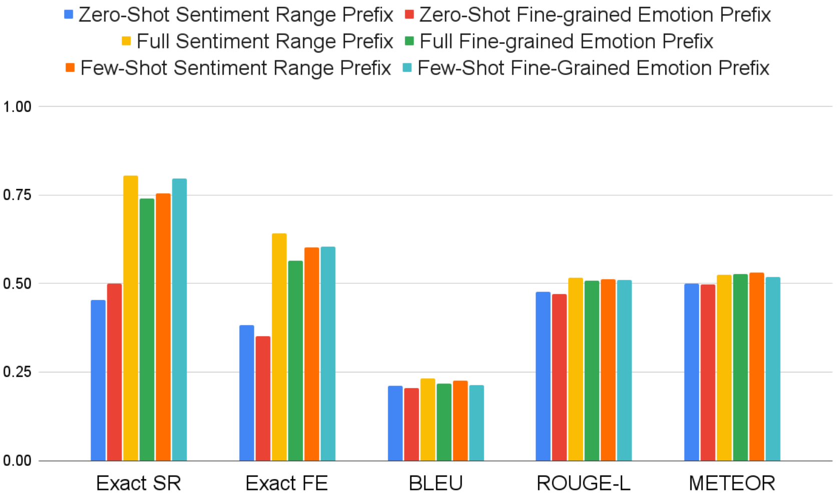


Figure 10: Evaluation Results of Quora Dataset on T5

Emotions in Spoken Language - Do we need acoustics?

Nadine Probol
University of Applied Sciences
Darmstadt
nadine.probol@h-da.de

Margot Mieskes
University of Applied Sciences
Darmstadt
margot.mieskes@h-da.de

Abstract

Work on emotion detection is often focused on textual data from i.e. Social Media. If multimodal data (i.e. speech) is analysed, the focus again is often placed on the transcription. This paper takes a closer look at how crucial acoustic information actually is for the recognition of emotions from multimodal data. To this end we use the IEMOCAP data, which is one of the larger data sets that provides transcriptions, audio recordings and manual emotion categorization. We build models for emotion classification using text-only, acoustics-only and combining both modalities in order to examine the influence of the various modalities on the final categorization. Our results indicate that using text-only models outperform acoustics-only models. But combining text-only and acoustic-only models improves the results. Additionally, we perform a qualitative analysis and find that a range of misclassifications are due to factors not related to the model, but to the data such as, recording quality, a challenging classification task and misclassifications that are unsurprising for humans.

1 Introduction

The correct detection of emotions in spoken language is a task which has been examined a lot in recent years. But the majority of research so far, has treated text and speech separately and there is little research on using both audio and textual data for emotion recognition, which would stand to reason as emotions are expressed not only in *what* is being said, but also *how*. One exception is Ho et al. (2020) who present a model using Multi-Level Multi-Head Fusion Attention mechanism and recurrent neural network (RNN) for the detection of the audio input in combination with information gained from text.

Building on previous work, we show extensive experiments on various types of models trained on text as well as speech. We examine, whether the combination of emotion recognition from textual data as well as audio data improves the results to either singular approach. We also explore different feature sets and models to find the best combination. This allows us to conclude, that, while using only textual data yields reasonable results for English, we achieve best results when combining it with additional acoustic data.

In a qualitative analysis we take a look at the most striking confusions between classes and give possible reasons for them. Additionally, we give an idea of how to counter these problems in future experiments.

In Section 2 we describe previously conducted work on emotion recognition. For this, in Section 2.1 we describe some of the more often used data sets available for this task. In Section 2.2 we describe some related work in more detail. The experimental setup is described in Section 3. It additionally focuses on the feature extraction for the audio data (Section 3.1), the preprocessing of the textual data (Section 3.2, as well as the model used for the transcriptions (Section 3.3 and the method of combination for the different models (Section 3.5. Section 4 shows the best results of the combined models and three different data set variations. In Section 5 we conduct a qualitative analysis of the results. Finally, we conclude in Section 6.

Our major contributions therefore are:

- A comparison of results on text only, speech only and the combination thereof.
- An analysis of various combinations of acoustic and textual features and models for the classification of emotions from speech data.
- A qualitative analysis of the problematic cases and an investigation into the sources of these cases.

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 71–84
July 14, 2023 ©2023 Association for Computational Linguistics

2 Previous Work

The detection of emotions in spoken language is a complex problem. Emotions in humans are expressed both in the choice of words used, but also in the way these words are expressed acoustically. The related work on emotion detection based on transcribed speech, acoustical information and a combination thereof, described below is focused on relevant experiments related to our work and does not represent an exhaustive review of the topic.

2.1 Emotion Data sets

There are various data sets for multimodal emotion detection, such as IEMOCAP[1], EmoDB,[2] DES,[3] SAVEE,[4] or CASIA.[5] These data sets cover various languages such as English (IEMOCAP), German (EmoDB), Chinese (CASIA, NNIME[6]) or Danish (DES).

Out of these, the IEMOCAP data set is one of the larger ones. Busso et al. (2008) introduced the IEMOCAP (Interactive Emotional dyadic Motion Capture Database) data set. It contains 9,924 example utterances and corresponding labels, spoken by ten different actors (five male, five female), resulting in a quite balanced set with a slightly higher amount of female data (51.37%, Figure 5 in Appendix C). The most common length for an utterance is one word (701 utterances), followed by six words (668) and five words (653) and the examples are further divided into five sessions, containing 80 improvised and 70 scripted dialogues (Figure 6 in Appendix C). Overall, the data set contains slightly more improvised examples (52.95%) than scripted ones (47.05%), but all recordings are of actors, so there are no naturally occurring dialogues. In the proposed models, each labelled example will be used without the context of the given dialogue and corresponding session. Each utterance has been annotated by three different annotators (out of six) and labelled as one of ten categories (Anger, Happiness, Sadness, Neutral state, Frustrated, Excited, Fear, Surprise, Disgust, Other), which differ greatly in size (Figure 7 in Appendix C).

The data set includes the full transcripts, hand-annotated emotion labels and the audio recordings.

The IEMOCAP data set has been widely used for example by Mirsamadi et al. (2017), Mao et al. (2019), Dangol et al. (2020), or Lieskovska et al. (2022).

2.2 Methods for emotion detection

Emotion recognition of speech is either done text-only or acoustics-only, combined models are rarer. In the following, we present work from all three areas as examples.

Acoustic data

Based on their work, Lieskovská et al. (2021) concluded, that the usage of deep convolution architectures, which are based on spectrum information only is increasing. The authors considered these architectures as well as recurrent networks as a strong base for emotion recognition systems for speech. They state that, even though many used attention mechanisms to improve the performance of their model, the magnitude of improvement is unclear, which makes this approach dispensable.

An interesting experiment was conducted by Mirsamadi et al. (2017), who compared a neural network with an SVM on the emotions happy, sad, neutral, and angry of the IEMOCAP data set. The authors proposed a deep neural network with two hidden layers followed by an LSTM layer. They used emotion LLDs with RNN-weighted pooling with a logistic regression attention model. In their work, this approach performed best, by focusing on emotional parts of utterances. They also trained an SVM on mean, standard deviation, minimum, maximum, range, skewness and kurtosis. Overall, the authors concluded, that the SVM approach needed a higher amount of statistical functions to reach its best performance, whereas the DNN was less sensitive to number and diversity of the used statistical functions.

Textual data

Also, there are interesting experiments for the research on textual data. For example, Mohammadi et al. (2019) compared the results of SVMs, neural networks and a combination of the two. The authors used the pre-trained ELMo word embedder by Peters et al. (2018). Following the input layer were two layers of 25 bidirectional GRUs and an attention layer. These steps were done three times in parallel and their output was then concate-

---

[1] `https://sail.usc.edu/iemocap/`

[2] `http://emodb.bilderbar.info/docu/`

[3] `http://universal.elra.info/product_info.php?cPath=37_39&products_id=78`

[4] `https://www.tensorflow.org/datasets/catalog/savee`

[5] `http://www.chineseldc.org/resource_info.php?rid=76`

[6] `https://nnime.ee.nthu.edu.tw/`

nated. Additionally, the authors used an SVM with polynomial kernel with a degree of 4 and set C to 2.5. Using a neural network for feature extraction and an SVM for classification gave the best results. However, as it was for task 3 at SemEval 2019[7], they only classified three classes: angry, happy, and sad.

Chakravartula and Indurthi (2019) present a model with a stacked BiLSTM architecture for the SemEval 2019 task 3, which is based on written dialogues. The authors used three different embedding layers: The first embedding layer converts each word into its corresponding 300 dimensional GloveEmb word vector, the second takes the POS tags and converts each of them into a constant one-hot vector and the third embedding layer converts each word into a vector based on the values in the DepecheMood affective lexicon (Staiano and Guerini, 2014). They achieved the best results by combining the first and third embedding layer with two BiLSTM layers, however, combining the first embedding layer with two BiLSTM layers and attention reached comparable results.

Acoustic and textual data combined

There is little research on combining acoustic and textual data for emotion detection. The following are the most important for the work at hand.

Yoon et al. (2018) built two encoders: The Audio Recurrent Encoder (ARE) and the Text Recurrent Encoder (TRE), which work in parallel. For the audio encoder, they use MFCCs and prosodic features, which they extract via the *openSMILE* toolkit. By using the *NLTK* toolkit, the authors tokenized and indexed the transcripts into a sequence of tokens. Both, the ARE and TRE use RNNs to each predict an emotion class. For a final prediction of both models together, the authors use a softmax function to concatenate the vectors of the predictions of the audio RNN and text RNN. Later, the authors improved their model (Yoon et al., 2019) by using a bidirectional encoder (BRE) for both the textual and audio data instead of unidirectional. The final hidden representation of the audio-BRE is then used as a context vector to bring attention to the text-based vector. The authors then apply second-hop attention with this information to the audio sequence, which they call MHA-2. Both times the authors used a variation of the IEMOCAP data set (using only the categories angry, happy, sad and neutral) to test their model.

Another approach by Ho et al. (2020) proposes a multi-level multi-head fusion attention model (MMFA). For the audio data, the authors extract MFCC features via the *openSMILE* toolkit as well. The audio frame is set to 100ms at a rate of 50ms and a Hamming window is applied, so the temporal length of the audio is ten times longer than the length of the utterance pre re-sampling. The authors state, that the attention mechanism in MMFA combines the contextual information of audio and text. The final model can be divided into two parts: first-level attention and second-level attention. The first part computes a representation for each audio and text RNN-feature at different positions of a single sequence. The second attention is a modified multi-head attention, in order to fuse the attention features from both audio and text. It does not compute this attention just once, but multiple times in parallel. The authors use two different versions of the IEMOCAP data set to test their models: one with four classes ("Neutral state", "Anger", "Sadness", and "Happiness/Excited") and one with eight classes (the original classes of the data set minus "Disgust" and "Other"). Also, they look at two scenarios: improvised (using only the improvised examples of the data) and mixed (using all data). When concentrating on the variation with four classes, the model achieves an accuracy of 0.73 on the mixed data and 0.77 on the improvised data. For the version with eight classes, the model reaches an accuracy of 0.57 for the mixed data and 0.61 on the improvised data.

3 Experimental Setup

For our experiments[8] we use the IEMOCAP data set by Busso et al. (2008). As, out of the above presented data sets (Section 2.1), the IEMOCAP data set is the largest and provides both textual and audio data in English. The EmoDB, CASI, NNIME and DES also provide both textual and audio data, but they are either very small (i.e. EmoDB only has 500 sentences) or in Chinese (CASIA, NNIME), which is a tonal language, as opposed to English, which is a non-tonal language and therefore, expresses emotions differently.[9]

---

[7] As only Task 3 of the SemEval 2019 workshop focuses on emotion detection (EmoContext), we only looked at the papers for this Task.

[8] All experiments are conducted on Windows 10 with Python 3.8.10. The additionally used libraries are listed with their corresponding version in Table 3 in Appendix A.

[9] Additionally, none of the authors are proficient in Chinese, which makes the qualitative analysis impossible.

The authors performed the emotion category annotation and report a Fleiss' Kappa of 0.27 on the entire annotation. Our re-calculation of Fleiss' Kappa resulted in an observed agreement of 0.23, an expected agreement of 0.27 and an overall agreement of -0.06.[10] However, Fleiss' Kappa is not applicable to data sets with empty annotations, which is the case in this data set, as there are six annotators in total, but only three given annotations per example. Thus, we use Krippendorff's Alpha to verify their results. This leads to an observed disagreement of 0.77, an expected disagreement of 0.73 and an agreement of -0.06, which confirms the low agreement between the annotators.

As the original data set does not provide a gold standard, we use a majority vote between the three annotations. If all annotators decide on different labels, we use the label of the first annotator. Another option would have been to randomly take one annotation, but by always deciding on the first annotator in these cases, it might lead to a slightly higher consistency. Table 1 shows the spreading of the distribution of classes in the different variations.

Emotion	Original	Variation 1	Variation 2
Anger	x	xX	xZ
Happiness	x	xY	xY
Sadness	x	x	xZ
Neutral state	x	–	x
Frustration	x	xX	xZ
Excited	x	xY	xY
Fear	x	–	–
Surprise	x	–	–
Disgust	x	–	–
Other	x	–	–

Table 1: Different variations of the IEMOCAP data set. Added capital letters show a combination of two or more classes. For example, In Variation 1, classes "Anger" and "Frustration" are combined (shown with a X), as well as the classes "Happiness" and "Excited" (shown with a Y).

The classes "Fear" and "Neutral state" are excluded in Variation 1. "Fear" is an extremely small class, though not as small as "Disgust". The class "Neutral state" is also excluded. Although it is the largest class, it shows no explicit emotion.

Variation 2 combines the classes "Anger", "Frustration" and "Sadness", as they are all negative emotions. It contains the combined class of the happy emotions of "Happiness" and "Excited" as well as the class "Neutral state". This class can be seen as a sentiment classifications (positive, negative and neutral).

Previous work show that some authors group different classes together based on similar acoustic signals. For example, Nwe et al. (2003) combined the classes "Anger", "Surprise" and "Joy" in one cluster and the classes "Fear", "Sadness" and "Disgust" in another one. This approach is based on findings by Williams and Stevens (1981) (cited by Nwe et al. (2003)), who found that emotions such as "Anger" and Fear" but also "Joy" arouse the sympathetic nervous system, while emotions such as "Sadness" arouse the parasympathetic nervous system. An aroused sympathetic nervous system leads to an accelerated heart rate and higher blood pressure, a dry mouth and even occasional muscle tremors. This shows in a loud, fast and enunciated speech with strong high frequency energy. In contrast, an aroused parasympathetic nervous system leads to lower blood pressure and heart rate, as well as an increased salivation. Speech produced under these circumstances is slow with little high frequency energy. Even though Nwe et al. (2003) improved their results by grouping the emotion classes accordingly (accuracy reaches up to 90%), it is questionable whether it is useful to the actual use case to subsume such different classes.

In our experiments, we combine emotions which are similar, such as "Happiness" and "Excited", or show the same sentiment ("Anger", "Frustration" and "Sadness" are all negative).

3.1 Feature extraction for audio data

To extract features from the audio files, we use two different approaches.

First, we use the *openSMILE* toolkit by Eyben et al. (2010), which is accessible via a Python API.[11] This API has six different extractable feature sets and we chose "ComParE_2016", which was first introduced for the *Interspeech 2016* (Schuller et al., 2016), as this provides the largest amount of different features extracted.

We extract features as two-dimensional tables along the time-axis. For each feature, we calculate the maximum, minimum, mean and standard deviation, which results in 100 different features. This is comparable to Mirsamadi et al. (2017), who also used the mean, standard deviation, minimum, maximum, and other features to train their models.

---

[10] Calculated using `https://dkpro.github.io/dkpro-statistics/`

[11] `https://github.com/audeering/opensmile-python/`

The second approach is based on the *librosa* library (McFee et al., 2015), which extracts Mel Frequency Cepstral Coefficients (MFCCs). The audio signal is split evenly into $slices$ of 10 ms and for each slice we extract 13 MFCCs. For longer recordings, this *can* result in files with more than $1,000$ slices. As this is the case for only 12 examples: Six from class "Sadness", three from class "Frustration", two from class "Excited" and one example from class "Anger", but considerably affects computing time, we cut off all MFCCs after $1,000$slices.

3.2 Preprocessing text data

We tokenize the transcription using the *NLTK* tokenizer.[12] Then, we used part of the code from the *tensorflow* tutorial for word2vec[13] to change those tokens to numbers, so the model can process them, as well as add them to a vocabulary dictionary. The maximum word vector length is 200. Longer examples are cut off after the 200th word and shorter examples are filled up with zeros. Zeros have no word associated within the vocabulary dictionary as they are used as padding. Additionally, we change the class labels to numbers and then change those numbers to categorical tensors.

3.3 Model for Transcripts

For the transcriptions, we trained a neural network with bidirectional LSTM and GRU layers and set the vocabulary size to 15,000. Bidirectional LSTM layers are used in many other experiments, such as the ones by Chakravartula and Indurthi (2019) and seem to achieve good results. There are not as many experiments using GRU layers, however, the aforementioned experiment by Mohammadi et al. (2019) achieved good results using GRU layers. Chatterjee et al. (2019) described that most participants in the SemEval 2019 Task 3 were using LSTM and BiLSTM models, though GRU and CNN models were also used by a few teams. The complete model, including hyperparameters, we use for the transcripts is visualized in Appendix B (Figure 4).

3.4 Model for Acoustics

In general, we trained SVMs and neural networks on the acoustic data.

First, we trained an SVM on all standard derivations (named SVM SD in the following) and one on all 100 available features extracted through *openSMILE* (SVM 100).[14]

Second, we use the MFCCs extracted through the *librosa* library to train various neural networks, as recurrent networks seem to be a good basis for emotion recognition systems (Lieskovská et al., 2021). This is comparable to Wang et al. (2015), however, we focused on the statistical parameters and used a different data set. Based on preliminary experiments done by ourselves, as well as on experiments of Mirsamadi et al. (2017) and Dangol et al. (2020), we test different architectures with LSTM and GRU layers and different combinations to improve the results. Additionally, we test the different usages of ReLU or SELU activation functions. By using SELU activation we avoid running into the so called dying ReLU problem, which was described for example by Agarap (2018) or Lu et al. (2019). The dying ReLu problem refers to neurons becoming inactive and therefore only have output 0 for any input. SELU activation function induces self-normalization, which is faster than an external normalization, and therefore leads to a faster convergence of the network. Contrary to ReLU, it can go below 0, avoiding dying neurons.

The exact models and their names are listed in Table 5 in Appendix D.

One model (Audio NN 4 as named in Table 5 in the Appendix) is more complex than the other models as it consists of a convolutional 2D layer with ReLU activation and 32 hidden units, followed by a max pooling 2D layer and a batch normalization layer. These layers are then following once again. After, a dense layer with 64 hidden units, SELU activation and L2 regularization follows and again a max pooling 2D and batch normalization layer. Then comes a flatten layer and another dense layer with 64 hidden units and ReLU activation. After a dropout of 0.3 follows the same output layer as in the former models.

For the *openSMILE* features, we used similar architectures, as for the *librosa* features. Differences in the architectures are due to the different shape of the data, which for example results in the usage of a standard input layer as an input for the NNs trained on *openSMILE* features, whereas this is not possible for *librosa* features.

---

[12] `https://www.nltk.org/_modules/nltk/tokenize.html`

[13] `https://www.tensorflow.org/tutorials/text/word2vec`

[14] Additionally trained SVMs are listed in Table 6 in Appendix D.

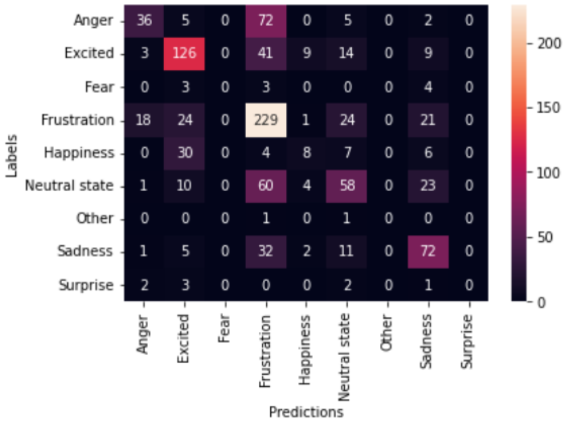


Figure 1: Confusion Matrix of the combination of the Text NN, SVM SD and SVM 100 on the original data set

The exact specifications of each model are listed in Table 4 in Appendix D and link them to the names we use for the single models.

3.5 Method of combination

In order to combine results from the acoustical classification model and the textual classification model, we use two different approaches:

The first approach combines every model with every other model with a weighting of 50% to 50%. Yoon et al. (2018) and Yoon et al. (2019) combine their models with this weighting, however, they always combine their models trained on textual features with one trained on acoustic features. The combinations in our experiments can also include only models based on acoustic features, as the ablation process only focuses on the best results and not the feature type.

The second approach of combining models is very similar to the first one. Here, we combine three models with a weighing of 33% to 33% to 33%.This also excludes the combination of two or three same models, so there is no single model prediction or a weighing of 33% to 67%. With this method of combination, 21,168 different combinations are possible which we all tested and compared their results in order to find the best combination.

4 Results

To classify the results, we compare them to a majority baseline (Table 2).

4.1 Original data set

Table 2 shows the best results for the original data set classifying all emotions available. The best single model is the Text NN. By combining the model with the SVM 100, the results improve, however, combining those two models with the SVM trained on the standard derivations of all features gives the best results on the original data set (Section 3.4).

This leads to an accuracy of 0.53, which is much lower than Ho et al. (2020), however, they dropped the smallest classes "Disgust" and "Other", which do have a negative impact on our model performance.

A look at the confusion matrix in Figure 1 shows, there are three classes, that do not get predicted at all. The combined models do not predict any examples as "Fear", "Other" or "Surprise". This aligns with class size, as these three classes are much smaller than the other classes (except for "Disgust"). This also applies to class "Disgust", which is not visible in the confusion matrix, as there is no example of it in the test split. There are only two examples of class "Disgust" in the data set, which were both automatically sorted into the train split.

Only 33.3% of all examples predicted as "Happiness" are correctly identified as such. Most confusions happen on class "Frustration". 48.2% of the examples predicted as "Frustration" do not belong to this class. 33.8% of these misclassifications are examples of class "Anger", which also means, there are more examples of class "Anger" predicted as "Frustration" (60.0%), than correctly identified. These confusions are also the most likely ones with these combined models (15.5% of all misclassifications). The second highest amount of misclassifications happen with examples of class "Happiness" as "Frustration" (12.9% of all misclassifications). There are also slightly more confusions of examples of class "Happiness" as "Frustration" (38.5%), than correctly identified (37.2%).

A more detailed analysis based on the single classes of the data set can be found in Section E in the Appendix.

4.2 Data set variation 1

Table 2 shows the best results for data set variation 1 which combined "Anger" with "Frustration", "Happiness" with "Excited" and kept "Sadness" and "Surprise" as separate categories. The results on data set variation 1, as for the original data set,

Variation	Models	Baseline	Macro Precision	Macro Recall	Accuracy	Macro F1
Original	Text NN	0.32	0.30	0.28	0.48	0.28
	Text NN + SVM 100		0.32	0.29	0.50	0.30
	Text NN + SVM SD + SVM 100		**0.34**	**0.31**	**0.53**	**0.31**
Variation 1	Text NN	0.49	**0.74**	0.52	0.70	0.54
	Text NN + SVM 100		0.53	0.52	0.70	0.52
	Text NN + Audio NN 4 + SVM 100		0.58	**0.54**	**0.76**	**0.55**
Variation 2	Text NN	0.55	0.62	**0.59**	0.69	0.59
	Text NN + SVM 100		0.65	**0.59**	0.71	0.59
	Text NN + SVM SD + SVM 100		**0.72**	**0.59**	**0.72**	**0.60**

Table 2: Results on different data set variations and their baselines. The best results are marked in bold.

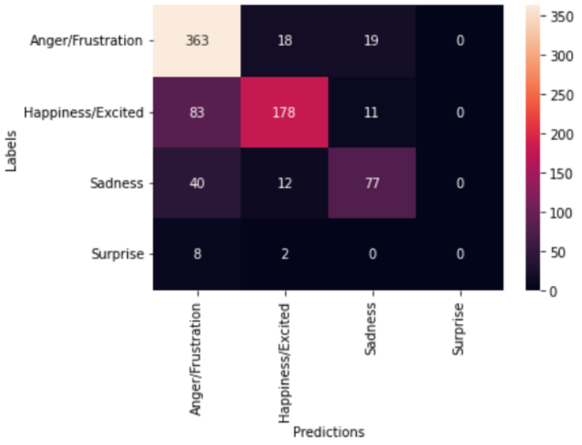


Figure 2: Confusion Matrix of the combination of the Text NN, Audio NN 4 and SVM 100 on data set variation 1

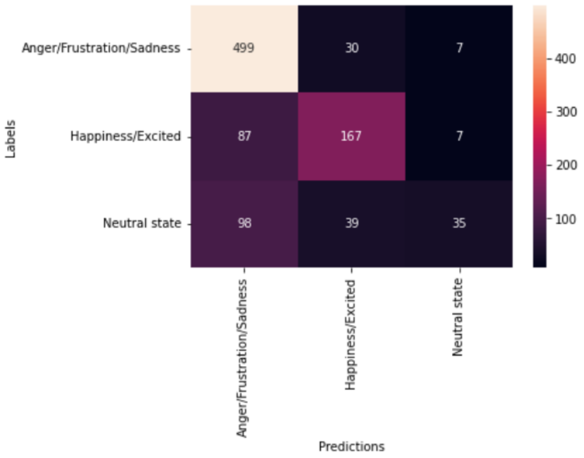


Figure 3: Confusion Matrix of the combination of the Text NN, SVM SD and SVM 100 on data set variation 2

show the best single model as the Text NN. However, the best combination of two models (Text NN and SVM 100) reach lower results. The combination of three models reaches also the best results out of all our experiments by combining the Text NN with the Audio NN 4 and SVM 100.

This combination achieves an accuracy of 0.76, which is higher than the accuracy Ho et al. (2020) reach on their four classes model (0.73). However, the classes are slightly different, as they detected neutral, angry, sad, and happy/excited. Further information on the performance of the single classes can be found in the Appendix E.

Overall, most correct classifications happen on class "Anger/Frustration" (58.7% and 50.9% of all correct predictions).

On the combination of the Text NN with Audio NN 4 and SVM 31 (Figure 2), the class "Happiness/Excited" is most often confused with other classes. 30.5% of the examples of class "Happiness/Excited" are predicted as "Anger/Frustration", which make up 43.0% of all misclassifications. The second highest amount of misclassifications happen with examples of class "Sadness" as "Anger/Frustration" (20.7%). The examples of class "Surprise" are mostly predicted as "Anger/Frustration", even though there are only ten examples on the test split.

4.3 Data set variation 2

The best results on data set variation 2 (which combines "Anger", "Sadness" and "Frustration", "Happiness" and "Excited" and keeps "Neutral state" as a separate category) are also shown in Table 2. The best single model is, like on the former data set variations, the Text NN, whose results improve by adding SVM 100. Again, the combination of Text NN, with SVM SD and SVM 100 (similar to the model combination for the original data set Section 4.1) achieves the best results (Accuracy of 0.72). While the recall remains the same for the single, two and three combined models, the additional information from the other models does specifically improve precision. More details on the single classes can be found in Appendix E.

The confusions matrix (Figure 3) shows 57.0% of the examples of class "Neutral state" are predicted as "Anger/Frustration/Sadness". This also represents the highest amount of confusions between two single classes (36.6% of all misclassifications). The second highest amount of misclassifications happen with examples of class "Happiness/Excited" as "Anger/Frustration/Sadness" (32.5% of all misclassifications).

Most correct predictions are on class "Anger/Frustration/Sadness" (71.2% of all correct predictions).

5 Qualitative Analysis

In our qualitative analysis, we take a closer look at those examples that have been misclassified. In general, most of the examples which are miscategorized are very short.

Also, we observed that there are examples that are not correctly identified by neither of the three models and data set variations. A closer look, revealed they are spoken by a female, however, in some instances a male speaker can be heard clearly in the background. While the female seems to be very happy in those, the background noise of the male could have negatively impacted the recognition process to "Anger" as he sounds quite angry.

Also, there was an example, which was predicted as "Excited", even though it was meant to be "Sad" as indicated by the manual annotation. This was interesting, as the person speaking sounded quite desperate, which shows more energy than the average sad person, which normally exhibits a rather low energy level. Therefore, the wrong classification as "Excited" makes sense and should be addressed in further experiments.

Additionally, we looked at some examples, which are wrongly classified in at least two of the three models.

After a proper examination, there are two types of confusions which stand out in particular: **Negative emotions which are confused as happy ones**. The aforementioned confusion of an example of a desperate person was not happening in just one example. Additionally, there was an example of a frustrated male which was quite energetic. However, in the background a woman was starting to speak in a higher voice, but the recording was cut shortly thereafter, making it impossible to understand what the woman ways saying. Her higher voice might have influenced the levels of the acoustic statistics we trained our model on.

Happy emotions which are confused as negative ones. Those happy examples tend to present low energy, for example, a person states "I love you a great deal!". This is a very happy statement, however, as the person sounds very close to tears, the model predicts it as "Frustration". Another person is very calmly speaking, however, due to the calmness, the person appears to the model to be sad. Both misclassifications do make sense as, as described by Nwe et al. (2003), speech representing sadness is slower and characterized by less high frequency energy.

Another noteworthy aspect is the length of the examples. While Seehapoch and Wongthanavasu (2013) already concluded that it is more difficult to correctly recognise the emotion if the speech is too long, our findings suggest the same is true for examples which are too short. There are several very short examples on the data set (approx. 1s), which are incorrectly classified. While some of them do express emotions, it can be hard even for humans to correctly determine them without context.

Many of the wrongly predicted examples also could be categorized in various classes, which explains the low inter-annotator-agreement.

6 Conclusion

In our work, we examined the benefit of combining information from text with information from speech in order to categorize emotions in spoken language. To this end, we used the IEMOCAP data set, in its original classification, but also in different combinations, to train various machine learning models (SVM and neural networks) to classify emotions. We combined the trained models in various ways to find the best combination of models for the classification.

Our results indicate that a combination of text-based features together with acoustic features provide the best results, as all combinations contained both models based on textual information, as well as acoustic information.

A detailed look at the results reveals that neural networks trained on textual data perform best, when only one modality (text) is used. However, when multimodal data (text and speech) is available, making use of all modalities improves the classification and that textual data is crucial for a successful classification. Our results also indicate, that using only acoustical data gives results that are even below those for text-only based classification. So yes, we need acoustics, but acoustics alone do not provide enough information for a successful classification.

Also, there is no clear indication, if *openSMILe* or *librosa* features provide better results. But, we observed that the SVMs trained on various acoustic features perform much better than the neural networks trained on the same features. This is probably due to the relatively small data set size.

In general, it seems that negative classes are easier to classify than positive emotions. As our results are only based on one data set, it would be worthwhile to explore this in more detail.

Additionally, we see a systematic problem with the correct classification of examples with unusual energy levels for the respective emotion. Therefore, it might be an interesting approach for examples of the class "Excited" to be divided into positive and negative excitedness. However, this would also need larger data sets, as the class "Excited" is already quite small in the IEMOCAP data set, which is already one of the larger ones available. To avoid the confusion of happy expression with unusually low energy levels, it might be important to add more examples like these to the training data. A first approach to do so could be achieved by simply oversampling happy examples with statistically low energy levels.

In general, some of the examples of the data set are hard to understand even for humans. This is due to poor audio quality as well as background noise. Sometimes we needed to look at the label in order to decipher which talking person was labelled, which also accounts for the low inter-annotator agreement we observed. It might be a useful approach to detect these examples and drop them before training a model, as this does only apply to a small part of the data set.

Broadly speaking, a combination of more audio features improves the results in comparison to single audio features. However, it should be further investigated, whether focusing on specific features might improve the results even more. A focus hereby should lay on the standard deviation of the features, as they reach the best results after the combination of all acoustic features.

Another interesting evaluation would be, to see if the model predicts one of the manually assigned labels, under the assumption that *all* human labels are correct. This would reduce the need for a single-value gold standard and would give us an insight into those cases, where the model clearly misclassifies the examples.

Overall, the findings suggest, that future research should explore, whether the combination of audio and textual data on one model improves the results even more, than the combination of three separate models. Additionally, it should be further investigated whether there is an influence of the gender of the speaker on the correctness of the predictions as we did not look at it at all.

Limitations

While working with the data of the IEMOCAP data set, we realised that some of the examples are of very low quality which can negatively affect the performance of the models. This, however, does only apply to a small part of the data set. Additionally, it is not possible to correctly asses the influence of the gender on the performance. The same applies to a possible influence of the way, the data is generated: scripted or improvised.

In some models we use SELU activation function, which is still not widely used, therefore, it is possible that there are problems that are not that well known.

In general, there are limitations based on the data set. It only contains scripted and improvised recordings, by actors, which might not be representative of naturally occurring emotions. Also, as the data set is recorded in English, any generalizations outside this language are not possible.

Ethics Statement

As we work with data that has been published before the ACL Ethics Charter was implemented, we cannot guarantee that the way the data was collected and handled meets current Ethics Standards. As far as we can tell, it is still a suitable data set for this type of research. There are however the limitations mentioned above. Also, there is no information given about the age or ethnicity of the speakers.

References

Abien Fred Agarap. 2018. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*.

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan. 2008. IEMOCAP: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42(4):335.

Nikhil Chakravartula and Vijayasaradhi Indurthi. 2019. EMOMINER at SemEval-2019 task 3: A stacked

BiLSTM architecture for contextual emotion detection in text. In *Proceedings of the 13th International Workshop on Semantic Evaluation (SEMEVAL 2019),* Minneapolis, Minnesota, USA, Jun 6th – 7th 2019, pages 205–209. Association for Computational Linguistics.

Ankush Chatterjee, Kedhar Nath Narahari, Meghana Joshi, and Puneet Agrawal. 2019. Semeval-2019 task 3: Emocontext contextual emotion detection in text. In *Proceedings of the 13th International Workshop on Semantic Evaluation (SEMEVAL 2019),* Minneapolis, Minnesota, USA, Jun 6th – 7th 2019, pages 39–48. Association for Computational Linguistics.

Ranjana Dangol, Abeer Alsadoon, PWC Prasad, Indra Seher, and Omar Hisham Alsadoon. 2020. Speech Emotion Recognition Using Convolutional Neural Network and Long-Short Term Memory. *Multimedia Tools and Applications*, 79(43):32917–32934.

Florian Eyben, Martin Wöllmer, and Björn Schuller. 2010. Opensmile: the munich versatile and fast open-source audio feature extractor. In *Proceedings of the 18th ACM international conference on Multimedia (MM 2010),* Firenze, Italy, Oct 25th – 29th 2010, pages 1459–1462.

Ngoc-Huynh Ho, Hyung-Jeong Yang, Soo-Hyung Kim, and Gueesang Lee. 2020. Multimodal approach of speech emotion recognition using multi-level multi-head fusion attention-based recurrent neural network. *IEEE Access*, 8:61672–61686.

Eva Lieskovska, Maros Jakubec, and Roman Jarina. 2022. RNN with Improved Temporal Modeling for Speech Emotion Recognition. In *2022 32nd International Conference Radioelektronika (RADIOELEKTRONIKA)*, pages 1–5. IEEE.

Eva Lieskovská, Maroš Jakubec, Roman Jarina, and Michal Chmulík. 2021. A review on speech emotion recognition using deep learning and attention mechanism. *Electronics*, 10(10):1163.

Lu Lu, Yeonjong Shin, Yanhui Su, and George Em Karniadakis. 2019. Dying relu and initialization: Theory and numerical examples. *arXiv preprint arXiv:1903.06733*.

Shuiyang Mao, Dehua Tao, Guangyan Zhang, P. C. Ching, and Tan Lee. 2019. Revisiting Hidden Markov Models for Speech Emotion Recognition. In *Proceedings of the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2019),* Brighton, UK, May 12th – 17th 2019, pages 6715–6719.

Brian McFee, Colin Raffel, Dawen Liang, Daniel P Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. 2015. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, volume 8, pages 18–25. Citeseer.

Seyedmahdad Mirsamadi, Emad Barsoum, and Cha Zhang. 2017. Automatic speech emotion recognition using recurrent neural networks with local attention. In *Proceedings of the 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2017),* New Orleans, LA, USA, Mar 5th – 9th 2017, pages 2227–2231. IEEE.

Elham Mohammadi, Hessam Amini, and Leila Kosseim. 2019. CLaC lab at SemEval-2019 task 3: Contextual emotion detection using a combination of neural networks and SVM. In *Proceedings of the 13th International Workshop on Semantic Evaluation (SEMEVAL 2019),* Minneapolis, Minnesota, USA, Jun 6th – 7th 2019, pages 153–158.

Tin Lay Nwe, Say Wei Foo, and Liyanage C De Silva. 2003. Speech emotion recognition using hidden Markov models. *Speech communication*, 41(4):603–623.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, and Kenton Lee. 2018. Luke" Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1.

Björn Schuller, Stefan Steidl, Anton Batliner, Julia Hirschberg, Judee K. Burgoon, Alice Baird, Aaron Elkins, Yue Zhang, Eduardo Coutinho, Keelan Evanini, et al. 2016. The Interspeech 2016 Computational Paralinguistics Challenge: Deception, Sincerity & Native Language. In *Proceedings of the 17th Annual Conference of the International Speech Communication Association (INTERSPEECH 2016), Vols 1-5,* San Francisco, USA, Sep 8th – 12th 2016, pages 2001–2005.

Thapanee Seehapoch and Sartra Wongthanavasu. 2013. Speech emotion recognition using support vector machines. In *Proceedings of the 5th international conference on Knowledge and smart technology (KST 2013),* Chonburi Province, Thailand, Jan 31th – Feb 1st 2013, pages 86–91. IEEE.

Jacopo Staiano and Marco Guerini. 2014. Depechemood: a lexicon for emotion analysis from crowd-annotated news. *arXiv preprint arXiv:1405.1605*.

Kunxia Wang, Ning An, Bing Nan Li, Yanyong Zhang, and Lian Li. 2015. Speech Emotion Recognition Using Fourier Parameters. *IEEE Transactions on Affective Computing*, 6(1):69–75.

Carl E. Williams and Kenneth N. Stevens. 1981. Vocal correlates of emotional states. *Speech evaluation in psychiatry*, pages 221–240.

Seunghyun Yoon, Seokhyun Byun, Subhadeep Dey, and Kyomin Jung. 2019. Speech emotion recognition using multi-hop attention mechanism. In *Proceedings of the 2019 IEEE International Conference on*

Acoustics, Speech and Signal Processing (ICASSP 2019), Brighton, UK, May 12th – 17th 2019, pages 2822–2826. IEEE.

Seunghyun Yoon, Seokhyun Byun, and Kyomin Jung. 2018. Multimodal speech emotion recognition using audio and text. In *Proceedings of the 2018 IEEE Spoken Language Technology Workshop (SLT 2018)*, Athens, Greece, Dec 18th – 21th 2018, pages 112–118. IEEE.

A Computational specifications

Library	Version	Library	Version
Python	3.8.10	NLTK	3.6.7
Tensorflow	2.7.0	Seaborn	0.11.2
Keras	2.7.0	Scikit-learn	1.0.2
Tensorflow Addons	0.15.0	Matplotlib	3.5.1
Numpy	1.22.0	Librosa	0.8.1
Pandas	1.3.5		

Table 3: Libraries and their versions used in the experiments

B Method for Transcription

The model (Figure 4) starts with a vectorization layer, which is not part of the *tensorflow* or *keras* library, though it is described on their explanation page for word embeddings.[15]

C Data set

The following figures show the gender distribution of the data (Figure 5), the distribution between improvised and scripted conversations (Figure 6) and the distribution of emotion classes (Figure 7).

D Methods for audio data

The first 25 SVMs use the minimum, maximum, average and standard deviation of the listed feature (Table 6). Mirsamadi et al. (2017) trained their SVMs on these statistical features, however, they added range, skewness and kurtosis as well, which we leave out in our experiments. The classifier trained on all_frequ_ban_amp is trained on maximum, minimum, average and standard derivation of the features F1 Frequency to F3 Amplitude Log Rel F0. The classifiers all_max, all_min, all_avg and all_std are respectively trained on the maximum, minimum, average and standard deviation of the first 25 features. Classifier 31 is trained on all available features attained by using *openSMILE*.

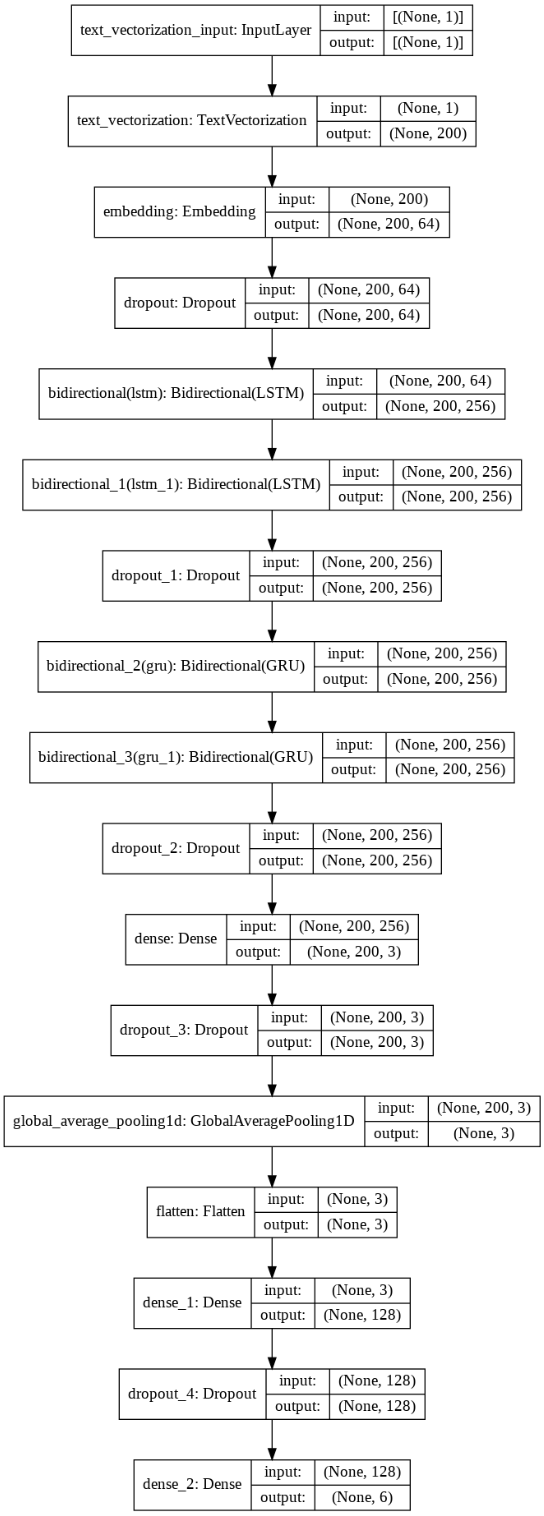


Figure 4: Architecture of the Text NN model

---

[15] `https://www.tensorflow.org/tutorials/text/word_embeddings`

Model specifics	Model name
Input layer, 3 dense layers with ReLU activation, output layer with softmax activation function	Audio NN 1 OS
Input layer, 3 dense layers with SELU activation, output layer with softmax activation function	Audio NN 2 OS
Input layer, 3 dense layers with ReLU activation, Dropout (0.3), output layer with softmax activation function	Audio NN 3 OS
Input layer, 3 dense layers with SELU activation, Dropout (0.3), output layer with softmax activation function	Audio NN 4 OS
2 LSTM layers, dense layer with ReLU activation, Dropout (0.2), output layer with softmax activation function	Audio NN 5 OS
2 BiLSTM layers, 2 BiGRU layers, dense layer with ReLU activation, Dropout (0.3),output layer with softmax activation function	Audio NN 6 OS
BiLSTM layer, BiGRU layer, dense layer with ReLU activation, Dropout (0.3), output layer with softmax activation function	Audio NN 7 OS
2 BiLSTM layers, dense layer with ReLU activation, Dropout (0.3), output layer with softmax activation function	Audio NN 8 OS
2 BiGRU layers, dense layer with ReLU activation, Dropout (0.3), output layer with softmax activation function	Audio NN 9 OS

Table 4: Different Neural Networks trained on the features extracted via *openSMILE*. The Model name listed is the name, we use, to refer to the specific model.

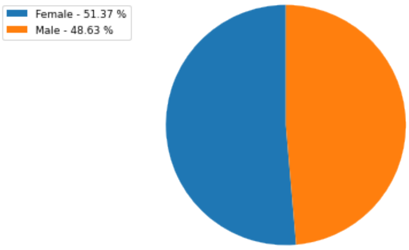


Figure 5: This Figure shows the spreading of female and male speakers on the data

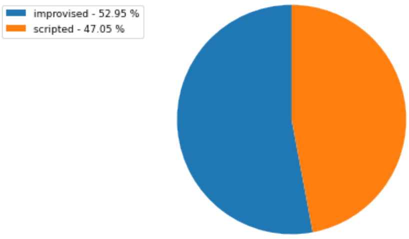


Figure 6: This Figure shows the spreading of improvised and scripted data

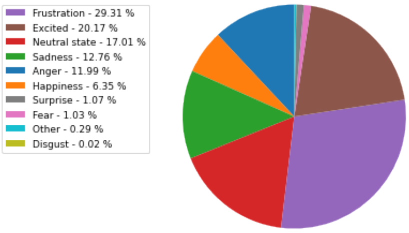


Figure 7: This Figure shows the spreading of the different classes on the data set

E Further details on results

Original Data

Class "Excited" achieves the highest precision (0.61). This aligns with the results of the combination of the Text NN with SVM SD (0.60) and the Text NN with SVM 100 (0.59). On all single models, class "Excited" reaches also the highest and second highest precision (Text NN 0.55, SVM SD 0.44 and SVM 31 0.45).

The second highest precision achieves class "Anger" (0.59), followed by classes "Frustration" and "Sadness" (both 0.52). Class "Frustration" gains the highest recall (0.72), followed by class "Excited" (0.62) and "Sadness" (0.59).

43.3% of all correctly predicted examples belong to class "Frustration", followed by class "Excited", which make up 23.8% of all correct predictions.

Data set variation 1

On the first combination, class "Happiness/Excited" reaches the highest precision of 0.85, followed by class "Anger/Frustration" (0.73) and "Sadness" (0.72). Class "Anger/Frustration" reaches the highest recall of 0.91, followed by class "Happiness/Excited" (0.65) and class "Sadness" (0.60). This leads to the highest F1 of 0.81 on class "Anger/Frustration", 0.74 on class "Happiness/Excited" and 0.65 on class "Sadness". Class "Surprise" neither has any correct predictions, nor are any examples predicted on this class at all.

Data set variation 2

Class "Anger/Frustration/Sadness" achieves the highest precision of 0.73, both other classes reach a slightly lower precision of 0.71. More differences are visible in the results of recall and F1. The highest recall (0.93) and therefore also the highest F1 (0.82) achieves class "Anger/Frustration/Sadness". Class "Happiness/Excited" reaches a recall of 0.64 and F1 of 0.67. The results on class "Neutral state, however, are even lower, with a recall of 0.20 and F1 of 0.32.

F Further combination methods

Additionally to the aforementioned combination, we combine the models based on the highest precision, recall and F1 on each class. This means, the model with the highest precision on class one is combined with the model with the highest precision on class two and so on. The same goes for recall and F1. The weighing, however, differs from the the first two approaches. In order to put more weight on the model for classes, for which it specifically gains better results, the model makes up 50% of the final prediction on this class. The other classes share the remaining 50% evenly.

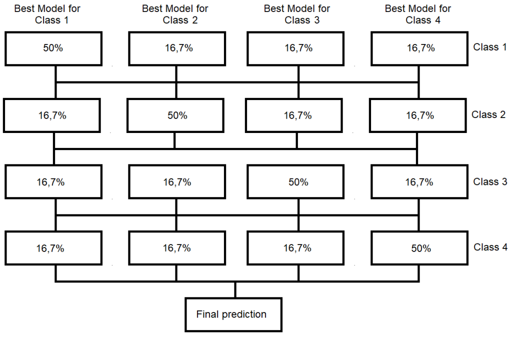


Figure 8: This Figure shows the way, the best models for each class are combined. For the class, the model performs best, it makes up 50%. The other remaining 50% percent are evenly separated between the other classes.

This means, on a data set variation with for classes (which is the case in data set variation 6 and 8), four models are combined, which can be seen in Figure 8. The best model for class one is weighing 50% in the final prediction of class one. The three models with the best performance for the other classes make up the remaining 50%. This happens for every class and on precision, recall and F1. In order to avoid to include models, which have most or even all examples of the whole data set classified on one class (recall near 1.0) or a model barely predicting a class, but if, then they are correct (precision near 1.00), we set a threshold in place. If the precision, recall of F1 is lower than 0.2, then the highest other measurement has to be lower than 0.90. This means, if, for example, the precision of a class reaches 1.00, then recall and F1 have to be at least 0.20. If not, then the model with the next lower precision is used, as long as it does not oppose to the aforementioned criteria with regard to the relation between precision, recall and F1.

Feature	Model name
Loudness	SVM 1
Alpha Ratio	SVM 2
Hammarberg Index	SVM 3
Slope 0-500	SVM 4
Slope 500-1500	SVM 5
Spectral Flux	SVM 6
MFCC 1	SVM 7
MFCC 2	SVM 8
MFCC 3	SVM 9
MFCC 4	SVM 10
F0 Semitone From 27.5Hz	SVM 11
Jitter Local	SVM 12
Shimmer Local dB	SVM 13
HNRdBACF	SVM 14
Log Rel F0-H1-H2	SVM 15
Log Rel F0-H1-A3	SVM 16
F1 Frequency	SVM 17
F1 Bandwith	SVM 18
F1 Amplitude Log Rel F0	SVM 19
F2 Frequency	SVM 20
F2 Bandwith	SVM 21
F2 Amplitude Log Rel F0	SVM 22
F3 Frequency	SVM 23
F3 Bandwith	SVM 24
F3 Amplitude Log Rel F0	SVM 25
all_frequ_ban_amp	SVM 26
all_max	SVM 27
all_min	SVM 28
all_avg	SVM 29
all_std	SVM SD
all_features	SVM 100

Table 6: Different classifiers trained on the features extracted via *openSMILE*

Model specifics	Model name
Flatten layer (input layer), 3 dense layers with ReLU activation, output layer with softmax activation function	Audio NN 1
Flatten layer (input layer), 3 dense layers with ReLU activation, Dropout (0.3), output layer with softmax activation function	Audio NN 2
Flatten layer (input layer), 3 dense layers with SELU activation, Dropout (0.3), output layer with softmax activation function	Audio NN 3
2 Conv2D layer (input layer), 3 MaxPooling2D layers with ReLU activation, 3 BatchNormalization layer, 1 dense layer with SELU activation, Flatten layer, Dropout (0.3), output layer with softmax activation function	Audio NN 4
2 LSTM layers, dense layer with ReLU activation, Dropout (0.3), output layer with softmax activation function	Audio NN 5
2 BiLSTM layers, 2 BiGRU layers, dense layer with ReLU activation, Dropout (0.3), output layer with softmax activation function	Audio NN 6
BiLSTM layer, BiGRU layer, dense layer with ReLU activation, Dropout (0.3), output layer with softmax activation function	Audio NN 7
2 BiLSTM layers, dense layer with ReLU activation, Dropout (0.3),output layer with softmax activation function	Audio NN 8
2 BiGRU layers, dense layer with ReLU activation, Dropout (0.3), output layer with softmax activation function	Audio NN 9

Table 5: Different Neural Networks trained on the features extracted via *librosa*. The Model name listed is the name, we use, to refer to the specific model.

Understanding Emotion Valence is a Joint Deep Learning Task

Gabriel Roccabruna, Seyed Mahed Mousavi, Giuseppe Riccardi
Signals and Interactive Systems Lab, University of Trento, Italy
`gabriel.roccabruna@unitn.it,giuseppe.riccardi@unitn.it`

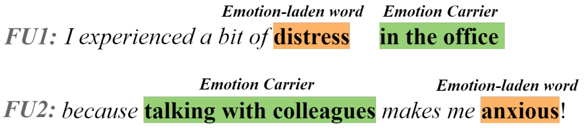
Abstract

The valence analysis of speakers' utterances or written posts helps to understand the activation and variations of the emotional state throughout the conversation. More recently, the concept of Emotion Carriers (EC) has been introduced to explain the emotion felt by the speaker and its manifestations. In this work, we investigate the natural inter-dependency of valence and ECs via a multi-task learning approach. We experiment with Pre-trained Language Models (PLM) for single-task, two-step, and joint settings for the valence and EC prediction tasks. We compare and evaluate the performance of generative (GPT-2) and discriminative (BERT) architectures in each setting. We observed that providing the ground truth label of one task improves the prediction performance of the models in the other task. We further observed that the discriminative model achieves the best tradeoff of valence and EC prediction tasks in the joint prediction setting. As a result, we attain a single model that performs both tasks, thus, saving computation resources at training and inference times.

1 Introduction

Speakers express their emotions in the language in different modalities (e.g. speech and/or text) and interaction contexts (e.g. dyadic interactions or social media posts). A type of document imbued with emotions conveyed through the recollection of personal events experienced by the speaker is the personal narrative. Personal Narratives (PN) have been recently studied to promote healthier mental health by modelling the patients' life events and monitoring emotional states (Mousavi et al., 2021; Danieli et al., 2021, 2022).

Monitoring the narrators' emotional states in PNs is achieved through valence analysis and the identification of related emotion carriers. Valence analysis addresses the identification of emotion levels ranging from pleasantness to unpleasantness generated by an event or a stimulus (Russell, 1980). The valence can be manifested explicitly via emotion-laden words, such as *Death* or *Birthday* in the PN. Besides emotion-laden words, valence can also be manifested implicitly through Emotion Carriers (EC), i.e. persons, objects or actions, that may not represent any emotion at the surface level (such as "the office" or "Wednesday"). Figure 1 shows an example of a sentence consisting of two Functional Units (FU1, FU2; the minimal span of text expressing a dialogue act (Bunt et al., 2012)) by Mousavi et al. (2022) with the emotion-laden words and the ECs in each unit. Recent studies show that ECs yield a detailed and understandable representation of the emotional state by highlighting the source of the valence such as "colleagues", "a vacation" or "a stroll along the river" (Tammewar et al., 2020; Mousavi et al., 2022).



FU1: I experienced a bit of **distress** **in the office**

FU2: because **talking with colleagues** *makes me* **anxious**!

Figure 1: Example of two Functional Units (FU1, FU2) by Mousavi et al. (2022). In each unit, the emotion-laden words convey an explicit emotion while the emotion carriers are implicit manifestations of emotions even though they represent neutral emotion at the surface level.

The two elements of valence and EC are interdependant since valence represents the intensity of the experienced emotions while the ECs are the means through which emotions are expressed and conveyed throughout the PN. Consequently, when narrators recount an event that activated their emotional state, the intensity of such emotion is manifested as valence while the expression of the emotion is through the recollection of the event and/or the participants characterising the activation of the emotional state.

In this work, we explore the natural interdependency of valence and the related ECs in PNs.

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 85–95
July 14, 2023 ©2023 Association for Computational Linguistics

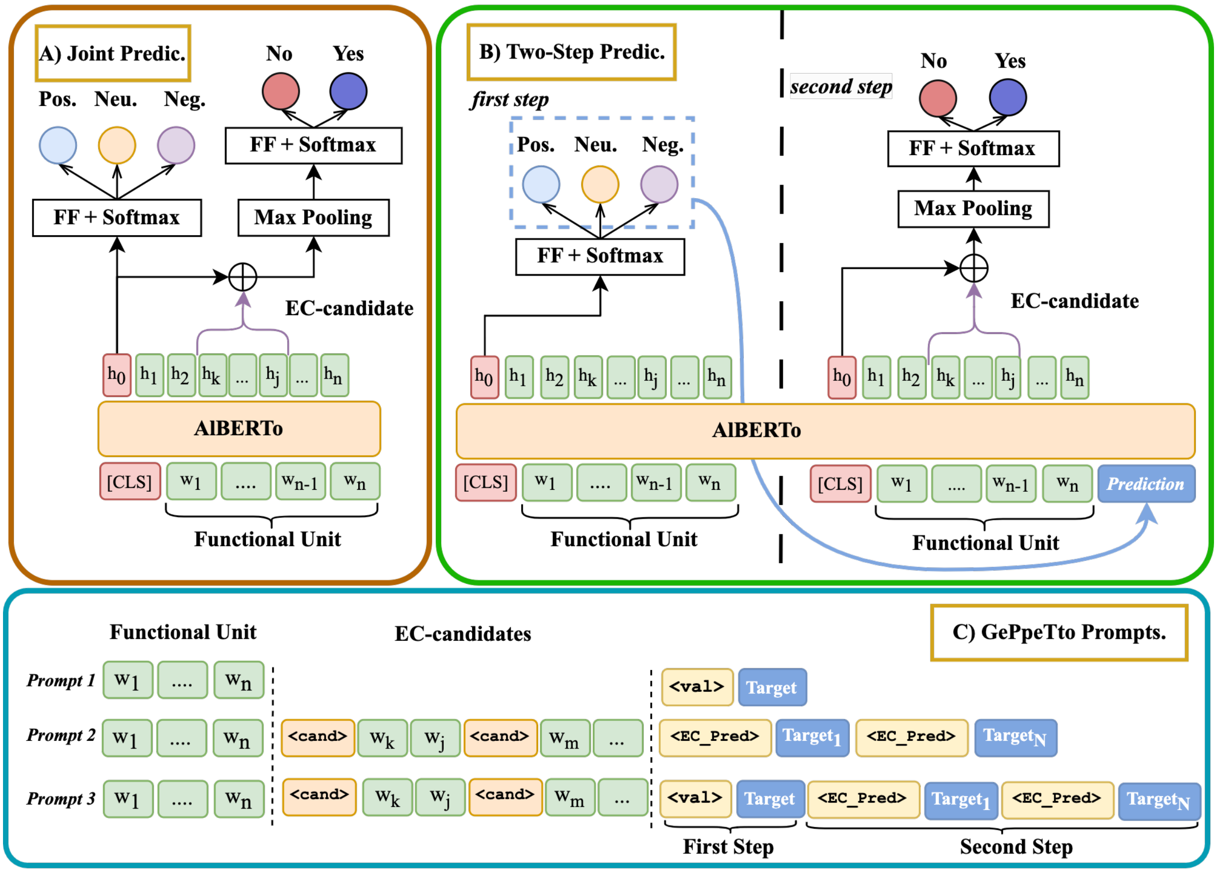

A) Joint Predic.
Pos.
Neu.
Neg.
No
Yes
FF + Softmax
FF + Softmax
Max Pooling
EC-candidate
h_0 h_1 h_2 h_k ... h_j ... h_n
AlBERTo
[CLS] w_1 .... w_{n-1} w_n
Functional Unit
B) Two-Step Predic.
first step
second step
Pos.
Neu.
Neg.
No
Yes
FF + Softmax
FF + Softmax
Max Pooling
EC-candidate
h_0 h_1 h_2 h_k ... h_j ... h_n
AlBERTo
[CLS] w_1 .... w_{n-1} w_n
Functional Unit
[CLS] w_1 .... w_{n-1} w_n Prediction
Functional Unit
C) GePpeTto Prompts.
Functional Unit
EC-candidates
Prompt 1 w_1 .... w_n <val> Target
Prompt 2 w_1 .... w_n <cand> w_k w_j <cand> w_m ... <EC_Pred> Target_1 <EC_Pred> Target_N
Prompt 3 w_1 .... w_n <cand> w_k w_j <cand> w_m ... <val> Target <EC_Pred> Target_1 <EC_Pred> Target_N
First Step
Second Step

Figure 2: The joint and two-step settings applied to discriminative (AlBERTo) and generative (GePpeTto) PLMs in three sections: A) fine-tuning AlBERTo with the joint prediction; B) the two-step prediction applied to AlBERTo (the first task is valence prediction and the second task is EC prediction); C) the prompts designed to fine-tune GePpeTto for valence prediction (prompt 1), EC prediction (prompt 2), and two-step approach (prompt 3).

This inter-dependency is characterised by the relations between the presence or absence of ECs and neutral or non-neutral valence. Namely, the presence of ECs in a FU implies a non-neutral valence, while neutral valence for a FU implies the absence of ECs. Moreover, the polarity of the valence might be related to the presence of domain-specific ECs. For instance, the ECs “the office” or “boss” might appear more frequently with a negative valence as opposed to “vacation” or “children”.

We investigate the contribution of this inter-dependency in the prediction of the valence and the related ECs in a Multi-Task Learning (MTL) approach. MTL is to train a single model on multiple related tasks to achieve inductive transfer between the tasks, which is to leverage additional information sources while learning the current task. Inductive transfer enhances generalisation by introducing an additional source of inductive bias used by the learner to prefer a hypothesis over the other hypothesis (Caruana, 1998). We experiment with two MTL approaches i.e. joint and two-step prediction. While in the joint approach, the labels of the valence and EC prediction tasks are predicted simultaneously, in the two-step approach, the prediction of one label is used as the context for the prediction of the other task.

We investigate whether this inter-dependency can be learned by Pre-trained Language Models (PLM). PLMs have prevailed over the other deep neural models in sentiment analysis (Mao et al., 2021; Roccabruna et al., 2022), and they have been effectively used as a backbone of MTL architectures achieving state-of-the-art performance in intent classification and slot filling (Chen et al., 2019; Qin et al., 2020), dialogue state tracking (Hosseini-Asl et al., 2020; Su et al., 2021) and aspect-based sentiment analysis (Mao et al., 2021; Jing et al., 2021).

We experiment with discriminative AlBERTo (BERT) as well as generative GePpeTto (GPT-2) models. In particular, for the discriminative model, we combine the two architectural solutions for valence and EC prediction tasks proposed by Mousavi et al. (2022). While for the generative model, we design two prompts for valence and EC prediction and one prompt for the two-step prediction setting. In these experiments, we evaluate both models for each MTL setting, where the baseline is the performance of the model fine-tuned on every task separately. Moreover, we compute the upper bound in the two-step prediction setting by replacing the first prediction with the ground truth. This upper bound also represents the level of inter-dependency between the two tasks. In this work, we use a corpus of PNs in Italian with valence and ECs annotated at the functional unit level.

The contributions of this paper can be summarized as follows:

- We study the inter-dependency of the valence and related Emotion Carriers in the corresponding prediction tasks;
- We fine-tune two PLMs and experiment with multi-task learning settings for valence and Emotion Carrier predictions;
- We evaluate and compare the performance of discriminative and generative models in the task of valence and Emotion Carrier prediction.

2 Related Works

Valence & Sentiment Analysis The values of valence have been studied both in a continuous space (Ong et al., 2019; Kossaifi et al., 2017), and discrete space with a Likert scale (Tammewar et al., 2022; Mousavi et al., 2022), ranging from negative (unpleasant) to positive (pleasant). Using the discrete approach, valence can be assessed with different levels of granularity as the narrative level and functional unit level. A Functional Unit (FU) is the minimal span of text expressing a dialogue act (Bunt et al., 2012; Roccabruna et al., 2020). Narrative-level valence analysis provides a general yet flat perspective of the narrators' emotional state (Schuller et al., 2018), meanwhile, the sentence-level and FU-level analysis provide a detailed perspective as it highlights the variations and fluctuations of the valence throughout the narrative (Mousavi et al., 2022).

A common practice in developing models for emotion analysis is to model valence analysis as sentiment analysis by mapping the valence values into three sentiment classes, i.e. *positive*, *negative*, and *neutral* (Roccabruna et al., 2022; Mousavi et al., 2022). However, valence differs from sentiment as the latter identifies the polarity of attitudes or beliefs, such as *hating* or *liking*, towards a person (e.g. a politician) or an object (e.g. product or a movie) (Scherer et al., 2000). Meanwhile, valence represents the level of emotions in such as *anger* or *happiness*.

Emotion Carrier and Valence: Emotion carriers are closely related to emotional valence as they explain the valence. Mousavi et al. (2022) studied the correlation between the sequence tokens and the predicted valence (sentiment). The authors observed that the model focuses more on emotion-laden words (explicit), whereas humans identify the emotion carriers to explain the valence.

Multi-task Learning: Multi-Task Learning (MTL) has been used for affective computing in aspect-based sentiment analysis (Schmitt et al., 2018; Mao et al., 2021; Jing et al., 2021), and emotion classification and emotion cause extraction (Turcan et al., 2021). MTL has been studied using discriminative models (BERT) for entity-relation extraction(Xue et al., 2019), as well as generative models (GPT-2) for task-oriented dialogues (Hosseini-Asl et al., 2020), dialogue state tracking (Zhao et al., 2021), and task-oriented response generation (Su et al., 2022).

3 Approach

The inter-dependency of the valence and ECs results in the co-occurrence of both elements in the same utterance as the valence represents the level of emotion and the ECs are expressions through which the emotion is conveyed. In other words, the neutral valence of an utterance implies the absence of ECs, while the presence of ECs indicates a valence polarity for the same utterance. Furthermore, the ECs in an utterance can provide insights into the polarity of the utterance valence since certain ECs are more often associated with negative emotions (such as deadline) or positive ones (such as graduation).

We investigate the natural inter-dependency of valence and ECs via the MTL approach, which allows the models to leverage additional information learned from other tasks while learning the current task, improving the generalization. We experiment with joint training, where the two labels of valence

Positive	Negative	Intersection
"perfume"	"to tackle"	"work"
"vacations"	"administration"	"home"
"yoga"	"dentist"	"lunch"
"a stroll"	"be late"	"today"
"freedom"	"charged with"	"feeling"

Table 1: Examples of Emotion Carriers (EC) in positive and negative functional units of the dataset used (English translations). "Intersection" consists of the ECs that are present in both positive and negative FUs.

and ECs are predicted simultaneously, and two-step prediction, where one of the labels is predicted and used by the model to condition and guide the prediction of the second label.

3.1 Dataset

We use a corpus of written PNs collected and annotated by Mousavi et al. (2022). The dataset consists of 481 narratives from 45 subjects, who were employees with stress, with valence and EC annotations at the Functional Unit (FU). Out of 4273 FUs in the narratives, 40% are annotated by polarity (13% positive and 27% negative) and the related ECs, while the rest are annotated as neutral and do not contain any EC. Considering both neutral and non-neutral FUs, 18.5% of the span candidates are annotated as ECs (over 10763 span candidates). While the number of ECs considering only non-neutral FUs is 44.7% over 4452 span candidates. We use the official splits of the dataset, provided by the authors, consisting of train (80%), validation (10%) and test (10%) set, stratified on the polarity distribution.

To measure the number of ECs specific for a valence polarity, we started by computing the intersection of ECs set annotated in FUs with positive and negative valence. We observed that only 4% of the ECs (14.8% from the positive and 6% from the negative sets) are present in FUs with both polarities and can convey both positive and negative emotions. That is, the majority of ECs (the remaining 94%) are indicators and carriers of only one valence polarity. Table 1 presents a representative sample of the ECs extracted from the three sets.

3.2 Multi-Task Learning

We experiment with Multi-Task Learning (MTL) approach to exploit the dependency between the valence and EC prediction tasks. We compare the performance of the models for EC and valence predictions as task-specific models, as well as two-step and joint-prediction models, and evaluate the performance.

Single-Task Prediction In the single-task prediction, the models are trained and optimized for each task separately. This modality is a baseline to compare model performance in other MTL settings such as two-step and joint prediction.

Two-step Prediction Inspired by Kulhánek et al. (2021) and Hosseini-Asl et al. (2020), we experiment with the two-step prediction setting. In this setting, the model predicts the discrete label for the first task as the first step, and as the second step, this prediction is concatenated to the input sequence following a prompt structure to predict the label for the second task via the same model. Afterwards, the loss values of the two tasks are summed or aggregated with a linear interpolation before back-propagating it. In this setting, we experiment with alternating the order for the two tasks (Valence → EC, vs. EC → Valence).

The motivations behind this setting are that 1) the contribution of one task over the other task is explicit, enhancing the understanding of the interdependency between the two tasks; 2) this approach can potentially reduce the gap in the performance between two interdependent tasks by conditioning the prediction of the second task with the prediction of the first task. In this, the best-performing task is placed in the first step. To provide evidence of this, we experimented by replacing the first step with an oracle providing the ground truth.

The two-step setting is similar to the pipeline setting (Zhang and Weiss, 2016) and Stack-Propagation framework (Zhang and Weiss, 2016; Qin et al., 2019). The two-step prediction is similar to the pipeline setting because the discrete output of a task is explicitly used in the prediction of another task, but in the pipeline setting, two different models are utilised instead of one. Moreover, the two-step prediction is close to the Stack-Propagation framework as the back-propagation of the loss updates the weights of the model used to predict the first task as well as the second task. However, the label guiding the model's prediction of the second task is not differentiable as in the Stack-Propagation.

Joint Prediction The joint prediction setting is commonly used in MTL (Cerisara et al., 2018; Jing et al., 2021) where a single model predicts labels for the different tasks simultaneously. Thus, the prediction of one task does not explicitly contribute to the prediction of another task. The loss is computed and back-propagated as in the two-step prediction approach.

4 Models

We experiment with discriminative (BERT) and generative (GPT-2) models and investigate the performance of the two models for the joint and two-step prediction of valence and ECs. The joint and two-step prediction settings along with the two architectures are depicted in Figure 2. All the hyperparameters and model settings are reported in Appendix 8 to achieve the reproducibility of the results.

4.1 Prediction tasks

In the discriminative model, we formalize the valence and Emotion Carrier prediction tasks as text classification tasks by following the formalization of Mousavi et al. (2022). The valence prediction task is formally defined as calculating the probability for a given functional unit as $p(valY_i|FU_i)$, where label $valY_i \in \{positive, negative, neutral\}$ and $FU_i = \{w_1, w_2, .., w_n\}$ as a sequence *i* tokens w. Meanwhile, the EC prediction task is to predict for each EC candidate span, which is an automatically extracted verb or noun chunk, in a FU if it is an EC or not. That is $p(ecY_j|EC\text{-}candidate_j, FU_i)$ where the FU provides context information for the prediction, $EC\text{-}candidate_j = \{w_k, .., w_l\} \in FU$ and $ecY_j \in \{yes, no\}$.

In the generative model, inspired by (Hosseini-Asl et al., 2020), we model the two tasks as causal language modelling tasks, in which the model is tasked to learn the joint probability over a sequence. For valence prediction, the sequence used to train the model is formally defined as $x_i = [FU_i; valY_i]$, i.e. the concatenation of FU_i and $valY_i$, where the functional unit FU_i is the context for the model in the prediction of $valY_i \in \{positive, negative, neutral\}$. While the training sequence for the EC prediction task is $x_i = [FU_i; EC\text{-}candidate_i; ecY_i]$ where FU_i is the functional, $EC\text{-}candidate_i$ is the complete list of the EC-candidate spans of FU_i, and ecY_i is the list of the EC decision labels, i.e. $\{yes, no\}$, corresponding to the list EC-candidate span. In this sequence, both FU_i and $EC\text{-}candidate_i$ is the context for the model in the prediction of ecY_i.

4.2 Discriminative

Discriminative models based on PLMs have been effectively used for text classification tasks (Lei et al., 2019), however, such models may need additional architectural components, such as conditional random fields and/or additional feed-forward layers (Shang et al., 2021), to tackle a specific task.

Architecture Our discriminative model is based on the same architectural components for valence and EC predictions proposed by Mousavi et al. (2022). This architecture is composed of a PLM and a set of feed-forward layers used to make the prediction. The PLM is based on AlBERTo which is BERT-based with 110M parameters pre-trained on a corpus in the Italian language collected from Twitter (Polignano et al., 2019). The PLM takes as input a FU with special tokens `[CLS]` and `[SEP]`, added at the head and the tail of the FU, and returns a sequence of hidden states. The valence is predicted from the hidden state of the `[CLS]` token by first applying a feed-forward layer with softmax to compute the probabilities over the classes of the valence (*positive*, *negative* and *neutral*). For the EC prediction, an EC-candidate span is represented by a set of hidden states corresponding to the tokens of the span. Furthermore, the hidden state of the `[CLS]` token is concatenated to the hidden states of the EC span to give context information contained in the FU. These hidden states are passed through a max-pooling layer, to get the vector representation of the EC candidate, and a feed-forward layer with softmax to yield the prediction on the two classes (*yes* and *no*).

Joint Regarding the join prediction setting, as depicted in Figure 2 part *A)*, the valence and ECs are predicted in one step. In this, the shared part of the model between the two tasks is the PLM AlBERTo only.

Two-step The model in the two-step prediction setting, Figure 2 part *B)*, has the same shared parts of the joint model, but the prediction of valence and ECs are done in two steps. The prediction of the first task is computed on a FU, while the prediction for the second task is computed on the concatenation of the FU with the label predicted in the first task. The prompt that concatenates the prediction when the first task is valence prediction is:

$$FU_i = \{w_1, w_2, .., w_n\}$$

$$FU_i \oplus \textit{valence:} \oplus Prediction_i$$

where i is a functional unit of the dataset, $\oplus$ is the concatenation by white space, *valence:* is plain text, $Prediction_i = (0|1|2)$, and $\{0, 1, 2\}$ are the labels *negative*, *positive* and *neutral* respectively. When the first task is EC prediction the prompt is

$$EC_j = \{w_k, .., w_l\} \in FU_i$$

$$FU_i \oplus \textit{EC:} \oplus \{EC_1, .., EC_N\}$$

where *EC:* is plain text, and $\{EC_1, .., EC_N\}$ is the list of EC spans of the detected ECs in the FU_i. Furthermore, to reduce the training time and stabilise the performance, we experiment with the teacher forcing technique (Lamb et al., 2016) that substitutes the prediction of the first task with the ground truth with a certain probability (to be selected as a hyperparameter).

Loss function In both joint and two-step prediction settings, the loss function is the cross entropy and the loss values of the two tasks are combined with a linear interpolation:

$$loss_{total} = \lambda * (loss_{valence}) + (1 - \lambda) * loss_{EC}$$

where λ is a hyperparameter with a range from 0 to 1.

4.3 Generative

We used GePpeTto (De Mattei et al., 2020) an auto-regressive model based on GPT-2 architecture which is pre-trained for the Italian language with 117M parameters. For valence, EC and two-step predictions, we have designed three prompts by following the formalization of the two tasks.

Prompt design The prompt for valence prediction is composed of two segments, where the first segment is the FU_i and the second segment is the valence label $valY_i$ to predict preceded by a special token. This prompt is depicted in Figure 2 part *C) prompt 1*, where $Target = (0|1|2)$ indicating *negative*, *positive* and *neutral* respectively. The prompt used for EC prediction is organised into three segments: a) FU_i; b) $EC\text{-}candidate_i$ spans separated by a special token and; c) the list of labels corresponding to each EC candidate ecY_i separated by another different special token. This prompt is shown in Figure 2 part *C) prompt 2* where $Target = (y|n)$. A difference with the discriminative model is that in EC prediction the predicted label of one EC candidate is used as context to predict the next EC candidates due to the fact that the model is auto-regressive.

Joint In the joint prediction setting, we fine-tune a single model on valence and EC predictions using the two corresponding distinct prompts appearing in the same training batch. Thus, the prediction of one task does not occur in the context of the other task.

Two-step For the two-step prediction, we designed a specific prompt by combining the prompts for valence and EC predictions, which is composed of the 4 segments: a) FU_i, b) $EC\text{-}candidate_i$ spans, c) valence $valY_i$ and d) ECs ecY_i targets, Figure 2 part *C) prompt 3*. The first two segments are the FU_i and $EC\text{-}candidate_i$. The other two segments are the target labels of the valence and EC predictions tasks. Thus, in the first step, the model predicts, based on FU_i and $EC\text{-}candidate_i$, the labels of the first task that are used as context in the second step to predict the other task. Moreover, alternating the order of the last two segments results in two prediction settings Valence → EC and EC → Valence.

Generation strategy At inference time, the generation of the target is guided by forcing the special tokens, i.e. they are not predicted by the model, into the generated sequence and limiting the possible output labels by considering the probabilities of the tokens in our searching space i.e. $\{0, 1, 2\}$ for valence prediction and $\{y, n\}$ for the EC prediction. Moreover, for the EC prediction task, we force the same number of special tokens of the EC candidates to get one output label for each EC candidate, relieving the model from the complexity of counting the EC candidates.

Loss function The generative model is trained as a language model, i.e. the model is tasked to predict the next most probable word given a sequence of words. In this, the loss function is the cross entropy with the objective of minimizing the perplexity on the training set.

5 Experiments

Table 2 presents the macro F1-score achieved by AlBERTo and GePpeTto models with single-task, two-step and joint prediction settings.

The results on valence prediction achieved via the single-task discriminative model are on-par with those reported by Mousavi et al. (2022), while the results achieved on EC prediction are incomparable with Mousavi et al. (2022) since our training set consists of all FUs with the authors train the models only using the FU with a valence polarity.

Valence Prediction

Model	Single Task	Two-Step			Joint
		Val. → EC	EC → Val.	w. ground truth	
AlBERTo	76.0	76.0	75.7	81.2	76.0
GePpeTto	77.1	74.7	65.1	86.5	75.6
+ domain adapt.	-	77.0	70.6	-	-
EC Prediction					
AlBERTo	63.7	63.4	64.8	74.9	65.0
GePpeTto	57.8	58.3	58.2	65.4	59.5
+ domain adapt.	-	59.5	60.7	-	-

Table 2: The macro-F1 scores (average over 10 runs) in percentage of the Valence Prediction and Emotion Carrier (EC) Prediction tasks. The scores are achieved using discriminative (AlBERTo) and generative (GePpeTto) PLMs with single-task, two-step and joint prediction settings. Single task and two-step with (w.) ground truth are respectively the baseline and the upper-bound for the joint and the two-step settings.

Regarding the single task setting, we observe that AlBERTo outperforms GePeTto for EC prediction, while GePpeTto outperforms AlBERTo on the valence prediction task.

Regarding the two-step prediction setting Val → EC, we observe a slight worsening in the performance of EC prediction for the AlBERTo model and a drop in valence prediction score for the generative model compared to the single-task setting. Nevertheless, domain adaptation improves the performance to achieve close results on the valence prediction and boosts the performance on EC prediction. Domain adaptation is performed by initially fine-tuning only on the first task and further fine-tuning on both tasks with the two-step approach.

Regarding the reverse order of predictions, i.e. EC → Val., the models do not manage to outperform the single-task alternatives on valence prediction, with the degradation being more significant for the generative model. On the contrary, the models exhibit a better performance for EC prediction in the two-step setting compared to single-task models, where GePpeTto with domain adoption achieves its best performance in all settings.

Additionally, we have computed the upper bound for the two-step prediction by substituting the prediction of the first step with the corresponding ground truth and fine-tuning the model only on the second step. The results show a solid contribution of one task in predicting the other task by outperforming the models in all the other prediction settings. Furthermore, GePpeTto and AlBERTo achieve the highest performance in valence prediction and EC prediction respectively.

The results of the discriminative model in the joint prediction setting are on par and better than the other two settings for valence and EC predictions, respectively. While the performance of the generative model is worse than the two-step prediction for both tasks, but slightly better than the single task in EC prediction. In particular, AlBERTo achieves the highest macro F1-score on the EC prediction task compared to the other settings and the generative model.

6 Discussion

The inter-dependency between valence prediction and EC prediction tasks is quantified in terms of performance by the upper bound computed by substituting the first step prediction with the ground truth. We observe that the performance of both tasks is enhanced. Furthermore, we observe that the proposed MTL approaches, i.e. two-step and joint prediction, are effective in exploiting such inter-dependency. In particular, the two-step prediction boosts the performance of the generative model, while joint prediction improves the performance of the discriminative model. Moreover, we observe that these improvements affect mainly the EC prediction task. This is because, compared to valence prediction, the EC prediction task is objectively more challenging for the models due to the unbalanced distribution and the sparsity of the ECs (some ECs are personal w.r.t the narrator). Thus, the predictions of the EC tasks are too noisy to be exploited by the models to improve the performance of valence prediction. Indeed, the worst performance for valence prediction is achieved by both models in the two-step setting in which the first task is the EC prediction (i.e. EC → Val.).

Regarding the comparison between discriminative and generative models, the best trade-off between valence and EC prediction tasks is achieved by AlBERTo, although GePpeTto fine-tuned with the single-task setting achieves the best performance on valence prediction.

7 Conclusions

In this work, we studied the inter-dependency between valence and ECs in personal narratives. For valence and EC prediction tasks, we compared task-specific models with two MTL settings, namely joint and two-step prediction. We experimented with discriminative and generative PLMs. The results indicate that PLMs fine-tuned with MTL settings achieve improved performance by exploiting the inter-dependency between valence and EC prediction tasks. In particular, the two-step setting is more effective for the generative model, while the joint setting best fits the discriminative model. Furthermore, the generative model outperforms the discriminative model on the valence prediction task, while the discriminative model achieves better results on EC prediction and the best trade-off between the valence and EC prediction tasks. Consecutively, one discriminative model performs the two tasks, reducing the demand for computational resources at training and inference time and, therefore, lowering carbon emissions in the environment.

8 Future works

In the two-step prediction setting, we have only experimented with unidirectional inter-dependency of the two tasks i.e. EC $\rightarrow$ Val. and Val. $\rightarrow$ EC. A possible future work is to design and experiment with a neural network that bidirectionally exploits the two predictions implementing the configuration Val. $\leftrightarrow$ EC. However, a larger dataset with more narratives per narrator is needed as a positive contribution of ECs to the Valence prediction task is observed on *with ground truth* setting only due to the fact that ECs are sparse in the corpus and specific w.r.t the narrators.

Limitations

The dataset used in this work is in Italian and the PLMs are pre-trained for the Italian language. The performance of the models and the results may be influenced by language-specific properties.

To reduce the ECs sparsity and, therefore, better modelling the inter-dependency between EC and Valence prediction tasks, particularly in the experiments EC $\rightarrow$ Val., a larger dataset with more narratives per narrator is needed.

Acknowledgement

We acknowledge the support of the MUR PNRR project FAIR - Future AI Research (PE00000013) funded by the NextGenerationEU and the PNRR Project “Interconnected Nord-Est Innovation Ecosystem (iNEST)”, ECS00000043.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Harry Bunt, Jan Alexandersson, Jae-Woong Choe, Alex Chengyu Fang, Kôiti Hasida, Volha Petukhova, Andrei Popescu-Belis, and David R. Traum. 2012. Iso 24617-2: A semantically-based standard for dialogue annotation. In *International Conference on Language Resources and Evaluation*.

Rich Caruana. 1998. *Multitask learning*. Springer.

Christophe Cerisara, Somayeh Jafaritazehjani, Adedayo Oluokun, and Hoa T. Le. 2018. Multi-task dialog act and sentiment recognition on mastodon. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 745–754, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Qian Chen, Zhu Zhuo, and Wen Wang. 2019. Bert for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909*.

Morena Danieli, Tommaso Ciulli, Seyed Mahed Mousavi, and Giuseppe Riccardi. 2021. A conversational artificial intelligence agent for a mental health care app: Evaluation study of its participatory design. *JMIR Formative Research*, 5(12):e30053.

Morena Danieli, Tommaso Ciulli, Seyed Mahed Mousavi, Giorgia Silvestri, Simone Barbato, Lorenzo Di Natale, and Giuseppe Riccardi. 2022. Assessing the impact of conversational artificial intelligence in the treatment of stress and anxiety in aging adults: Randomized controlled trial. *JMIR Mental Health*, 9(9):e38067.

Lorenzo De Mattei, Michele Cafagna, Felice Dell’Orletta, Malvina Nissim, and Marco Guerini. 2020. Geppetto carves italian into a language model. *arXiv preprint arXiv:2004.14253*.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. *Advances in Neural Information Processing Systems*, 33:20179–20191.

Hongjiang Jing, Zuchao Li, Hai Zhao, and Shu Jiang. 2021. Seeking common but distinguishing difference, a joint aspect-based sentiment analysis model. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3910–3922.

Jean Kossaifi, Georgios Tzimiropoulos, Sinisa Todorovic, and Maja Pantic. 2017. Afew-va database for valence and arousal estimation in-the-wild. *Image Vision Comput.*, 65(C):23–36.

Jonáš Kulhánek, Vojtěch Hudeček, Tomáš Nekvinda, and Ondřej Dušek. 2021. AuGPT: Auxiliary tasks and data augmentation for end-to-end dialogue with pre-trained language models. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 198–210, Online. Association for Computational Linguistics.

Alex M Lamb, Anirudh Goyal ALIAS PARTH GOYAL, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio. 2016. Professor forcing: A new algorithm for training recurrent networks. *Advances in neural information processing systems*, 29.

Jiahuan Lei, Qing Zhang, Jinshan Wang, and Hengliang Luo. 2019. Bert based hierarchical sequence classification for context-aware microblog sentiment analysis. In *Neural Information Processing: 26th International Conference, ICONIP 2019, Sydney, NSW, Australia, December 12–15, 2019, Proceedings, Part III 26*, pages 376–386. Springer.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Yue Mao, Yi Shen, Chao Yu, and Longjun Cai. 2021. A joint training dual-mrc framework for aspect based sentiment analysis. In *AAAI Conference on Artificial Intelligence*.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. 2021. On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines. In *9th International Conference on Learning Representations*, CONF.

Seyed Mahed Mousavi, Roberto Negro, and Giuseppe Riccardi. 2021. An unsupervised approach to extract life-events from personal narratives in the mental health domain.

Seyed Mahed Mousavi, Gabriel Roccabruna, Aniruddha Tammewar, Steve Azzolin, and Giuseppe Riccardi. 2022. Can emotion carriers explain automatic sentiment prediction? a study on personal narratives. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 62–70, Dublin, Ireland. Association for Computational Linguistics.

Desmond C Ong, Zhengxuan Wu, Zhi-Xuan Tan, Marianne Reddan, Isabella Kahhale, Alison Mattek, and Jamil Zaki. 2019. Modeling emotion in complex stories: the stanford emotional narratives dataset. *IEEE Transactions on Affective Computing*, 12(3):579–594.

Marco Polignano, Pierpaolo Basile, Marco de Gemmis, Giovanni Semeraro, and Valerio Basile. 2019. Alberto: Italian bert language understanding model for nlp challenging tasks based on tweets. In *CEUR Workshop Proceedings*, volume 2481, pages 1–6. CEUR.

Libo Qin, Wanxiang Che, Yangming Li, Haoyang Wen, and Ting Liu. 2019. A stack-propagation framework with token-level intent detection for spoken language understanding. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2078–2087, Hong Kong, China. Association for Computational Linguistics.

Libo Qin, Tailu Liu, Wanxiang Che, Bingbing Kang, Sendong Zhao, and Ting Liu. 2020. A co-interactive transformer for joint slot filling and intent detection. *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8193–8197.

Gabriel Roccabruna, Steve Azzolin, and Giuseppe Riccardi. 2022. Multi-source multi-domain sentiment analysis with BERT-based models. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 581–589, Marseille, France. European Language Resources Association.

Gabriel Roccabruna, Alessandra Cervone, and Giuseppe Riccardi. 2020. Multifunctional iso standard dialogue act tagging in italian. In *CLiC-it*.

James A Russell. 1980. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161.

Klaus R Scherer et al. 2000. Psychological models of emotion. *The neuropsychology of emotion*, 137(3):137–162.

Martin Schmitt, Simon Steinheber, Konrad Schreiber, and Benjamin Roth. 2018. Joint aspect and polarity classification for aspect-based sentiment analysis with end-to-end neural networks. In *Conference on Empirical Methods in Natural Language Processing*.

Björn Schuller, Stefan Steidl, Anton Batliner, Peter B. Marschik, Harald Baumeister, Fengquan Dong, Simone Hantke, Florian B. Pokorny, Eva-Maria Rathner, Katrin D. Bartl-Pokorny, Christa Einspieler, Dajie Zhang, Alice Baird, Shahin Amiriparian, Kun Qian, Zhao Ren, Maximilian Schmitt, Panagiotis Tzirakis, and Stefanos Zafeiriou. 2018. The interspeech

2018 computational paralinguistics challenge: Atypical & self-assessed affect, crying & heart beats. In *Proc. Interspeech 2018*, pages 122–126.

Xichen Shang, Qianli Ma, Zhenxi Lin, Jiangyue Yan, and Zipeng Chen. 2021. A span-based dynamic local attention model for sequential sentence classification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 198–203, Online. Association for Computational Linguistics.

Yixuan Su, Lei Shu, Elman Mansimov, Arshit Gupta, Deng Cai, Yi-An Lai, and Yi Zhang. 2021. Multi-task pre-training for plug-and-play task-oriented dialogue system. In *Annual Meeting of the Association for Computational Linguistics*.

Yixuan Su, Lei Shu, Elman Mansimov, Arshit Gupta, Deng Cai, Yi-An Lai, and Yi Zhang. 2022. Multi-task pre-training for plug-and-play task-oriented dialogue system. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4661–4676.

Aniruddha Tammewar, Franziska Braun, Gabriel Roccabruna, Sebastian Bayerl, Korbinian Riedhammer, and Giuseppe Riccardi. 2022. Annotation of valence unfolding in spoken personal narratives. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 7004–7013.

Aniruddha Tammewar, Alessandra Cervone, Eva-Maria Messner, and Giuseppe Riccardi. 2020. Annotation of emotion carriers in personal narratives. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1517–1525.

Elsbeth Turcan, Shuai Wang, Rishita Anubhai, Kasturi Bhattacharjee, Yaser Al-Onaizan, and Smaranda Muresan. 2021. Multi-task learning and adapted knowledge models for emotion-cause extraction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3975–3989.

Kui Xue, Yangming Zhou, Zhiyuan Ma, Tong Ruan, Huanhuan Zhang, and Ping He. 2019. Fine-tuning bert for joint entity and relation extraction in chinese medical text. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 892–897. IEEE.

Yuan Zhang and David Weiss. 2016. Stack-propagation: Improved representation learning for syntax. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1557–1566, Berlin, Germany. Association for Computational Linguistics.

Jeffrey Zhao, Mahdis Mahdieh, Ye Zhang, Yuan Cao, and Yonghui Wu. 2021. Effective sequence-to-sequence dialogue state tracking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7486–7493.

Model	Parameter	Single Task		Two-Step		Joint
		Valence Pred.	EC Pred.	Val. → EC	EC → Val.	
AlBERTo	Learning Rate	5e-5	4e-5	4e-5	6e-5	1e-5
AlBERTo	λ	-	-	0.5	0.4	0.3
GePpeTto	Learning Rate	9e-3	8e-3	9e-4	7e-4	8e-3

Table 3: List of hyperparameters used to fine-tune the two models.

Appendix

A Hyperparameters

The special tokens used in the prompts, `<val>`, `<cand>` and `<EC_pred>`, are added to the vocabulary of the model. Moreover, we encoded valence textual labels with numbers because they are language-independent and perform better than additional special tokens.

We used AdamW (Loshchilov and Hutter) as an optimization algorithm to fine-tune the discriminative and generative models. To stabilise the performance while fine-tuning, we used a linear warm-up scheduler on the learning rate with the warm-up steps set at 10% of the total training steps (Mosbach et al., 2021). We used the library Optuna optimizer (Akiba et al., 2019) to search for the best hyperparameter for each setting (single prediction, joint prediction and two-step prediction) and models (discriminative and generative). The complete list of learning rates is presented in Table 3. The learning rates used in the two-step with ground truth are the same as Val. → EC when the first step is valence prediction and EC → Val. when the first task is EC prediction. Moreover, we used a batch size of 32 for both models, 30 epochs for AlBERTo and 60 epochs for GePpeTto and, early stopping with patience set to 5 epochs. In the two-step experiments, we used a teacher forcing probability of 1.0 in Val. → EC and 0.1 in EC → Val. . We trained our models using one single 3090Ti GPU.

Czech-ing the News: Article Trustworthiness Dataset for Czech

Matyáš Boháček[1,2], Michal Bravanský[1,3], Filip Trhlík[1,3], and Václav Moravec[1]

[1]Faculty of Social Sciences, Charles University, Prague, Czech Republic
[2]Gymnasium of Johannes Kepler, Prague, Czech Republic
[3]University College London, United Kingdom

`matyas.bohacek@matsworld.io, michal@bravansky.com,`
`me@trhlikfilip.com, vaclav.moravec@fsv.cuni.cz`

Abstract

We present the Verifee dataset: a multimodal dataset of news articles with fine-grained trustworthiness annotations. We bring a diverse set of researchers from social, media, and computer sciences aboard to study this interdisciplinary problem holistically and develop a detailed methodology that assesses the texts through the lens of editorial transparency, journalist conventions, and objective reporting while penalizing manipulative techniques. We collect over $10,000$ annotated articles from 60 Czech online news sources. Each item is categorized into one of the 4 proposed classes on the credibility spectrum – ranging from entirely trustworthy articles to deceptive ones – and annotated of its manipulative attributes. We fine-tune prominent sequence-to-sequence language models for the trustworthiness classification task on our dataset and report the best F-1 score of 0.53. We open-source the dataset, annotation methodology, and annotators' instructions in full length at `https://www.verifee.ai/research/` to enable easy build-up work.

1 Introduction

Donald Trump has called journalists and news outlets "fake news" nearly $2,000$ times since the beginning of his presidency, averaging more than one daily broadside against the press between 2016 and 2020 (Woodward, 2020). Because of Trump, the term fake news underwent a fundamental change in its meaning. At first, it referred to a satirical and ironic genre of fictional news designed to entertain the audience. The original "fake news" have appeared on TV shows such as Saturday Night Live on NBC or in print, such as The Onion. However, during Trump's campaign for the US presidential election in 2016 and his presidency, the concept of fake news became an integral part of his political communication. It aimed to discredit critical journalistic content or the whole news media as "fake media." The successful stigmatization strategy of "fake news" has led to a fascination with this phenomenon in the public discourse and science.

Fake news has become a label for false news and a synonym for both disinformation and misinformation. This has strengthened the binary perception of the credibility of information in a true-false dichotomous perspective. However, this reductionist approach has become a barrier to understanding the more profound meaning that the buzzword "fake news" covers. If we want to examine the credibility of the news content seriously, it is not possible to adopt the binary approach of either truth or lie. By creating the Verifee dataset, we try to overcome the interdisciplinary barrier between social sciences (especially journalism and media studies) and computer science. This barrier prevents specialists in automated or robotic journalism from adopting a more analytical approach to various types of information disorders that we have become used to labelling with the general term "fake news".

2 Related Work

Herein, we first review the current literature focusing on disinformation and misinformation in the journalistic ambit. We later provide an overview of existing methods treating these phenomena within the Artificial intelligence (AI) and Natural language processing (NLP) research communities. We first list some of the already available datasets and then focus on the architectures solving the tasks of fake news detection and automatic fact-checking.

The task of fake news detection resides in classifying whether a given news article (or another medium, such as a Tweet) is considered fake (disinformative) or truthful (credible). There is no consensus in the literature on what specific parameters constitute this distinction, but truthfulness is usually considered the primary one. Some approaches recognize more fine-grained scales with specific classes (e.g., tabloid news, mixed reliability news),

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 96–109
July 14, 2023 ©2023 Association for Computational Linguistics

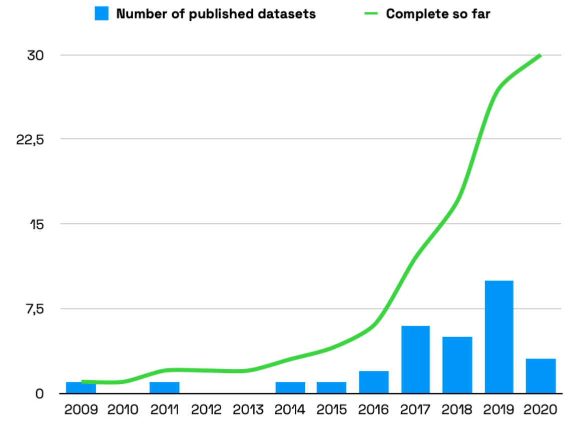


Figure 1: Continual statistics on disinformation classification datasets publishing throughout the years 2009-2020. The bar charts denote the number of new datasets (y-axis) published in the respective year (x-axis), while the overlay line captures the cumulative number of datasets published until that year.

whereas others are binary (including fake and credible classes). Either way, the sole text is considered.

Meanwhile, the task of automatic fact-checking requires a source of truth to which the news article is compared. The task then lies in determining whether the article is supported by facts therein. Hence, one can consider this task a specific variant of stance detection focusing on news media and large-scale ground-truth databases.

We review datasets and approaches in both of these tasks, as our dataset lies somewhere in between.

2.1 Disinformation, Misinformation

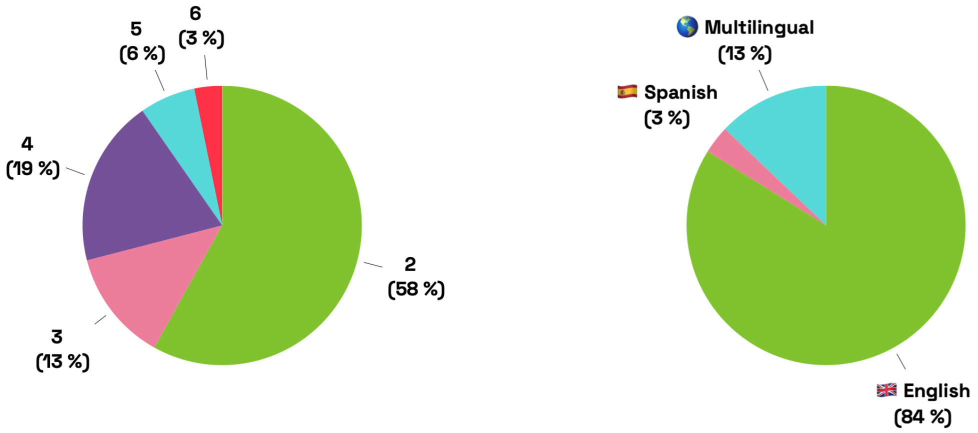
With the advent and development of digital network media at the beginning of the 21st century, there has been a dynamic spread of unverified, inaccurate, or false information (ranging from textual to audiovisual), which is referred to as information disorders. Information disorders as part of information pollution are thus in direct contrast to trustworthy content that is accurate, factually correct, verified, reliable, and up-to-date. According to the media and journalism theorist (Wardle, 2018), it is misleading to label information disorders with the umbrella term “fake news.” Although the definition of fake news is complicated, it is possible to define at least seven criteria that contribute to the contamination of information to such an extent that the use of the term information disorder is appropriate.

Satire/parody, as the least problematic form of information pollution and, therefore, a factor reducing the credibility of news content, is on the one end of the seven-scale spectrum. In contrast, fictional content created to disseminate false information intentionally lies at the other end. Wardle introduces a typology of the three main information disorders based on the seven criteria. The typology is established on the degree of truth/falsity and the intention to cause harm. Erroneous, inaccurate, or untrue content that is not intended to harm recipients because it reflects, for example, ignorance of the disseminator is referred to as misinformation. This term includes satire, parody, or misleading texts, images, or quotes. False or untrue content that is distributed to deceive or manipulate its recipients, whether for financial, ideological, political, social, or psychological reasons, is referred to as disinformation. This term includes malicious lies, fabricated information, disinformation campaigns, etc. Finally, true information disseminated with the intention to cause harm (for example, by revealing a person’s religion, sexual orientation, etc.) is referred to as malinformation.

The conceptual framework of individual information disorders in the professional literature is relatively inconsistent. Thus, part of the scientific community (Fetzer, 2004) considers disinformation “misinformation with an attitude,” while attitude is the aforementioned deliberate deception of recipients. According to another approach (Swire-Thompson et al., 2020; Wang et al., 2019), disinformation is part of misinformation because it is difficult to demonstrate the intention (not) to spread it. In both cases, the notion of misinformation encompasses the term disinformation. However, one can also encounter a more subtle division of individual forms of information disorders (Meel and Vishwakarma, 2020). In addition to the terms disinformation and misinformation, the authors also distinguish autonomous terms such as rumor, conspiracy, hoax, propaganda, opinion spam, false news (i.e., fake news), clickbait, satire, etc. Within the classification of information disorders, we can perceive disinformation and misinformation as overarching concepts because disinformation can take the form of clickbait, rumor, hoax, opinion spam, or conspiracy theory. Similarly, misinformation can be based on rumors or satire.

2.2 Disinformation-related datasets

D’Ulizia et al. (2021) have conducted a thorough study on fake news detection datasets. We high-



(a) Number of classes recognized by the dataset

(b) Language of the dataset

Figure 2: Proportional statistics of the existing datasets for disinformation classification.

light three of these based on the traction within the research community and direct the reader to this review for more detail. Wang (2017) created the LIAR dataset with $12,836$ text excerpts of 6 classes. Later, Nørregaard et al. (2019) published NELA-GT dataset containing 713,000 news articles belonging to 2 classes. Lastly, Slovikovskaya and Attardi (2020) presented the FNC-1 dataset with $49,972$ news articles classified into 4 labels. All these datasets are in English.

Guo et al. (2022) have presented a survey of the current fact-checking datasets. Once again, we mention some of these below and refer the reader to the study for more detail. First, Mitra and Gilbert (2015) created the CredBank dataset with over $1,000$ English Tweets classified into 5 labels. Multiple works followed, including the much larger Suspicious dataset (Volkova et al., 2017) containing over $130,000$ English Tweets with 2 assigned classes. Nakov et al. (2021) presented the CheckThat21-T1A dataset with over $17,000$ Tweets of 2 classes. These Tweets come from multiple languages. Recently, domain-specific datasets have also emerged, prominently centered around COVID-19 facts (Saakyan et al., 2021).

Shown in Figure 1 is a visualization of datasets publishing (D'Ulizia et al., 2021) over time, showing that the popularity of this task in the AI and NLP community is a recent phenomenon, corresponding to the general focus on disinformation in the public discourse. However, the sizeable collective excitement goes hand-in-hand with the inconsistency of the problem's framing and methodologies. This can be demonstrated with Figure 2a, which captures the distribution of these datasets by the pure number of labels they recognize. Furthermore, we see significant inconsistencies in the methodologies leading to these classifications. Some works (Nørregaard et al., 2019) derive the class based on the high-level credibility assessment of its source (i.e., they assess a source and treat all its articles in the same manner, leaving no room for exceptions). Others (Wang, 2017; Slovikovskaya and Attardi, 2020) treat the articles on an individual basis. Alongside, all of these differ in the specific features deducing the classification. Some consider the context of the article and editorial proprieties, while others only use the texts and their attributes.

Moreover, other major problematic characteristics of the dataset population emerge. Despite disinformation being a global threat, the vast majority of these datasets are in English only, as can be seen in Figure 2b. Alarmingly, most of the datasets did not include professionals or academics from the relevant fields, such as the media sciences. We believe that this calls for establishing a robust and uniform methodology for approaching the problem of disinformation holistically and an emphasis on developing datasets for non-English speaking regions with the oversight of relevant experts across domains and industries.

2.3 Automated fake news detection

The task of automated fake news detection has usually been approached by fine-tuning general-purpose language models, such as BERT (Devlin et al., 2019), ELECTRA (Clark et al., 2020), or

RoBERTa (Liu et al., 2019). Specific architectures for this task have been studied in the literature, too. (Reis et al., 2019), for instance, provide additional parameters such as political bias, the domain from the article's originating URL, and prior information about the domain as inputs to their model. (Singhal et al., 2019) create the first multi-modal architecture for this task as they combine the texts at the input with images included in the article. Some recent works also formulate the task as graph classification (Vaibhav et al., 2019).

2.4 Automated fact-checking

Architectures for automated fact-checking usually consist of an evidence retrieval module and a verification module (Thorne et al., 2018). Recent dense retrievers with learned representations and fast dot-product indexing (Lewis et al., 2020; Maillard et al., 2021) have shown strong performance, too. There have also been approaches considering multiple texts with potential evidence for the claims as a single evidence piece by concatenating them (Luken et al., 2018; Nie et al., 2019). Later, an entailment model is employed to determine whether the article's text is supported or refuted by the evidence. We refer the reader to (Guo et al., 2022) for a concise overview of such methods. Recent works also use sequence-to-sequence models to generate natural logic-based inferences later used as proofs (Krishna et al., 2022).

3 Trustworthiness Assessment Methodology

Having familiarized ourselves with the current state of research, we concluded that the best way forward is to build upon the previous work and introduce a new language-agnostic methodology for classifying news articles. The primary motivation for this was the inability of prior approaches to fully reflect the complexity of the problem in terms of media studies and fully appreciate each article uniquely and independently of its source. We hope to provide better data for AI-based tools concerned with handling dubious news articles with this methodology. Below, we introduce the basic framework of our methodology. Its complete overview is available in Appendix A.

3.1 Trustworthiness

To establish a clear division between the fake news detection and fact-checking tasks, our methodology focuses solely on the content aspects of the article. We do not reflect the truthfulness or context of the news, as we believe such practices fall under the latter task. These parameters on their own serve as robust evidence of an article being disinformative (Damstra et al., 2021).

In our framing of the problem, trustworthiness is assessed by the presence of transparent and deceptive attributes. Focusing solely on trustworthiness allows us to streamline the annotation process since there is no requirement for outside context, and the given class is thus final (i.e., unlike with fact-checking methods employing truthfulness, no later information can reverse the assessment).

3.2 Classes

To quantify trustworthiness, we propose 15 negative linguistic attributes of an article (e.g., hate speech, clickbait title, logical fallacies) and 6 positive ones (e.g., real author, references, objective profiling). With these, we define the following classes of trustworthiness:

1. **Trustworthy**: Such an article is entirely credible, cites its sources, and presents the opinions of all involved parties. In our framework, it does not contain any negative attributes while having at least five positive ones.

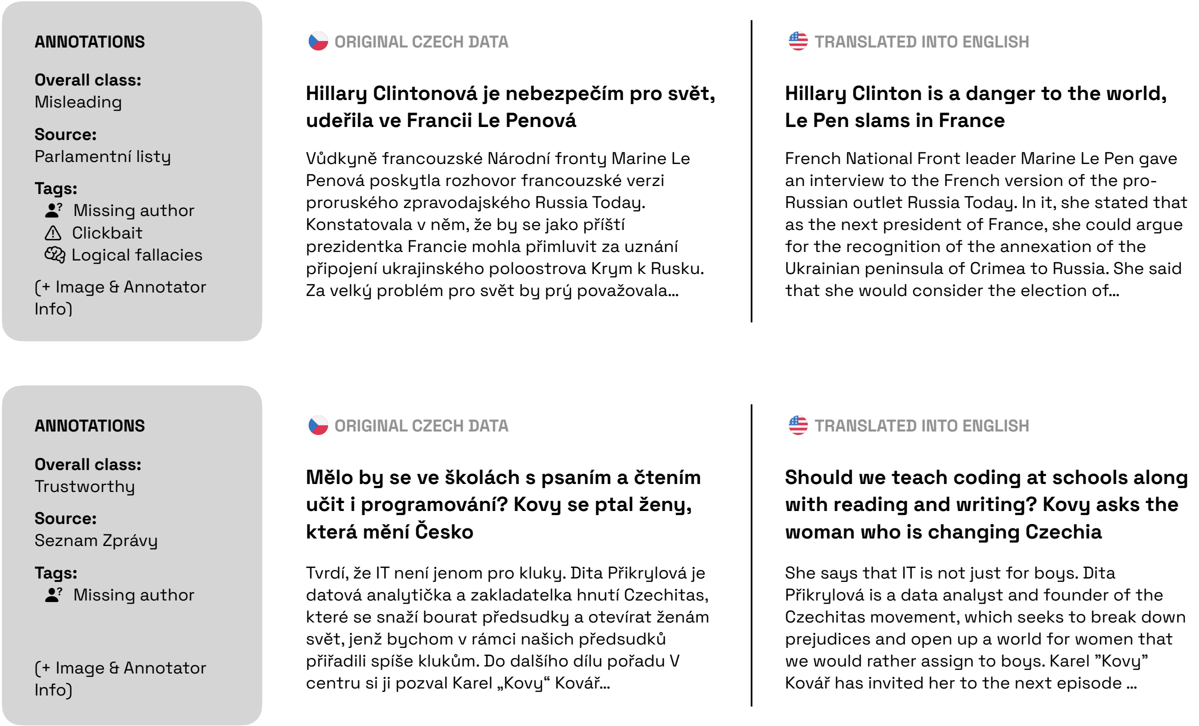
2. **Partially Trustworthy**: While not deceiving its readers, such an article attempts to exaggerate the topic while neglecting to uphold journalistic norms. In our framework, it includes 2 to 5 negative attributes.

3. **Misleading**: Such an article contains deception but does not yet fall under conspiracies. In our framework, it includes 6 to 8 negative attributes.

4. **Manipulative**: Such an article strives to manipulate its reader by employing conspiratory narratives. In our framework, it contains over 8 negative attributes or one of 3 highly problematic ones (e.g., conspiracies, hate speech).

4 Dataset

We collected a dataset of 10, 197 Czech news articles. Each entry in the dataset consists of the article's text, HTML source, title, description, authors, source name, URL, main image, and the



ANNOTATIONS

Overall class:
Misleading

Source:
Parlamentní listy

Tags:
Missing author
Clickbait
Logical fallacies

(+ Image & Annotator Info)

ORIGINAL CZECH DATA

Hillary Clintonová je nebezpečím pro svět, udeřila ve Francii Le Penová

Vůdkyně francouzské Národní fronty Marine Le Penová poskytla rozhovor francouzské verzi proruského zpravodajského Russia Today. Konstatovala v něm, že by se jako příští prezidentka Francie mohla přimluvit za uznání připojení ukrajinského poloostrova Krym k Rusku. Za velký problém pro svět by prý považovala...

TRANSLATED INTO ENGLISH

Hillary Clinton is a danger to the world, Le Pen slams in France

French National Front leader Marine Le Pen gave an interview to the French version of the pro-Russian outlet Russia Today. In it, she stated that as the next president of France, she could argue for the recognition of the annexation of the Ukrainian peninsula of Crimea to Russia. She said that she would consider the election of...

ANNOTATIONS

Overall class:
Trustworthy

Source:
Seznam Zprávy

Tags:
Missing author

(+ Image & Annotator Info)

ORIGINAL CZECH DATA

Mělo by se ve školách s psaním a čtením učit i programování? Kovy se ptal ženy, která mění Česko

Tvrdí, že IT není jenom pro kluky. Dita Přikrylová je datová analytička a zakladatelka hnutí Czechitas, které se snaží bourat předsudky a otevírat ženám svět, jenž bychom v rámci našich předsudků přiřadili spíše klukům. Do dalšího dílu pořadu V centru si ji pozval Karel „Kovy“ Kovář...

TRANSLATED INTO ENGLISH

Should we teach coding at schools along with reading and writing? Kovy asks the woman who is changing Czechia

She says that IT is not just for boys. Dita Přikrylová is a data analyst and founder of the Czechitas movement, which seeks to break down prejudices and open up a world for women that we would rather assign to boys. Karel "Kovy" Kovář has invited her to the next episode ...

Figure 3: Representative examples of two article items in the dataset including their annotations and metadata. The original texts are translated into English for demonstrative purposes.

annotated class. A subset of the dataset also contains the linguistic attribute annotations, which led up to the classification. Representative examples of two article items are shown in Figure 3.

We open-source the dataset at `https://www.verifee.ai/research/` under a custom license[1]. We provide pre-defined train (80 %), validation (10 %), and testing splits (10 %) that have been assigned randomly.

4.1 Scraping and Pre-processing

Initially, we assembled nearly $94,000$ articles by scraping URLs of 60 Czech news sources[2] obtained from Common Crawl[3]. These sources included mainstream journalistic websites, tabloids, independent news outlets, and websites that are part of the disinformation ecosystem (Štětka et al., 2021), capturing the full scope of journalistic content in the Czech Republic. Their complete list can be found in Appendix C.

4.1.1 Enrichment

Next, we determined the category (opinion, interview, general) and the topic (general, sport, economics, hobby, tabloid) of each article through pattern matching. Similarly, we detected mentions of any controversial topics relevant to the Czech media context. Additionally, we ascertained whether the article disposes of a real author via an out-of-the-box Named Entity Recognition model (Sido et al., 2021) for the Czech language.

4.1.2 Filtering

We applied multiple filters and balancing mechanisms based on text length and topics to mitigate deficiencies caused by inherent flaws in Common Crawl, which reduced the dataset's size from $94,000$ to $10,197$ items. This way, we also ensured that the data is as representative of the Czech news ecosystem and as diverse as possible. The detailed parameters used for filtering are described in Appendix B.

4.2 Annotations Organization

We conducted two rounds of annotation. The first round involved $7,528$ articles, where just the class was assigned to each article. The second round included $2,669$ articles. This time, annotators were

---

[1] Our license — building on top of Creative Commons BY-NC-SA (`https://creativecommons.org/licenses/by-nc-sa/2.0/`) — is available at `https://www.verifee.ai/files/license.pdf`.

[2] The complete list of sources can be found in Appendix C.

[3] `https://commoncrawl.org`

Class	Number of articles
Trustworthy	3520
Partially trustworthy	2574
Misleading	1524
Manipulative	1848
Unclassifiable	731

Table 1: Distribution of article items per annotated trustworthiness class.

asked to provide both the class and flag any problematic attributes of each article defined in our methodology. This enabled us to examine the importance of the various metrics in the methodology. Every annotator was assigned 40 articles per round.

4.2.1 Annotators

All the raters were students of journalism who were native speakers of the Czech language. They thus had a more advanced understanding of news credibility than the general population. Due to their age (Peltzman, 2019) and education (Scott, 2022), their possible bias toward more progressive/liberal schools of thought may have influenced the rating of topics in corresponding areas. We briefed all the annotators on an extensive seminar, provided them with detailed materials, and encouraged them to come forward with any problems.

To further mitigate the influence of annotators' media and author preferences on the assessment, we masked any elements in the article that would enable the annotators to identify the source or author of the text. Specifically, we replaced their mentions with placeholders.

4.2.2 Platform

We used a tailored version of the open-source Doccano[4] tool. Inside the application, annotators were presented with one article at a time in its HTML form with all images included. The platform allowed the user to add necessary tags and comments to each piece.

The platform enabled us to track the annotators' activity, including the time spent on each article. In the second wave of annotation, 10 % of articles were shared among all annotators to evaluate the inter-annotator agreement. These were preselected and equipped with our ground-truth annotations.

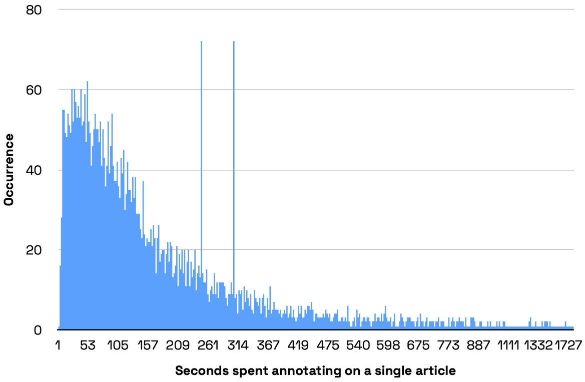


Figure 4: Distribution of single article annotation time. The x-axis denotes the number of seconds, and the y-axis the count of respective occurrences.

4.3 Data Analysis

By average, annotators spent 2.97 minutes (179 seconds) on a single article, which indicates reasonable time allocation. The distribution of the per-item time allocation is shown in Figure 4.

The overall class distribution, reported in Table 1, shows a skew towards the trustworthy end of the spectrum. We pay close attention to the per-source class distributions and ensure that the general tendencies in annotations match the Czech media space analyses studying the high-level credibility of news outlets. State-owned media (ČTK, ČT24, and iROZHLAS) and local newspapers (Jihlavské listy and Mostecké listy) have a majority of their stories classified as 'Trustworthy.' Articles from private media outlets (Seznam Zprávy, iDnes, Deník) are also most often classified as 'Trustworthy.' This time, however, other classes are more prominent. Openly left-wing (A2larm) or right-wing (Echo 24 and Forum24) sources have more items classified as misleading or manipulative in comparison to their counterparts without distinctive political tendencies. The 'Partially trustworthy' news stories occur the most in tabloid news sites (Blesk, Aha!, Extra.cz).

We can see the disinformative news sites (Aeronet, Protiproud, Skrytá pravda) on the other side of the spectrum, as their articles get exceedingly labeled as 'misleading' and 'manipulative.'

Overall, we can see that the high-level patterns in the annotations match the news sources' characteristics, as described in media science literature (Štětka et al., 2021).

---

[4] `https://doccano.github.io/doccano/`

Model	**Trustworthy**	**Partially trustworthy**	**Misleading**	**Manipulative**	**Overall**
RoBERTa	0.63	0.44	**0.45**	0.59	**0.53**
BERT	0.55	**0.47**	0.44	0.61	0.52
Electra	**0.66**	0.43	0.39	0.58	0.51
TF-IDF	0.52	0.40	0.35	**0.68**	0.49
FastText	0.58	0.28	0.14	0.60	0.40

Table 2: Dataset benchmarks fine-tuned for the task of trustworthiness classification. We report the testing split F-1 score for each class, as well as an overall average.

4.3.1 Inter-annotator Agreement

The Randolph's Kappa (Randolph, 2010) of the second annotation wave amounts to 0.615, corresponding to a moderate agreement (McHugh, 2012). Atop this innate solid baseline, we filtered problematic annotators, who were identified by largely deviating in categorizing these duplicated articles.

5 Experimental Results

We conduct baseline results by fine-tuning general-purpose architectures.

5.1 Data Preparation

We follow the pre-defined configuration of train, test, and validation splits described in Section 4. To balance the training, we selected a random sample of 1400 articles from each credibility class. We insert the article's title and body (concatenated with a period) as the input to the model.

5.2 Model architectures

We experimented with 5 model architectures: RoBERTa-based RobeCzech (Liu et al., 2019; Straka et al., 2021), BERT-based Czert (Devlin et al., 2019; Sido et al., 2021), Electra-based Small-e-Czech (Clark et al., 2020; Kocián et al., 2022), Term frequency-inverse document frequency (TF-IDF)-based Support Vector Machines (SVM) classifier (Sammut and Webb, 2010b; Hearst et al., 1998), and FastText Embedding-based Logistic Regression classifier (Joulin et al., 2017). Respective training details follow.

5.2.1 RobeCzech (RoBERTa)

RobeCzech is a RoBERTa model trained on the Czech national corpus (Křen et al., 2016). Setting the learning rate to $3 * 10^-5$ and the batch size to 16, we fine-tuned this model using cross-entropy loss over a span of 4 epochs.

5.2.2 Czert (BERT)

Czert is a BERT model trained on the Czech national corpus. Setting the learning rate to $3 * 10^-5$ and the batch size to 32, we fine-tuned this model using cross-entropy loss over a span of 4 epochs.

5.2.3 Small-e-Czech (Electra)

Small-e-Czech (Kocián et al., 2022) is an ELECTRA-small trained on an internal Czech web corpus of Seznam.cz. Setting the learning rate to $3 * 10^-4$ and the batch size to 64, we fine-tuned this model using cross-entropy loss over a span of 3 epochs.

5.2.4 TF-IDF SVM

Our TF-IDF SVM model pipeline vectorizes the text using TF-IDF and later classifies these representation using SVM. We used the scikit-learn library (Pedregosa et al., 2011) and kept the model's vocabulary unfiltered by setting its `min_df` and `max_df` parameters to 1. For the SVM, we used Radial basis function kernel and Regularization parameter set to 1.

5.2.5 FastText

Our FastText pipeline vectorizes the tokenized words found by nltk (Bird and Loper, 2004) using FastText (Grave et al., 2018). By averaging these, a single article vector is obtained for each article, and a one-vs-rest logistic regression (Sammut and Webb, 2010a) is used to yield the predicted class. We used L2 penalty term combined with regularization set to 1.

5.3 Results

We present per-class F-1 score results, as well as their average, in Table 2. As can be observed, the scores differ distinctly across classes. Upon closer inspection, both TF-IDF SVM and FastText models perform better on the at the pole classes of the trustworthiness spectrum (i.e., 'Trustworthy' and 'Manipulative'), but under-perform at the middle ones, resulting in overall testing F-1 scores of 0.49 and 0.40, respectively. We hypothesize that the poor performance of the FastText model is caused due to the inability to capture apt representation of causality and argumentative approaches in the averaged semantic vectors.

The comparative results of the Transformer-based models follow the same order as in other Czech evaluation tasks. We see the RobeCzech (ROBERTa) model come on top with F-1 score of 0.53, followed by Czert (BERT) with F-1 score of 0.52 and lastly Small-e-Czech (ELECTRA) with F-1 score of 0.51.

6 Ethical Discussion and Limitations

Due to the high-impact nature of the solved task, we review the ethical considerations made during this research project. Additionally, we outline further steps to ensure safety and transparency beyond publication, as well as recommendations for build-up work.

First, let us focus on the presence of biases in the data. We put extensive procedures in place even at the very start of the project. By inviting media researchers into our core team, we wanted to minimize misunderstandings and mistakes that scientists from the field of computational linguistics could easily make when assembling the methodology for the task of trustworthiness assessment due to their limited knowledge of the current literature and theory in the area of journalism. Prior to the data annotation, we invited scholars in media studies and journalists from the industry to a series of workshops, where we asked them to submit feedback and discuss the methodology. Based on the assembled comments, we kept updating it until a general consensus was reached. In terms of the annotation process itself, multiple safeguards have been employed to prevent annotators' bias towards specific sources or authors (that may affect the classification).

Second, let us shift towards the ethics of using any technology built around this data in the wild. We want to stress that anyone using this dataset for the purposes of creating a trustworthiness classification system should provide transparent information to the users that this process is automatic and hence faulty to a certain extent. We must note that it still needs to be determined how models trained on this data generalize for future articles (i.e., news about topics and events they have not encountered in the training set) and news sources not included in the training set. A study into these should be conducted prior to making this technology available unrestrictedly to the public.

Despite bearing these safety questions in mind is crucial, such systems can eventually be great assistive tools for people reading news stories online. The potential benefits of such technology should support initiatives to safeguard it first and establish public and academic trust.

7 Conclusion

This work presents a novel methodology for classifying news article trustworthiness and presents a multimodal dataset of $10,197$ Czech news articles with respective annotations. Unlike previous methods that classify all texts from a given media outlet with the same class, we treat the articles on an individual level. The high inter-annotator agreement shows that our methodology constitutes a good feature-based framework, leaving little to no room for personal annotators' inducement.

To the best of our knowledge, we are the first to include media and computer science researchers in the core team when developing a similar dataset. Additionally, all of our annotators were journalism students. As our methodology underwent extensive feedback loops with professionals in the industry, we hope to establish a new interdisciplinary standard for future related works to follow.

We provide benchmark results on our dataset using 5 different classifier architectures and obtain promising results – the best-performing RoBERTa model achieves a testing F-1 score of 0.53. We open-source the complete dataset and encourage researchers to undertake similar initiatives in new languages and social contexts, especially low-resourced ones. Since the framework derives all parameters based on the text contents, it is language-agnostic. Hence, minimal additional methodological work is necessary before new annotations.

In future work, we intend to study the generalization abilities of systems trained using this data and the application of task-specific architectures. Moreover, we wish to further explore the potential of multimodality that our dataset offers and analyze the attached images.

Acknowledgements

This paper was supported by the Technology Agency of the Czech Republic under grant No. TL05000057 "The Signal and the Noise in the Era of Journalism 5.0 - A Comparative Perspective of Journalistic Genres of Automated Content".

References

SG Bird and Edward Loper. 2004. Nltk: the natural language toolkit. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

Alyt Damstra, Hajo G. Boomgaarden, Elena Broda, Elina Lindgren, Jesper Strömbäck, Yariv Tsfati, and Rens Vliegenthart. 2021. What does fake look like? a review of the literature on intentional deception in the news and on social media. *Journalism Studies*, 22(14):1947–1963.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Arianna D'Ulizia, Maria Chiara Caschera, Fernando Ferri, and Patrizia Grifoni. 2021. Fake news detection: a survey of evaluation datasets. *PeerJ Computer Science*, 7:e518.

James H Fetzer. 2004. Disinformation: The use of false information. *Minds and Machines*, 14(2):231–240.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10:178–206.

Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. 1998. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28.

Armand Joulin, Edouard Grave, and Piotr Bojanowski Tomas Mikolov. 2017. Bag of tricks for efficient text classification. *EACL 2017*, page 427.

Matěj Kocián, Jakub Náplava, Daniel Štancl, and Vladimír Kadlec. 2022. Siamese bert-based model for web search relevance ranking evaluated on a new czech dataset. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(11):12369–12377.

Michal Křen, Václav Cvrček, Tomáš Čapka, Anna Čermáková, Milena Hnátková, Lucie Chlumská, Tomáš Jelínek, Dominika Kovaříková, Vladimír Petkevič, Pavel Procházka, Hana Skoumalová, Michal Škrabal, Petr Truneček, Pavel Vondřička, and Adrian Zasina. 2016. SYN v4: large corpus of written czech. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Amrith Krishna, Sebastian Riedel, and Andreas Vlachos. 2022. ProoFVer: Natural logic theorem proving for fact verification. *Transactions of the Association for Computational Linguistics*, 10:1013–1030.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Jackson Luken, Nanjiang Jiang, and Marie-Catherine de Marneffe. 2018. Qed: A fact verification system for the fever shared task. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER)*, pages 156–160.

Jean Maillard, Vladimir Karpukhin, Fabio Petroni, Wen-tau Yih, Barlas Oguz, Veselin Stoyanov, and Gargi Ghosh. 2021. Multi-task retrieval for knowledge-intensive tasks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1098–1111.

Mary L McHugh. 2012. Interrater reliability: the kappa statistic. *Biochemia medica*, 22(3):276–282.

Priyanka Meel and Dinesh Kumar Vishwakarma. 2020. Fake news, rumor, information pollution in social media and web: A contemporary survey of state-of-the-arts, challenges and opportunities. *Expert Systems with Applications*, 153:112986.

Tanushree Mitra and Eric Gilbert. 2015. Credbank: A large-scale social media corpus with associated credibility annotations. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 9, pages 258–267.

Preslav Nakov, Giovanni Da San Martino, Tamer Elsayed, Alberto Barrón-Cedeno, Rubén Míguez, Shaden Shaar, Firoj Alam, Fatima Haouari, Maram Hasanain, Nikolay Babulkov, et al. 2021. The clef-2021 checkthat! lab on detecting check-worthy claims, previously fact-checked claims, and fake news. In *European Conference on Information Retrieval*, pages 639–649. Springer.

Yixin Nie, Haonan Chen, and Mohit Bansal. 2019. Combining fact extraction and verification with neural semantic matching networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6859–6866.

Jeppe Nørregaard, Benjamin D Horne, and Sibel Adalı. 2019. Nela-gt-2018: A large multi-labelled news dataset for the study of misinformation in news articles.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Sam Peltzman. 2019. Political ideology over the life course. *Public Choice: Analysis of Collective Decision-Making eJournal*.

Justus Randolph. 2010. Free-marginal multirater kappa (multirater κfree): An alternative to fleiss fixed-marginal multirater kappa. volume 4.

Julio CS Reis, André Correia, Fabrício Murai, Adriano Veloso, and Fabrício Benevenuto. 2019. Supervised learning for fake news detection. *IEEE Intelligent Systems*, 34(2):76–81.

Arkadiy Saakyan, Tuhin Chakrabarty, and Smaranda Muresan. 2021. COVID-fact: Fact extraction and verification of real-world claims on COVID-19 pandemic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2116–2129, Online. Association for Computational Linguistics.

Claude Sammut and Geoffrey I. Webb, editors. 2010a. *Logistic Regression*, pages 631–631. Springer US, Boston, MA.

Claude Sammut and Geoffrey I. Webb, editors. 2010b. *TF–IDF*, pages 986–987. Springer US, Boston, MA.

Ralph Scott. 2022. Does university make you more liberal? estimating the within-individual effects of higher education on political values. *Electoral Studies*, 77:102471.

Jakub Sido, Ondřej Pražák, Pavel Priban, Jan Pašek, Michal Seják, and Miloslav Konopík. 2021. Czert – czech bert-like model for language representation.

Shivangi Singhal, Rajiv Ratn Shah, Tanmoy Chakraborty, Ponnurangam Kumaraguru, and Shin'ichi Satoh. 2019. Spotfake: A multi-modal framework for fake news detection. In *2019 IEEE fifth international conference on multimedia big data (BigMM)*, pages 39–47. IEEE.

Valeriya Slovikovskaya and Giuseppe Attardi. 2020. Transfer learning from transformers to fake news challenge stance detection (fnc-1) task. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1211–1218.

Milan Straka, Jakub Náplava, Jana Straková, and David Samuel. 2021. Robeczech: Czech roberta, a monolingual contextualized language representation model. In *Text, Speech, and Dialogue: 24th International Conference, TSD 2021, Olomouc, Czech Republic, September 6–9, 2021, Proceedings 24*, pages 197–209. Springer.

Briony Swire-Thompson, Joseph DeGutis, and David Lazer. 2020. Searching for the backfire effect: Measurement and design considerations. *Journal of Applied Research in Memory and Cognition*, 9(3):286–299.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018. The fact extraction and verification (fever) shared task. *EMNLP 2018*, 80(29,775):1.

Vaibhav Vaibhav, Raghuram Mandyam, and Eduard Hovy. 2019. Do sentence interactions matter? leveraging sentence level representations for fake news classification. In *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*, pages 134–139, Hong Kong. Association for Computational Linguistics.

Svitlana Volkova, Kyle Shaffer, Jin Yea Jang, and Nathan Hodas. 2017. Separating facts from fiction: Linguistic models to classify suspicious and trusted news posts on twitter. In *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 2: Short papers)*, pages 647–653.

William Yang Wang. 2017. "liar, liar pants on fire": A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 422–426.

Yuxi Wang, Martin McKee, Aleksandra Torbica, and David Stuckler. 2019. Systematic literature review on the spread of health-related misinformation on social media. *Social science & medicine*, 240:112552.

Claire Wardle. 2018. The need for smarter definitions and practical, timely empirical research on information disorder. *Digital Journalism*, 6(8):951–963.

A Woodward. 2020. Fake news": A guide to trump's favourite phrase and the dangers it obscures. *Independent. Retrieved from https://www. independent. co. uk/news/world/americas/us-election/trump-fake-news-counter-history-b732873. html.*

Václav Štětka, Jaromír Mazák, and Lenka Vochocová. 2021. "nobody tells us what to write about": The disinformation media ecosystem and its consumers in the czech republic. *Javnost - The Public*, 28(1):90–109.

A Annotation Methodology and Annotators Instructions

A.1 Annotation instructions

Each class is defined by the positive aspects it contains and the negative aspects it can and cannot contain. When annotating, we start with the most trustworthy class (credible). We then move down a class whenever an article does not meet the requirements of the current class, for example when it contains too many permissible negative aspects or contains a negative aspect that must not occur in that class.

A.2 Trustworthiness classes

A.2.1 Trustworthy

Positive aspects contained in the article (min. 5):

- Citation of relevant authorities on the topic, representing credible institutions
- Views of all interested parties
- Facts presented within the context
- Grammatical correctness, without overtly expressive language
- An identifiable author
- Undistorted data

Negative aspects contained in the article (max. 1):

- Missing citations
- Unrepresented views of opposing parties
- Facts presented without a context
- Grammatically incorrect or overtly expressive language
- Unidentifiable author
- Distorted data

Negative aspects that must not appear in the article:

- Clickbait
- Hate speech
- An attack on an opinion opponent without justification
- Manipulating the reader
- Conspiracy theories
- Appeal to emotion
- Logical fallacies
- Tabloid language

A.2.2 Partially Trustworthy

Positive aspects contained in the article:

- Grammatical correctness, without overtly expressive language
- Undistorted data

Negative aspects contained in the article (2-5):

- Missing citations
- Unrepresented views of opposing parties
- Facts presented without a context
- Grammatically incorrect or overtly expressive language
- Unidentifiable author
- Distorted data
- Clickbait
- Appeal to emotion
- Tabloid language

Negative aspects that must not appear in the article:

- Hate speech
- An attack on an opinion opponent without justification
- Manipulating the reader
- Conspiracy theories
- Logical fallacies

A.2.3 Misleading

Positive aspects contained in the article:

None need to be present

Negative aspects contained in the article (6-7):

- Missing citations
- Unrepresented views of opposing parties
- Facts presented without a context
- Grammatically incorrect or overtly expressive language
- Unidentifiable author
- Distorted data
- Clickbait
- Appeal to emotion
- Tabloid language
- Logical fallacies
- An attack on an opinion opponent without justification

Negative aspects that must not appear in the article:

- Hate speech
- Manipulating the reader
- Conspiracy theories

A.2.4 Manipulative

Positive aspects contained in the article:

None need to be present

Negative aspects contained in the article:

It either contains 8+ negative aspects:

- Missing citations
- Unrepresented views of opposing parties
- Facts presented without a context
- Grammatically incorrect or overtly expressive language
- Unidentifiable author
- Distorted data
- Clickbait
- Appeal to emotion
- Tabloid language
- Logical fallacies
- An attack on an opinion opponent without justification

Or it contains any of these 3 aspects:

- Hate speech
- Manipulating the reader
- Conspiracy theories

Negative aspects that must not appear in the article:

All negative aspects can be present

A.3 Resolving unclassifiable articles and errors

A.3.1 Unclassifiable articles

Articles that, due to their length or structure, cannot be classified according to this methodology (or do not have sufficient content to allow the aspects mentioned to be analysed) must be labeled as unclassifiable. This may include one-sentence flash news announcements, paywall texts and others. This allows them to be filtered out and not corrupt the rest of the annotated data.

A.3.2 Errors

In the case that an error with the platform or an uncertainty with an article is encounter, we fully encourage annotators to report those issues through comment functionality on the Doccano platform. Our team will do their best to resolve any problem and clarify any ambiguity.

B Filtering Procedure

Following filters and balancing mechanisms were employed to select the dataset articles out of the greater pool of scraped articles.

- **Length of the text**: Only articles with a length of 400 to $10,000$ characters were included.
- **Category**: We filtered out opinion pieces for mainstream media. However, we kept these for alternative news sources, as the line between reporting and conveying opinion is often blurred here. Interviews were excluded in both cases.

- **Source**: We selected articles so that all sources are as balanced as possible, no matter their actual distribution in the media ecosystem.
- **Topic**: Articles concerning hobbies and sports each form only 5% of the dataset, as they typically are not connected to disinformation. The remaining topics (general, economic, and tabloid) each form 30 % of the dataset.
- **Controversial topics**: We balanced the coverage of controversial topics by including the same number of such articles from mainstream and alternative or extremely opinionated news sources.

C Detailed news source statistics

Continued on the next page.

News source	Article items per class				
	Trustworthy	**Part. trust.**	**Misleading**	**Manipulative**	**Unclassifiable**
A2larm	145	10	43	100	20
AC24	22	24	42	39	18
Aeronet	6	321	65	19	9
Aha!	20	7	41	73	16
Aktuálně	194	4	29	94	34
Bez politické korektnosti	1	0	2	1	0
Blesk	38	5	33	132	6
Brněnský deník	27	0	2	10	4
CNN Prima News	187	2	13	71	11
CZ24 News	16	21	12	28	2
Czech free press	3	10	12	15	2
Deník	58	0	4	14	2
Deník N	28	1	4	8	9
Deník Referendum	173	6	18	45	5
E-republika	4	16	13	7	2
E15	3	0	0	1	1
Echo 24	188	1	19	75	6
Euro	12	0	1	8	0
Euro Zprávy	52	0	3	18	5
Extra.cz	71	29	100	205	25
Forum24	140	21	35	60	12
Globe 24	15	0	2	7	0
Haló noviny	14	4	16	14	3
Hospodářské noviny	34	4	4	14	77
INFO.cz	16	2	3	18	13
Jihlavské listy	26	0	1	3	3
Lidovky.cz	5	3	4	19	30
MediaGuru	21	0	3	14	1
Metro	125	0	5	48	9
Mostecké listy	22	0	1	3	1
NWOO	8	63	37	35	15
Neviditelný pes	2	3	1	0	2
Novinky.cz	65	1	12	64	15
Nová republika	5	54	51	31	7
Outsider Media	94	234	162	118	91
Parlamentní Listy	255	80	119	222	32
Peak.cz	103	1	12	48	6
Proti Proud	13	273	97	39	22
Raptor TV	1	3	2	4	1
Reflex	1	1	3	1	11
Respekt	2	0	1	3	0
Rukojmí	18	242	101	43	12
Seznam Zprávy	164	1	12	45	8
Skrytá Pravda	6	162	61	17	10
Sputnik Česká republika	199	39	89	264	32
Stars 24	27	3	15	38	2
Svobodné noviny	13	68	44	22	8
Svobodný svět	0	1	3	0	0
TN.cz	201	3	37	200	17
Týden	54	0	4	14	4
Týdeník občanské právo	0	1	0	0	0
VOX Populi	3	69	54	13	21
Zvědavec	6	7	6	6	5
iDnes.cz	90	1	10	39	19
iROZHLAS	219	1	11	57	18
ČT24	226	3	4	36	35
ČTK	32	0	2	0	1
Časopis Šifra	11	8	10	10	3
Česko Aktuálně	36	35	34	42	8

Table C.1: Class distribution of all unique news sources found in the dataset.

Towards Detecting Harmful Agendas in News Articles

Warning: This paper contains examples of online text containing strong views and offensive language.

Melanie Subbiah[*]
Columbia University
m.subbiah@columbia.edu

Amrita Bhattacharjee[*]
Arizona State University
abhatt43@asu.edu

Yilun Hua
Columbia University
yh3228@columbia.edu

Tharindu Kumarage
Arizona State University
kskumara@asu.edu

Huan Liu
Arizona State University
huanliu@asu.edu

Kathleen McKeown
Columbia University
kathy@cs.columbia.edu

Abstract

Manipulated news online is a growing problem which necessitates the use of automated systems to curtail its spread. We argue that while misinformation and disinformation detection have been studied, there has been a lack of investment in the important open challenge of detecting harmful agendas in news articles; identifying harmful agendas is critical to flag news campaigns with the greatest potential for real world harm. Moreover, due to real concerns around censorship, harmful agenda detectors must be interpretable to be effective. In this work, we propose this new task and release a dataset, NEWSAGENDAS, of annotated news articles for agenda identification. We show how interpretable systems can be effective on this task and demonstrate that they can perform comparably to black-box models.

1 Introduction

In recent years, the spread of misinformation and disinformation has become a particularly persistent and harmful issue online (Bastick, 2021; Mueller III, 2020; Vosoughi et al., 2018; Zhang and Ghorbani, 2020). For example, during the COVID-19 pandemic in the United States, we saw several instances of malicious actors propagating disinformation regarding mask mandates, vaccines, and fake remedies and cures to discredit the government and public health officials. The people initiating these disinformation campaigns typically have some harmful agenda, such as discrediting an individual/group or encouraging disruptive real-world action. Furthermore, with new conversational language models such as ChatGPT and GPT-4 (OpenAI, 2023), a malicious actor can generate human-like harmful text content at scale.

Title: Are You Prepared For The War To End All Wars?

... Everyone in the know (global elitists) knows what is happening, just not exactly when they can pull it all together. And the media awaits this war with baited breath as they count down to the dramatic moment when they can report the incident that will compel the innocent to attack the guilty. Anyone with half a brain can see the greatly increased anti-Russian propaganda of the past few weeks. This has happened as the Russia-gate claims have fallen to pieces, as former CIA analyst Raymond McGovern, the late Robert Parry, Paul Craig Roberts, and others have documented so assiduously. All across the media spectrum, from the big name corporate stenographers like The New York Times, CNN, National Public Radio, The Washington Post to The Atlantic and Nation magazines and other leftist publications such as Mother Jones and Who What Why, the Russia and Putin bashing has become hysterical in tone ...

Table 1: Example article with annotated spans from our dataset, original article from *infiniteunknown.net*, a source with label *conspiracy* in the FakeNewsCorpus. Orange spans are annotated as *conspiracy*, yellow spans are *political bias*, and blue spans are *propaganda*.

Identifying these types of harmful news campaigns typically requires consideration of three important attributes:

1. **Factuality** - Does the article rely on false information?
2. **Authorial Deception** - Did the author knowingly deceive the reader?
3. **Agenda** - Why did the author deceive the reader?

Misinformation in news is any article which relies on false information and can therefore be identified by focusing on *factuality*. Disinformation is deliberately misleading information created/disseminated with an intent to deceive (Shu et al., 2020a), so can be identified by *factuality* and *authorial deception*. However, the degree of harm caused by disinformation and misinformation depends on the *agenda* (or goal) of the article.

---

[*] These authors contributed equally to this work.

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 110–128
July 14, 2023 ©2023 Association for Computational Linguistics

Fallis (2015) advocates for this kind of focus on agenda as a useful marker of intentionality in disinformation detection. Defining what constitutes a harmful agenda is an inherently subjective task and requires a notion of good and bad. Researchers in different domains have tried defining and formalizing the concept of harm, such as harmful online content (Scheuerman et al., 2021), COVID-19 related tweets (Alam et al., 2021), etc. However, to the best of our knowledge, the notion of harmful agendas in journalistic news articles has not been explored yet. In this paper, we therefore propose a new task of detecting harmful agendas in news articles. Inspired by definitions of harm in other works, we specifically focus on real-world harm, meaning articles that spur core belief change or actions that significantly harm someone.

To develop an initial detector, we formulate this task as classifying an article's agenda as *harmful* or *benign*, based solely on the article text, and we annotate a dataset, NEWSAGENDAS, to evaluate performance. We note that future work could also formulate this problem in several other ways, such as also identifying the target audience, or additionally using metadata or contextual cues such as author information, publication platform, etc.

We imagine this type of agenda detector could be used to flag potentially harmful articles for further inspection. It is therefore critical that any such detector be interpretable so that further examination could quickly reveal why an article was flagged and screen out any falsely identified articles. For sensitive application areas, there is a need to build models that are interpretable by design, rather than trying to interpret their results after the fact (Rudin, 2019). Given the subjectivity and the sensitivity of this task, we build an interpretable model that uses extractive rationale-based feature detectors to ensure faithfulness and interpretability, not only at the feature level but also at the text level.

Our primary contributions are:

1. Defining the important open challenge of detecting harmful agendas in news articles.
2. Annotating and releasing NEWSAGENDAS - 506 news articles, encompassing 882 fine-grained label annotations for this task.[1]
3. Developing a harmful agenda detector which jointly prioritizes interpretability and performance.

2 NEWSAGENDAS Dataset

In order to evaluate our model's performance and contribute an initial benchmark for this task, we annotated news articles which we are releasing as a novel dataset, NEWSAGENDAS.

2.1 Features of Interest

To promote interpretability, we hypothesize based on consultation with journalism professors at Arizona State University that the features shown in Table 2 (e.g., hate speech, propaganda, etc.) may have a significant relationship to the overall classification of article agenda in the sociopolitical context of the United States (see Table 2 for justification).

We are therefore interested in annotating these feature labels at the article-level as well as the overall agenda classification for the article. Using these features also allows us to build on the training datasets used in fine-grained news classification to classify news into these different categories.

2.2 Articles

We use articles from the FakeNewsCorpus[2] along with satire and real news articles from the Yang et al. (2017) dataset and propaganda articles from the Proppy corpus (Barrón-Cedeno et al., 2019) to cover a range of articles that should contain the features and agendas we are interested in. The FakeNewsCorpus contains articles in English from a web scrape of sources which frequently post misinformation. Each source has one or more specific labels indicating the general type of content it publishes and many of these labels match our features of interest (e.g., junk science, conspiracy theories, etc.). Since these labels are assigned at the source level, they serve as weak labels at the article level. We sample 600 articles for annotation, sampling to match the distribution of weak labels in the FakeNewsCorpus (based on the articles' primary weak labels; see Appendix E for more detail).

2.3 Annotation Method

We hired Columbia University students who study journalism, political science, or natural language processing and thus have experience interpreting news (see Appendix B for hiring details).

We presented each annotator with the title of the article and the first 1,700 characters of the article truncated to the last sentence. They were asked to assume the article contained some false claims, and

---

[1] All data and code is available at https://github.com/melaniesubbiah/harmfulagendasnews.

[2] https://github.com/several27/FakeNewsCorpus

Label	Definition	Notes on Connection to Article Agenda
Clickbait	An exaggeration or twisting of information to shock and grab the attention of the reader.	Can be used to promote a harmful agenda (Carnahan et al.; Chen et al.), but often just a marketing strategy which is relatively benign.
Junk Science	Untested or unproven theories presented as scientific fact.	Can be unintentional, but has a high potential for harm, particularly in the medical domain (Pandey; Poynter).
Hate Speech	Language that promotes or justifies hatred, violence, discrimination, or negative prejudice against a person or category of people.	Involves extreme language that indicates clear intent on the part of the author and has a high potential for harm, even physical violence (Haynes).
Conspiracy Theory	A belief that some covert but influential organization is responsible for a circumstance or event.	Erodes public trust in science, institutions, and government (Ahmed et al.; Oliver and Wood) which may not be intentional on the part of individual actors but is harmful.
Propaganda	Promoting or publicizing a particular political cause or perspective.	Polarizes readers and harms the democratic environment necessary for healthy political debate (Guarino et al.).
Satire	Using humor, irony, or exaggeration to critique something or to amuse.	Not typically harmful when used to reveal a social/political truth, rather than for hate (Levi et al.; Golbeck et al.).
Negative Sentiment	Evokes a negative emotional response in the reader.	Evoking negative emotionality can create a lasting reaction (Mastrine), which can be more benign like sensationalism (Ward), or more harmful like negative propaganda.
Neutral Sentiment	Generally neutral/factual tone throughout the article. Does not evoke strong emotion.	Credible news organizations often have guidelines for objective and neutral reporting of 'hard-news' (Rogers).
Positive Sentiment	Evokes a positive emotional response in the reader.	Research suggests positive sentiment is not often used in disinformation or to instigate/polarize readers (Alonso et al.).
Political Bias	Angling information toward a particular political cause or perspective.	Biased articles may misrepresent/slant facts to support (harmful) agendas in cases of contentious topics (Chen et al.).
Call to Action	Urging the reader to do (or not do) something in order to further some goal.	Instigating or urging the reader to take some action for example via bandwagoning (Da San Martino et al.) may result in a (harmful) real-world effect.

Table 2: The definitions for the full set of labels annotators were asked to label articles with.

then rate whether it advanced a harmful agenda on a scale of 1 to 5. We allowed for some subjective interpretation of what a *harmful agenda* meant, but we prompted them to think of the scale of impact and whether an article might promote a real-world negative action or a strong negative belief about an individual or group of people. Lastly, they were asked to label the features found in Table 2, with the associated definitions provided, and provide 1-3 supporting evidence spans from the article for each label. They were prompted to first consider the article's primary weak label, and not to exhaustively label features. Since the features and score were labeled separately, we did not enforce any particular relationship between an individual feature and the overall label. See Appendix C for the full task instructions. We asked them to annotate a broader list of features than we used in our models for this paper to enable future work on this problem.

The full evaluation dataset, NEWSAGENDAS, consists of 506 annotated articles with 882 fine-grained label annotations. Each article additionally has its original weak label. See Appendix D for the label and score distribution and dataset examples.

2.4 Annotation Quality

To measure agreement between annotators, we held out an additional 90 articles for annotation by at least 2 graduate students (on average 3.4 students per article) studying natural language processing or journalism. We asked annotators just to label the harmful agenda score and to identify whether a specific feature from Table 2 was present. For each feature, we presented 5 articles with that weak label and 5 random articles. For sentiment, we presented this task as a 3-way classification between positive, neutral, and negative (see Appendix C for full task instructions). We then computed Cronbach's alpha (a measure of internal consistency (Cronbach, 1951)) across the annotators' responses. We observed good agreement across the harmful agenda scores (Table 3), and moderate agreement across the individual feature labels. These results indicate the data is of reasonable quality but future work could place more emphasis on how to well-annotate some of the trickier features.

Annotation Type	Cronbach's Alpha
Harmful Agenda Scores	0.78 (0.69, 0.84)
Feature Labels	0.53 (0.35, 0.67)

Table 3: Cronbach's alpha consistency measure for the annotated scores and feature labels in the annotation quality experiments. 95% confidence intervals are shown in parentheses. As a reference, randomly generated scores/labels produce a Cronbach's Alpha <0.06.

2.5 Labels

We define different sets of feature labels used in the paper for clarity:

1. **Annotated gold labels** - Feature labels assigned by our annotators in NEWSAGENDAS.
2. **Weak labels** - Feature labels assigned at the source-level from the FakeNewsCorpus.
3. **BERT/FRESH labels** - Feature labels predicted by our trained models (seen in Sec. 4).

The annotated gold labels are the standard which we can evaluate our system against, but we cannot train on them since there is not enough data per label and we cannot contaminate evaluation results by training on the evaluation data. We therefore use the weak labels for training, since there is a large quantity of weak labelled articles, although they are not as accurate.

3 Methods

We leverage large weakly labeled datasets to train feature classifiers for our features of interest. We prioritize exploring different levels of interpretability in the models we compare and what performance tradeoffs come at each level. To focus our analysis, we select **7 features** to study in-depth: *clickbait*, *junk science*, *hate speech*, *conspiracy theories*, *propaganda*, *satire*, and *negative sentiment*. Out of the 4 features we excluded, 3 did not have enough labelled data. For the 4th, political bias, after consulting our journalism experts, we determined the relationship between harmful agendas in news articles and political bias is nuanced and needs further study. We therefore leave political bias to future work to promote simplicity and interpretability in our approach.

3.1 Models

As shown in Figure 1, our approach is to separately train individual neural feature classifiers for each of the 7 features of interest. We then combine these features using a linear classifier to produce the final agenda classification. Our model is interpretable at the final level since the feature vector indicates the features that contribute to the final classification. It is also interpretable at the feature-level, where 6 of 7 features are derived from rationale-based models, which indicate the subset of input tokens that contribute to the feature classification.

Since we want to ensure faithfulness and interpretability, we derive our rationale model from the FRESH framework (Jain et al., 2020) (see Figure 1). We first finetune a BERT model (`Extractor BERT`) to predict a feature label from the full article text. For each token in the document, we derive a saliency score from the [CLS] token attention weights in the penultimate layer of this extractor. We extract as a rationale the top 20% most important tokens (with respect to saliency scores), irrespective of contiguity (each word is treated independently). Next, we finetune a second BERT model (`Predictor BERT`) to predict the feature label using only these extracted rationale tokens concatenated as input. This approach differs from the original FRESH paper in that we do not use a human-annotated dataset to introduce additional token-level supervision in rationale extraction. We also modify the FRESH framework to leverage positional embeddings for tokens. See Appendix A for details on training hyperparameters.

For the sentiment classification, we use the VADER classifier built into the NLTK Python library (Hutto and Gilbert, 2014; Loper and Bird, 2002). We choose VADER over more recent LLM-based sentiment analysis models, to facilitate interpretability. We compute the compound polarity score on a concatenation of the article title and contents. Articles with a compound score less than 0 are labeled as negative.

3.2 Training Data

For training data for the individual feature detectors, we use articles and weak labels from the same datasets described in Section 2.2 (however, we remove any articles used in NEWSAGENDAS). We handle negative sentiment labels at the model level (discussed in the next section).

Since the FakeNewsCorpus was collected from a broad scrape of unreliable websites, we noticed many of the texts did not fit the format of a news article. We therefore only use articles from the Fak-

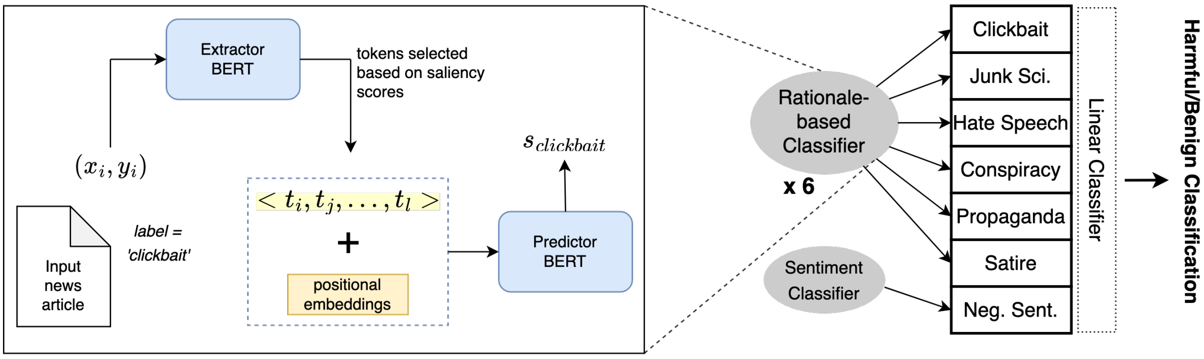


Figure 1: The full system consists of 6 different rationale-based classifiers and a simple sentiment classifier. These 7 features are input to a linear classifier which outputs the final classification.

eNewsCorpus whose source overlaps with the list of sources used by NELA-GT (Gruppi et al., 2021) or Li et al. (2020)'s Covid-19 dataset in order to filter for high quality sources. While this approach is not exhaustive, it significantly improves the quality of the data since the sources are validated by multiple misinformation datasets. We also search and remove URLs and variants of the source names from the articles to avoid model memorization of source-label pairings.

For each individual feature detector's training dataset, we sample 2,500 articles with the feature label we hope to detect (positive examples), and sample a range of negative examples based on a set of criteria (see Appendix E for details on negative examples for each feature). For each label, we adopt a weighted sampling strategy to increase the diversity of sources. We assign each article from a website w a weight $\frac{1}{c_w}$, where c_w is the total count of articles from website w. We then normalize these weights to sum to 1.

We additionally hold out 500 articles for the dev set and 500 articles for the test set. The test set articles come from a different set of websites than were used for the train and dev sets to make sure the test scores can not be inflated by any model memorization of website-specific styles.

4 Results

We investigate a series of research questions that analyze the efficacy of our overall approach, as well as individual components in our dataset and models.

4.1 How well can we predict the overall agenda score?

We experiment with predicting the NEWSAGENDAS annotated agenda score using different variants of our system. We fit the final logistic regression layer to the data using 10-fold cross-validation. The input is the 7 binary feature labels and the output is a binary classification of harmful or benign agenda - we bucket agenda scores 1-3 as benign and 4-5 as harmful (annotators gave a score of 3 when they were unsure of whether there was a harmful agenda in the text). We compare our method using the predicted features against three baselines: (1) predicting the majority class (0-benign), (2) using the weak source-level feature labels for logistic regression, and (3) finetuning a BERT model to classify the agenda (see Table 4). Baseline (2) demonstrates how this approach may be limited by the quality of the weak labels. Baseline (3) demonstrates a comparison against a fully black box model. We additionally compare against logistic regression using the annotated gold labels as an oracle. Using the annotated gold labels indicates a rough upper bound on performance for this type of feature-based approach, but could not be used in practice since it relies on a human annotating the articles. Note that the performance of the oracle implies a significant scope for improvement, and re-affirms our hypothesis that detecting harmful agendas in news articles is an especially difficult task for an automated system.

The oracle logistic regression model with the human annotated gold labels performs well, indicating our features of interest are very useful for the ultimate classification and promote interpretable classification of article agenda. The three systems we compare (with three different levels of interpretability) all perform better than both the majority baseline and logistic regression using just article weak labels. We also see that while we lose a little performance for every increase in interpretability (differences shown in table are statistically signif-

Method	Accuracy	Bal. Acc.
Oracle Logistic Reg.	76.7	75.6
Predict Majority Class	58.4	50.0
Weak Logistic Reg.	58.9	58.4
BERT-based Baseline	63.0 (0.21)	62.2 (0.36)
BERT System (Ours)	60.1 (0.02)	60.1 (0.02)
FRESH System (Ours)	59.3 (0.03)	59.3 (0.03)

Table 4: Overall performance (accuracy and balanced accuracy) on detecting harmful agendas in NEWSAGENDAS. The oracle logistic regression uses the annotated gold labels. Results are averaged with standard deviation shown in parentheses for neural models.

Feature	Annot.	Weak	BERT	FRESH
Clickbait	0.96	0.12	0.47	0.45
Junk Sci.	0.22	-0.16	-0.28	-0.45
Hate Sp.	**1.76**	0.21	**0.57**	**0.61**
Conspir.	0.86	0.20	-0.24	-0.07
Propagan.	1.31	**0.80**	0.55	0.42
Satire	0.62	0.34	0.22	0.17
Negative	1.55	n/a	0.53	0.52

Table 5: Weights for each feature learned by the logistic regression models across different feature label sets. The weights are averaged across the different cross-validation subsets and across seeds when appropriate.

icant by a two-sample t-test, p<.0001), it is possible to build interpretable models that are almost as effective as the black box models for this task. This interpretability is critical because a real-world system with this accuracy would require human oversight. The strong results of the oracle model also demonstrate that investing in better feature detectors could result in improved overall agenda classification, even beyond the black-box approach.

4.2 How are the features in NEWSAGENDAS related to the overall agenda score?

We first perform a pairwise analysis of which labels are more related to higher agenda scores over others in NEWSAGENDAS, using a pairwise Wilcoxon test. *Hate speech* and *negative sentiment* are associated with higher scores most often over other labels, suggesting that these two features are particularly strong indicators of a harmful agenda. Interestingly, *call to action* loses this pairwise comparison most often, even though it seems this label would be the biggest indicator of the article encouraging a real-world outcome. This may be because *call to action* was the least represented feature in the data (only labeled 8 times) so there is not a lot of data on this feature. *Neutral sentiment* and *satire* are associated with lower scores most often over other labels, suggesting that these two features are stronger indicators of a benign agenda. See Appendix F for more details on this analysis.

We also look at the weights learned by the final logistic regression layer over the features to determine what relationship the models learn between the feature labels and the final harmful agenda score. We see that almost all of the models place the highest weight (noted in bold) on *hate speech* with *negative sentiment* and *propaganda* generally coming in second. The models generally place the lowest weights on *junk science*, *conspiracy theories*, and *satire*.

4.3 How well do our feature detectors work?

In order to evaluate how well each feature classifier learned its training task (predicting the weak label from the FakeNewsCorpus for its feature), we evaluate predicted labels against weak labels across three datasets: 1) the validation set, 2) the test set, and 3) NEWSAGENDAS. We compare the FRESH-based models relative to the baseline of just using the fine-tuned extractor BERT model to predict the label to explore different levels of interpretability.

In Table 6, we see that the feature classifiers generalize effectively to articles from new sources in the test set, although the performance drop (relative to the validation set) indicates that the models are relying on some source-specific qualities of articles during training. We also see reasonable performance on the articles in NEWSAGENDAS with the exception of the satire model which performs poorly. We think the poor satire performance is because the training satire articles came from higher quality websites than many of the sites in the FakeNewsCorpus and therefore the text style may be too different to transfer to many of the articles in NEWSAGENDAS.

We then evaluate how well the predicted labels agree with the annotated gold labels. To measure overlap between predicted labels and annotated gold labels, we report the intersection-over-union (IOU) and the recall for the classifiers (see Table 7). As a baseline, we include the agreement between the weak labels and the annotated gold labels. The generally low weak label agreement shows that the source-level labels for articles provide fairly distant

	Val. Set		Test Set		NEWSAGENDAS	
Feature	BERT	FRESH	BERT	FRESH	BERT	FRESH
Clickbait	**90.5** (0.8)	88.7 (0.6)	**61.1** (1.3)	59.0 (0.3)	**76.9** (0.5)	71.6 (3.6)
Junk Science	**93.9** (0.7)	93.0 (0.8)	89.3 (0.8)	**89.5** (0.7)	**77.4** (1.6)	73.8 (2.5)
Hate Speech	**91.7** (0.3)	90.8 (0.8)	83.0 (1.3)	**83.4** (0.8)	**65.4** (0.7)	64.4 (1.1)
Conspiracy Theory	**94.2** (0.3)	93.2 (0.6)	**74.9** (1.0)	74.3 (1.6)	**62.7** (1.5)	61.7 (1.4)
Propaganda	**91.9** (0.6)	91.1 (0.3)	70.4 (1.1)	**71.5** (2.2)	**77.3** (1.3)	73.1 (1.8)
Satire	**95.9** (0.2)	94.5 (0.6)	66.9 (2.1)	**73.1** (2.4)	**51.5** (0.4)	**51.5** (0.6)

Table 6: Mean balanced accuracy scores (standard deviation in parentheses) for predicting the weak labels using the BERT and FRESH feature classifiers.

supervision relative to human judgment. We see that the BERT and FRESH models have worse but fairly similar overlap as the weak labels in many cases. The junk science and satire models have the least overlap. The black-box BERT model seems to have a slight advantage on the FRESH model, indicating there is an interpretability/performance tradeoff.

4.4 Are the extracted rationales useful?

We know that the FRESH rationales are useful to the BERT-predictors because our FRESH results show that BERT is able to achieve comparable prediction accuracy when using just the rationales as input as compared to using the entire text as input. Evaluating whether the FRESH rationales are also useful to humans is trickier. We analyze the percent of non-stopword rationale tokens that were also contained in the human-annotated rationales. However, we saw that the scores were not reliably different from just selecting the first 350 characters of the article as the rationale. This is likely because the generated rationales contain non-contiguous tokens from throughout the article, whereas the human-annotated rationales are 1-3 sentences. We therefore need to explore further human evaluation methods to quantitatively determine how well the model is rationalizing.

Through manual inspection, the rationales also seem meaningful to a human. We show three examples of common scenarios in Table 8 that demonstrate the quality of the rationales and the low word overlap score with the human-annotated rationales. The first example in this table illustrates a case where the human and FRESH model chose different labels for the article but both labels and rationales seem reasonable. The second example shows a case where the human and FRESH model agreed on the label, and the model rationale actually shares almost all the major keywords of the human rationale (although these words are not contiguous and in the same order as in the case of the human rationale). The final example then shows a case where the human and FRESH model agreed on the label, but chose rationales with very few overlapping words other than *Washington D.C.* and *socialism.*

5 Related Work

Disinformation and Misinformation. There are many previous approaches which have studied detection of misinformation and disinformation and which would be useful in combination with the detectors developed in this work (e.g., an agenda detection system flags an article to then go through a fact-checking pipeline). Research on detecting fake news includes detectors based on linguistic features (Gravanis et al., 2019), fact-checking based systems (Ciampaglia et al., 2015), social context or propagation network based approaches (Shu et al., 2020b; Wu et al., 2015; Liu and Wu, 2018), multimodal approaches (Khattar et al., 2019), etc. Other work has focused on characterizing/defining disinformation as a whole and developing classification schemas for campaigns (Booking et al., 2020; Fallis, 2015). However, neither disinformation detection nor characterization has explicitly looked at the more specific identification of a harmful agenda in an article.

Intent Detection. An agenda requires intention so detecting a harmful agenda is a type of intent detection. Intent detection is used in many settings with systems using slot-filling (Niu et al., 2019), conversational techniques (Larson et al., 2019; Casanueva et al., 2020), and language understanding (Qin et al., 2019). There has also been research into what intentions are involved with news articles specifically - on the intention of writing vs. sharing articles (Yaqub et al., 2020), the journalis-

	IOU			Recall-1		
Feature	Weak	BERT	FRESH	Weak	BERT	FRESH
Clickbait	**32.0**	30.9 (0.7)	25.3 (1.6)	**53.3**	46.4 (1.3)	40.2 (8.5)
Junk Science	**18.5**	17.1 (0.8)	12.5 (4.2)	41.7	75.0 (6.8)	**77.8** (7.8)
Hate Speech	16.5	18.0 (1.3)	**19.1** (2.8)	34.7	**64.6** (2.5)	57.1 (7.2)
Conspiracy	**27.7**	18.8 (0.5)	18.4 (0.3)	40.3	60.1 (1.2)	**67.5** (7.4)
Propaganda	**56.2**	43.0 (1.6)	40.1 (2.7)	**77.1**	60.0 (2.5)	59.8 (4.5)
Satire	**47.9**	2.8 (0.8)	2.3 (0.8)	**61.4**	2.9 (2.4)	2.4 (0.8)
Negative Sentiment	24.0			73.5		

Table 7: Agreement of the weak labels, BERT-predicted labels, and FRESH-predicted labels with NEWSAGENDAS' annotated gold labels. Metric reported is mean IOU/Recall-1 (standard deviation in parentheses for predicted labels).

Human-annotated	**Model-predicted**	**Article Opening**
Negative Sentiment: American and global audiences have been bombarded with media images of wailing children in holding facilities, having been separated from adults (maybe their parents, maybe not) detained for illegal entry into the United States.	**Propaganda**: Atrocity Porn and Hitler Memes and Daddy ! since parents - caging children racist FDRs Indeed , voted for Trump is now Americans Nazis Hitler President Donald Trump First Lady ? sanctuary - border Trumps terrorism - His	**Atrocity Porn and Hitler Memes** - American and global audiences have been bombarded with media images of wailing children in holding facilities, having been separated from adults (maybe their parents, maybe not) detained for illegal entry into the United States. The images have been accompanied by gut-wrenching audio...
Junk Science: Apple cider vinegar has so many benefits, but personally one of the reasons I like it best is because of the digestive and metabolism boosting benefits.	**Junk Science**: 6 Things Your Body Shot Apple Cider Vinegar Daily amazing health benefits apple cider vinegar (acv salad health benefits apple cider vinegar Why ? well 1 active Metabolism It This body protein It Boosts helps iron oxygen Iron oxygen the oxygen muscle 5 It helps glucose 6 acids produce sleep a video...	**6 Things That Happen To Your Body When You Take A Shot Of Apple Cider Vinegar Daily** - We've all heard about the amazing health benefits of apple cider vinegar (acv) right? but we forget to take it or how to use it. Most people add a little bit to the top of their salad when they remember but that's about it...
Propaganda: It isnnt any of their usual villains, the corporations and banks, who made Washington D.C. so miserable. It's the triumph of socialism.	**Propaganda**: Millionaire Poverty Pimps Fight 'Income Inequality' Washington D.C. citys food Obama Imperial City Income Inequality in America : Oligarchy Middle Class , the town hall oligarchic city income inequality government I nt Washington Its socialism bedroom left A socialist socialist socialist nt politics millionaire poverty	**Millionaire Poverty Pimps Fight 'Income Inequality'** - The bottom fifth of Washington D.C. account for just 2% of the city's income. It has one of the highest poverty rates in the country and the highest food stamp use. And under Obama, the Imperial City of the politicians and the poor was surrounded by some of the wealthiest districts in the country...

Table 8: Examples of labels/rationales annotated by humans and predicted by FRESH. The FRESH rationale is a concatenation of the most salient words in the text, whereas the human-annotated rationale is typically a sentence. We also highlight the FRESH-rationale words in the article opening (the title and first couple sentences) for clarity.

tic role of articles (Mellado, 2015; Tsang, 2020), and what motivates people to create and share fake news knowingly (Osmundsen et al., 2020). Finally, there has also been work on detecting deception (an intentional act) (Rubin and Conroy, 2012). However, these works have not looked specifically at automatic classification of a harmful agenda in news.

6 Conclusion

In this work, we formalize the open challenge of detecting harmful agendas in news articles, release an initial evaluation dataset, and develop an interpretable system for this task. We hope our work can encourage future investment in this area - such as exploring state-of-the-art intepretable models for detecting the features we discussed, further characterizing article agenda beyond a binary classification, or investigating the interplay between text features and metadata like article source.

7 Limitations

Given the subjective nature of our proposed task, this work does have some limitations and challenges. Firstly, the notion of harm or potential to do harm is seldom an objective factor and is also difficult to measure or quantify. Our experiments on inter-annotator agreement use a small dataset, so this study could be expanded with collaboration with social science researchers to better qualify how people perceive the agenda in different articles. Our work is also grounded in the United States, so it may have limited applications to the news in other countries (discussed more in Section 8). Secondly, our data and framework can be used to build and train a system to perform post-hoc detection of harmful agendas in news articles. However, in a real-world system, this identification would likely need to happen on the fly, so as to make readers aware of these agendas as they are exposed to the articles. Finally, another aspect that we have not addressed in this study is the effect that a platform or community may have on the perceived harm in an article. For example, on dedicated social media channels hosting discussions on alternate theories and contentious topics (such as the efficacy of COVID-19 vaccines), a junk science article with dubious claims may not be as "harmful" as opposed to the same article being posted on an open forum where readers may perceive it as scientific fact, thereby making the article more "harmful". The context in which news articles are disseminated may have a profound impact on this perceived harm and this may be an interesting direction for future exploration.

8 Ethical Considerations

8.1 Censorship

Detecting harmful agendas in news articles has the obvious possible downstream use of filtering or banning articles which are flagged as such from being shared on social media platforms. We have already seen debate over content filtering like this take place in relation to sites like Facebook, Instagram, and Twitter moderating the dissemination of "fake news" on their platforms. One could imagine an automatic harmful agenda detector becoming part of this kind of content moderation pipeline. However, if the AI system incorrectly flags articles, it may end up censoring legitimate political speech. For this reason, we discourage any real-world use of this system at this time until further research and analysis can be completed. Additionally, we want to emphasize that this detection system should be paired with a fact-checking system to make sure that the pipeline considers the interplay between agenda and misinformation, and does not just flag biased or opinionated free speech.

8.2 Cultural/Ideological Context

Characterizing an article as containing a harmful agenda forces definitions of what constitutes harm, which has been studied for millennia by philosophers of ethics. Normative ethics is the study of how to articulate the basic tenets of what is good and bad (Kagan, 2018). Broadly, normative ethics is divided into teleological/consequentialist (focusing on consequences to determine good/bad (Sinnott-Armstrong, 2021)) and Deontological (moral worth is intrinsic to an action (Alexander and Moore, 2021)). In this work, we focus on real-world harm which draws more on consequentialism.

Ultimately, as these opposing theories demonstrate, there is no universal interpretation of good and bad, or scale for evaluating harm. For this reason, any attempt to characterize news articles will come from a certain cultural context and perspective. The dataset we present is subject to the biases and cultural contexts of the annotators involved, so while it represents a useful starting point for work and data collection efforts in this area, future datasets around this problem must be conscious of recruiting a diverse and large annotator pool. An example of an individual bias could be that for a devout believer in the Christian God, writing which denounces God's existence could be considered harmful disinformation. Whereas from the broader societal perspective of the United States, such a piece of writing would likely be considered a benign opinion piece.

Additionally, we want to clearly state that the framing of this research (in terms of what constitutes harm, fact, etc.) was through a United States sociopolitical context, and therefore likely does not apply across other global contexts without modifications. In conclusion, any future applications of news agenda characterization in the real world need to be very clear about the particular cultural context it is designed to operate in, what assumptions it uses, and what applications it is appropriate for.

Acknowledgments

This work is supported by the DARPA SemaFor project (HR001120C0123). The authors would also like to thank journalism subject matter experts

Kristy Roschke, Dan Gillmor and Scott Ruston for their insightful feedback and discussions. The views, opinions and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

References

Wassim Ahmed, Joseph Downing, Marc Tuters, and Peter Knight. 2020. Four experts investigate how the 5g coronavirus conspiracy theory began. *The Conversation*.

Firoj Alam, Fahim Dalvi, Shaden Shaar, Nadir Durrani, Hamdy Mubarak, Alex Nikolov, Giovanni Da San Martino, Ahmed Abdelali, Hassan Sajjad, Kareem Darwish, et al. 2021. Fighting the covid-19 infodemic in social media: a holistic perspective and a call to arms. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 15, pages 913–922.

Larry Alexander and Michael Moore. 2021. Deontological Ethics. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*, Winter 2021 edition. Metaphysics Research Lab, Stanford University.

Miguel A Alonso, David Vilares, Carlos Gómez-Rodríguez, and Jesús Vilares. 2021. Sentiment analysis for fake news detection. *Electronics*, 10(11):1348.

Alberto Barrón-Cedeno, Israa Jaradat, Giovanni Da San Martino, and Preslav Nakov. 2019. Proppy: Organizing the news based on their propagandistic content. *Information Processing & Management*, 56(5):1849–1864.

Zach Bastick. 2021. Would you notice if fake news changed your behavior? an experiment on the unconscious effects of disinformation. *Computers in human behavior*, 116:106633.

Emerson T Booking, Alyssa Kann, and Max Rizzuto. 2020. Dichotomies of disinformation. Accessed: Jan. 2022.

Robert Carnahan, Sanjay Mishra, and Flora Teoh. 2021. Inaccurate clickbait headline in forbes article used to promote the false claim that covid-19 vaccines change our dna. Accessed: Jan. 2022.

Inigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. *arXiv preprint arXiv:2003.04807*.

Wei-Fan Chen, Khalid Al-Khatib, Benno Stein, and Henning Wachsmuth. 2020. Detecting media bias in news articles using gaussian bias distributions. *arXiv preprint arXiv:2010.10649*.

Yimin Chen, Niall J Conroy, and Victoria L Rubin. 2015. Misleading online content: recognizing clickbait as" false news". In *Proceedings of the 2015 ACM on workshop on multimodal deception detection*, pages 15–19.

Giovanni Luca Ciampaglia, Prashant Shiralkar, Luis M Rocha, Johan Bollen, Filippo Menczer, and Alessandro Flammini. 2015. Computational fact checking from knowledge networks. *PloS one*, 10(6):e0128193.

Lee J Cronbach. 1951. Coefficient alpha and the internal structure of tests. *psychometrika*, 16(3):297–334.

Giovanni Da San Martino, Seunghak Yu, Alberto Barrón-Cedeno, Rostislav Petrov, and Preslav Nakov. 2019. Fine-grained analysis of propaganda in news article. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 5636–5646.

Don Fallis. 2015. What is disinformation? *Library trends*, 63(3):401–426.

Jennifer Golbeck, Matthew Mauriello, Brooke Auxier, Keval H Bhanushali, Christopher Bonk, Mohamed Amine Bouzaghrane, Cody Buntain, Riya Chanduka, Paul Cheakalos, Jennine B Everett, et al. 2018. Fake news vs satire: A dataset and analysis. In *Proceedings of the 10th ACM Conference on Web Science*, pages 17–21.

Georgios Gravanis, Athena Vakali, Konstantinos Diamantaras, and Panagiotis Karadais. 2019. Behind the cues: A benchmarking study for fake news detection. *Expert Systems with Applications*, 128:201–213.

Maurício Gruppi, Benjamin D Horne, and Sibel Adalı. 2021. Nela-gt-2020: A large multi-labelled news dataset for the study of misinformation in news articles. *arXiv preprint arXiv:2102.04567*.

Stefano Guarino, Noemi Trino, Alessandro Celestini, Alessandro Chessa, and Gianni Riotta. 2020. Characterizing networks of propaganda on twitter: a case study. *Applied Network Science*, 5(1):1–22.

Suyin Haynes. 2020. As coronavirus spreads, so does xenophobia and anti-asian racism. Accessed: Jan. 2022.

Clayton Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 8.

Sarthak Jain, Sarah Wiegreffe, Yuval Pinter, and Byron C. Wallace. 2020. Learning to Faithfully Rationalize by Construction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4459–4473, Online. Association for Computational Linguistics.

Shelly Kagan. 2018. *Normative ethics*. Routledge.

Dhruv Khattar, Jaipal Singh Goud, Manish Gupta, and Vasudeva Varma. 2019. Mvae: Multimodal variational autoencoder for fake news detection. In *The world wide web conference*, pages 2915–2921.

Stefan Larson, Anish Mahendran, Joseph J Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K Kummerfeld, Kevin Leach, Michael A Laurenzano, Lingjia Tang, et al. 2019. An evaluation dataset for intent classification and out-of-scope prediction. *arXiv preprint arXiv:1909.02027*.

Or Levi, Pedram Hosseini, Mona Diab, and David A Broniatowski. 2019. Identifying nuances in fake news vs. satire: Using semantic and linguistic cues. *arXiv preprint arXiv:1910.01160*.

Yichuan Li, Bohan Jiang, Kai Shu, and Huan Liu. 2020. Mm-covid: A multilingual and multimodal data repository for combating covid-19 disinformation. *arXiv preprint arXiv:2011.04088*.

Yang Liu and Yi-Fang Brook Wu. 2018. Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In *Thirty-second AAAI conference on artificial intelligence*.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. *arXiv preprint cs/0205028*.

Julie Mastrine. 2022. How to spot 12 types of media bias. Accessed: Jan. 2022.

Claudia Mellado. 2015. Professional roles in news content: Six dimensions of journalistic role performance. *Journalism Studies*, 16(4):596–614.

Robert S Mueller III. 2020. *Internet research agency indictment*. Good Press.

Peiqing Niu, Zhongfu Chen, Meina Song, et al. 2019. A novel bi-directional interrelated model for joint intent detection and slot filling. *arXiv preprint arXiv:1907.00390*.

J Eric Oliver and Thomas Wood. 2014. Medical conspiracy theories and health behaviors in the united states. *JAMA internal medicine*, 174(5):817–818.

OpenAI. 2023. Gpt-4 technical report.

Mathias Osmundsen, Alexander Bor, Peter Bjerregaard Vahlstrup, Anja Bechmann, and Michael Bang Petersen. 2020. Partisan polarization is the primary psychological motivation behind "fake news" sharing on twitter.

Nishant Pandey. 2021. Fighting junk science in covid-ravaged india. Accessed: Jan. 2022.

Poynter. 2021. Several clickbait websites published misleading headlines claiming that "cigarettes block the coronavirus". Accessed: Jan. 2022.

Libo Qin, Wanxiang Che, Yangming Li, Haoyang Wen, and Ting Liu. 2019. A stack-propagation framework with token-level intent detection for spoken language understanding. *arXiv preprint arXiv:1909.02188*.

Tony Rogers. 2019. Objectivity and fairness in journalism.

Victoria L Rubin and Niall Conroy. 2012. Discerning truth from deception: Human judgments and automation efforts. *First Monday*, 17(5).

Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215.

Morgan Klaus Scheuerman, Jialun Aaron Jiang, Casey Fiesler, and Jed R Brubaker. 2021. A framework of severity for harmful content online. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW2):1–33.

Kai Shu, Amrita Bhattacharjee, Faisal Alatawi, Tahora H Nazer, Kaize Ding, Mansooreh Karami, and Huan Liu. 2020a. Combating disinformation in a social media age. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 10(6):e1385.

Kai Shu, Guoqing Zheng, Yichuan Li, Subhabrata Mukherjee, Ahmed Hassan Awadallah, Scott Ruston, and Huan Liu. 2020b. Leveraging multi-source weak social supervision for early detection of fake news. *arXiv preprint arXiv:2004.01732*.

Walter Sinnott-Armstrong. 2021. Consequentialism. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*, Fall 2021 edition. Metaphysics Research Lab, Stanford University.

Stephanie Jean Tsang. 2020. Issue stance and perceived journalistic motives explain divergent audience perceptions of fake news. *Journalism*, page 1464884920926002.

Soroush Vosoughi, Deb Roy, and Sinan Aral. 2018. The spread of true and false news online. *Science*, 359(6380):1146–1151.

Stephen J.A. Ward. 2010. Emotion in reporting: use and abuse. Accessed: Jan. 2022.

Ke Wu, Song Yang, and Kenny Q Zhu. 2015. False rumors detection on sina weibo by propagation structures. In *2015 IEEE 31st international conference on data engineering*, pages 651–662. IEEE.

Fan Yang, Arjun Mukherjee, and Eduard Dragut. 2017. Satirical News Detection and Analysis using Attention Mechanism and Linguistic Features. *arXiv:1709.01189 [cs]*. ArXiv: 1709.01189.

Waheeb Yaqub, Otari Kakhidze, Morgan L Brockman, Nasir Memon, and Sameer Patil. 2020. Effects of credibility indicators on social media news sharing intent. In *Proceedings of the 2020 chi conference on human factors in computing systems*, pages 1–14.

Xichen Zhang and Ali A Ghorbani. 2020. An overview of online fake news: Characterization, detection, and discussion. *Information Processing & Management*, 57(2):102025.

A Training Hyperparameters

We use BERT-for-Sequence-Classification (bert-base-cased) from Huggingface[3] for both the rationale extractor and the predictor, training on binary classification of the feature in question. We did not notice much sensitivity to hyperparameters during an initial grid-search, so we decided to use the AdamW optimizer with a learning rate of 1e-5; we applied an early stopping patience of 15 epochs and set the max number of epochs to be 50. All results are reported as an average with standard deviation across 3 different training runs (with random seeds 1000, 2000, 3000). We trained each FRESH model for several hours on 1 NVIDIA Titan Xp GPU. We also use the BERT models from the rationale extractor framework as a reference in our results since they are trained to predict the feature label from the article text. These BERT models are an artifact of training the FRESH models so they did not require additional computation.

B Annotator Recruitment and Training

We posted a recruitment notice on a journalism ListServ. We then hired the first four students who responded who met the criteria of current students at the same university as one of the authors and native English speakers. We hired the students through the university and compensated them at a rate of $20/hour for 9-12 hours of work each. This rate is above the minimum wage in the city where the students completed the work.

After completing hiring paperwork, students had a 1-on-1 call with one of the authors who explained the goal of the research and what the task would look like, and provided a chance to discuss concerns and questions. Throughout the process students could communicate with the authors at anytime over email with questions/concerns, and they could also opt-out of the work at anytime. Otherwise students were able to complete the work independently on the their own computers using the Amazon Mechanical Turk Workers Sandbox [4]. Students were compensated outside of the platform based on their hours, and no other workers on the platform completed the tasks.

C Annotator Instructions

For the annotation of NEWSAGENDAS, students were presented with the instructions shown in Figure 2. They were not required to answer any of the questions, which allowed them to skip a whole article if the content made them uncomfortable since many of the articles contained offensive language.

Articles were displayed to the annotators as shown in Figure 3. They were then asked the questions shown in Figure 4. The feature names we used with the annotators differed slightly from the wording presented in this paper to facilitate clarity for the annotators. Whereas for this paper, we wanted to use consistent terminology throughout. Annotators could expand the label definitions in Question 2 as shown in Figure 5.

We did not ask annotators any personal or demographic questions, and neither did we collect nor store any personal information about them.

For the annotation quality experiments, students were presented with the instructions shown in Figure 6. Articles were displayed as shown in Figure 3. The students were then asked the questions shown in Figure 7 for most feature labels, but the questions shown in Figure 8 for tone-related labels. They could once again expand the definitions of the labels if needed.

D NEWSAGENDAS Label Distribution and Examples

The distribution of agenda scores labeled in NEWSAGENDAS is shown in Table 9. The distribution of weak labels, annotated gold labels, and evidence spans for each feature is shown in Table 10. We also looked at the distribution of agenda scores across each feature, which is shown as heatmaps in Figure 9. Examples of annotated evidence spans for each feature label are shown in Table 11.

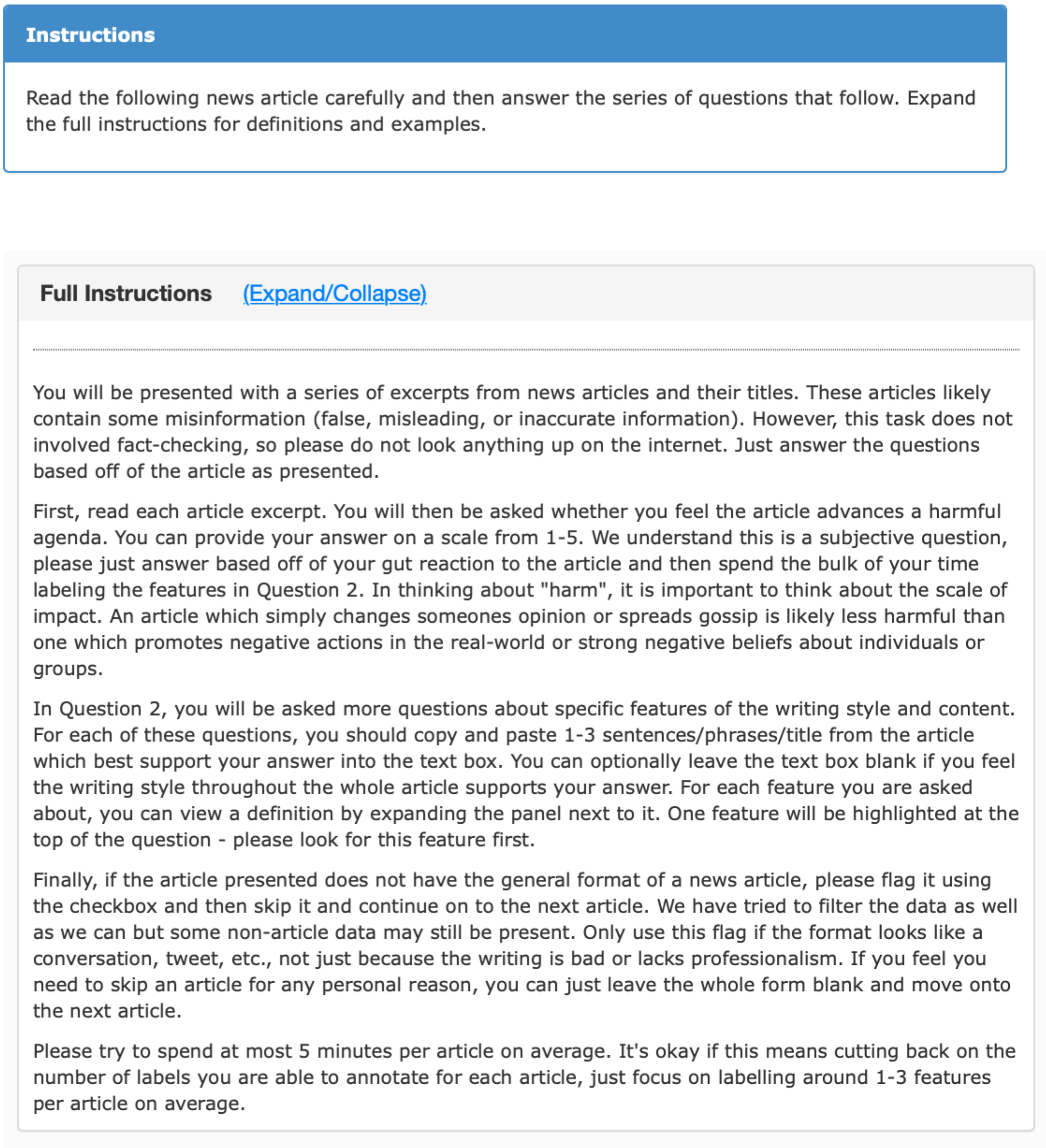
E Negative Examples for Training Feature Detectors

The challenge of negative sampling arises from the potential overlaps between the class labels. For example, an article can be both "junk science" and "conspiracy theory" in practice. In the FakeNewsCorpus, the websites (and thus the articles) can have multiple labels, including a primary label that best describes the source. However, these labels

---

[3] `https://huggingface.co/docs/transformers/index`

[4] https://workersandbox.mturk.com



Instructions

Read the following news article carefully and then answer the series of questions that follow. Expand the full instructions for definitions and examples.

Full Instructions (Expand/Collapse)

You will be presented with a series of excerpts from news articles and their titles. These articles likely contain some misinformation (false, misleading, or inaccurate information). However, this task does not involved fact-checking, so please do not look anything up on the internet. Just answer the questions based off of the article as presented.

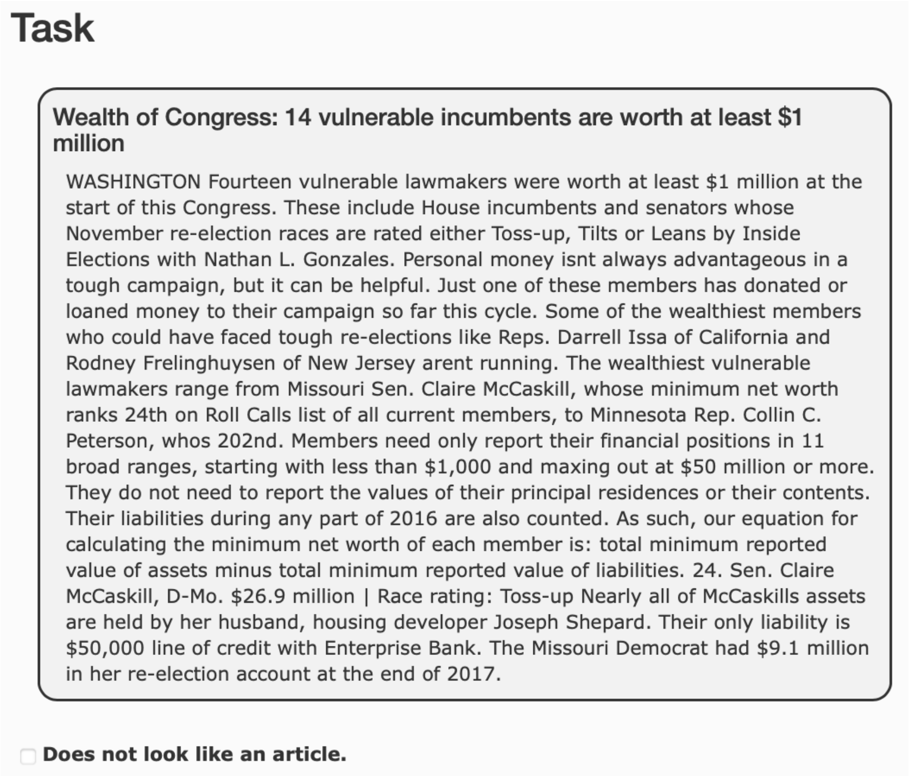
First, read each article excerpt. You will then be asked whether you feel the article advances a harmful agenda. You can provide your answer on a scale from 1-5. We understand this is a subjective question, please just answer based off of your gut reaction to the article and then spend the bulk of your time labeling the features in Question 2. In thinking about "harm", it is important to think about the scale of impact. An article which simply changes someones opinion or spreads gossip is likely less harmful than one which promotes negative actions in the real-world or strong negative beliefs about individuals or groups.

In Question 2, you will be asked more questions about specific features of the writing style and content. For each of these questions, you should copy and paste 1-3 sentences/phrases/title from the article which best support your answer into the text box. You can optionally leave the text box blank if you feel the writing style throughout the whole article supports your answer. For each feature you are asked about, you can view a definition by expanding the panel next to it. One feature will be highlighted at the top of the question - please look for this feature first.

Finally, if the article presented does not have the general format of a news article, please flag it using the checkbox and then skip it and continue on to the next article. We have tried to filter the data as well as we can but some non-article data may still be present. Only use this flag if the format looks like a conversation, tweet, etc., not just because the writing is bad or lacks professionalism. If you feel you need to skip an article for any personal reason, you can just leave the whole form blank and move onto the next article.

Please try to spend at most 5 minutes per article on average. It's okay if this means cutting back on the number of labels you are able to annotate for each article, just focus on labelling around 1-3 features per article on average.

Figure 2: The task instructions presented to annotators.



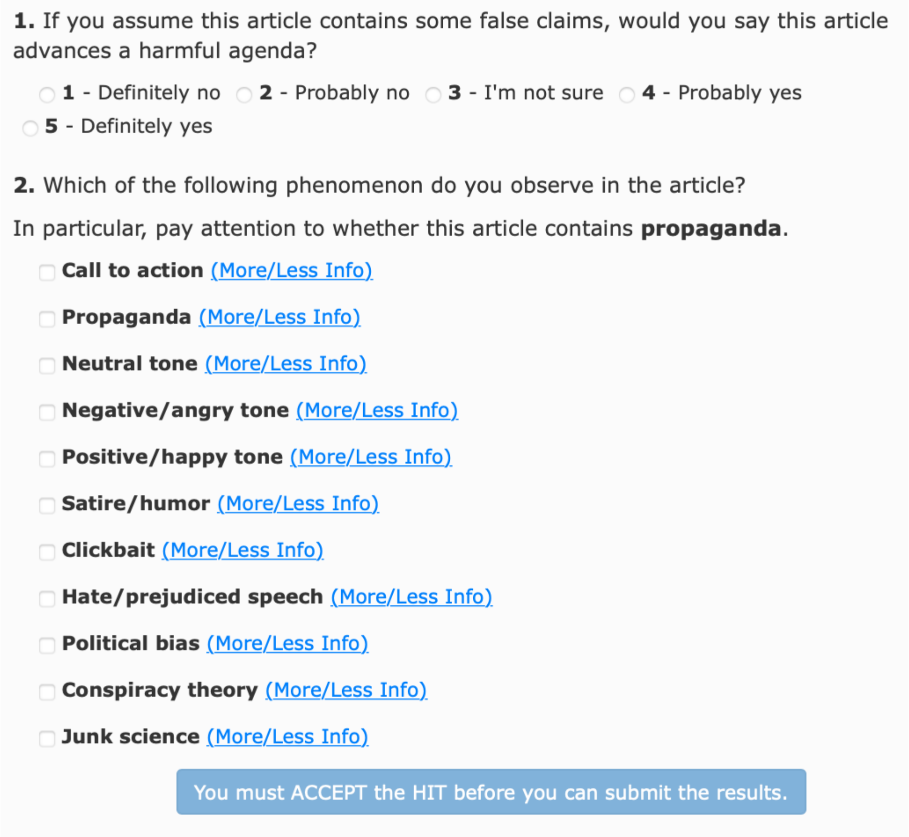
Task

Wealth of Congress: 14 vulnerable incumbents are worth at least $1 million

WASHINGTON Fourteen vulnerable lawmakers were worth at least $1 million at the start of this Congress. These include House incumbents and senators whose November re-election races are rated either Toss-up, Tilts or Leans by Inside Elections with Nathan L. Gonzales. Personal money isnt always advantageous in a tough campaign, but it can be helpful. Just one of these members has donated or loaned money to their campaign so far this cycle. Some of the wealthiest members who could have faced tough re-elections like Reps. Darrell Issa of California and Rodney Frelinghuysen of New Jersey arent running. The wealthiest vulnerable lawmakers range from Missouri Sen. Claire McCaskill, whose minimum net worth ranks 24th on Roll Calls list of all current members, to Minnesota Rep. Collin C. Peterson, whos 202nd. Members need only report their financial positions in 11 broad ranges, starting with less than $1,000 and maxing out at $50 million or more. They do not need to report the values of their principal residences or their contents. Their liabilities during any part of 2016 are also counted. As such, our equation for calculating the minimum net worth of each member is: total minimum reported value of assets minus total minimum reported value of liabilities. 24. Sen. Claire McCaskill, D-Mo. $26.9 million | Race rating: Toss-up Nearly all of McCaskills assets are held by her husband, housing developer Joseph Shepard. Their only liability is $50,000 line of credit with Enterprise Bank. The Missouri Democrat had $9.1 million in her re-election account at the end of 2017.

☐ **Does not look like an article.**

Figure 3: An example article as shown in the task format.



1. If you assume this article contains some false claims, would you say this article advances a harmful agenda?

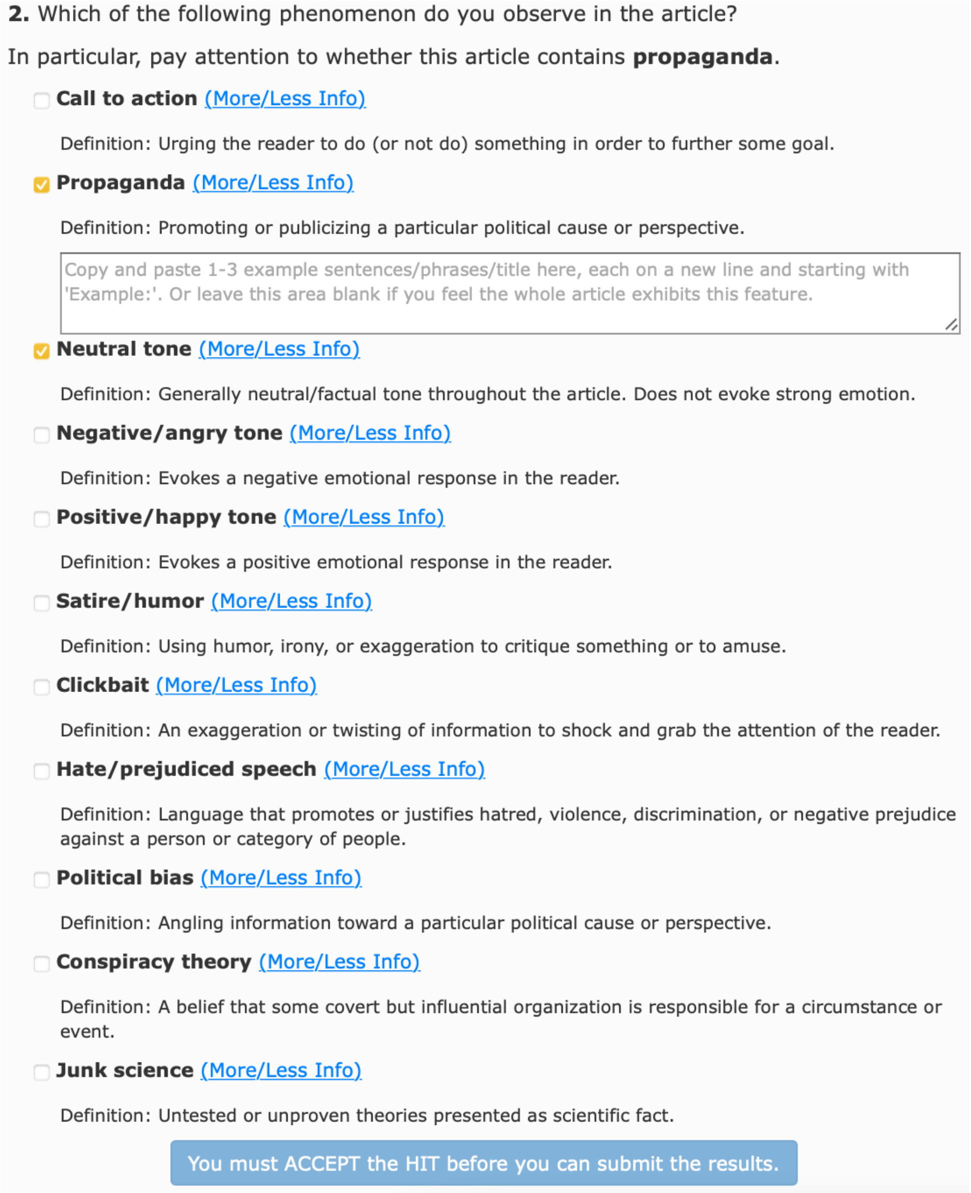
○ **1** - Definitely no ○ **2** - Probably no ○ **3** - I'm not sure ○ **4** - Probably yes ○ **5** - Definitely yes

2. Which of the following phenomenon do you observe in the article?

In particular, pay attention to whether this article contains **propaganda**.

- ☐ **Call to action** (More/Less Info)
- ☐ **Propaganda** (More/Less Info)
- ☐ **Neutral tone** (More/Less Info)
- ☐ **Negative/angry tone** (More/Less Info)
- ☐ **Positive/happy tone** (More/Less Info)
- ☐ **Satire/humor** (More/Less Info)
- ☐ **Clickbait** (More/Less Info)
- ☐ **Hate/prejudiced speech** (More/Less Info)
- ☐ **Political bias** (More/Less Info)
- ☐ **Conspiracy theory** (More/Less Info)
- ☐ **Junk science** (More/Less Info)

Figure 4: The questions asked of annotators with an example weak label.



2. Which of the following phenomenon do you observe in the article?

In particular, pay attention to whether this article contains **propaganda**.

☐ **Call to action** (More/Less Info)

Definition: Urging the reader to do (or not do) something in order to further some goal.

☑ **Propaganda** (More/Less Info)

Definition: Promoting or publicizing a particular political cause or perspective.

Copy and paste 1-3 example sentences/phrases/title here, each on a new line and starting with 'Example:'. Or leave this area blank if you feel the whole article exhibits this feature.

☑ **Neutral tone** (More/Less Info)

Definition: Generally neutral/factual tone throughout the article. Does not evoke strong emotion.

☐ **Negative/angry tone** (More/Less Info)

Definition: Evokes a negative emotional response in the reader.

☐ **Positive/happy tone** (More/Less Info)

Definition: Evokes a positive emotional response in the reader.

☐ **Satire/humor** (More/Less Info)

Definition: Using humor, irony, or exaggeration to critique something or to amuse.

☐ **Clickbait** (More/Less Info)

Definition: An exaggeration or twisting of information to shock and grab the attention of the reader.

☐ **Hate/prejudiced speech** (More/Less Info)

Definition: Language that promotes or justifies hatred, violence, discrimination, or negative prejudice against a person or category of people.

☐ **Political bias** (More/Less Info)

Definition: Angling information toward a particular political cause or perspective.

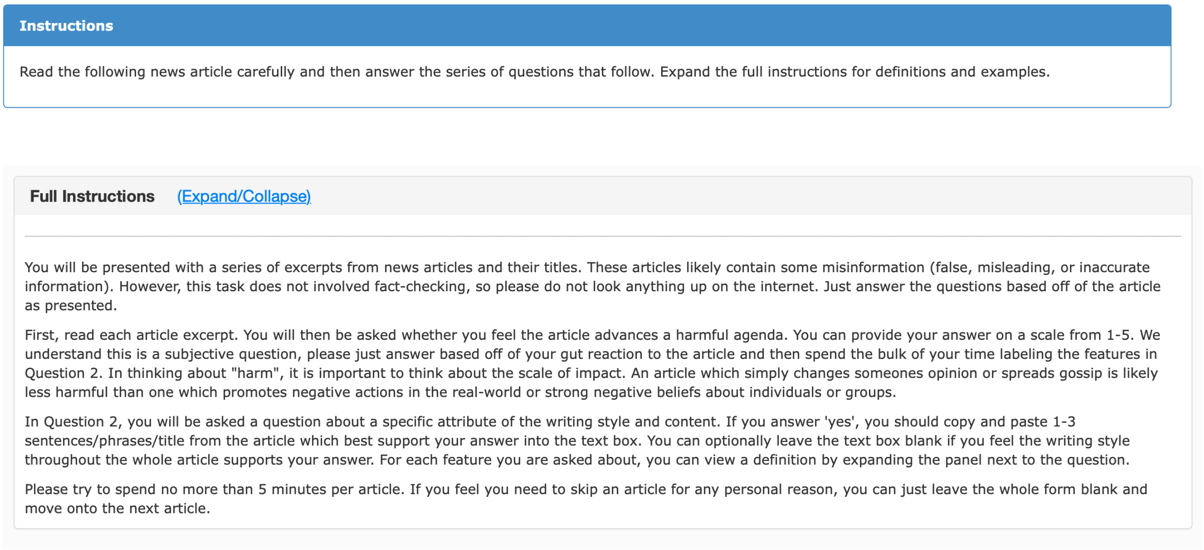
☐ **Conspiracy theory** (More/Less Info)

Definition: A belief that some covert but influential organization is responsible for a circumstance or event.

☐ **Junk science** (More/Less Info)

Definition: Untested or unproven theories presented as scientific fact.

Figure 5: An example of how the components in Question 2 could be expanded. Note that for 'Neutral Tone', there was no option to provide evidence as this feature was generally present throughout the article. Otherwise, if an annotator selected a checkbox, the option to provide evidence would appear.



Instructions

Read the following news article carefully and then answer the series of questions that follow. Expand the full instructions for definitions and examples.

Full Instructions (Expand/Collapse)

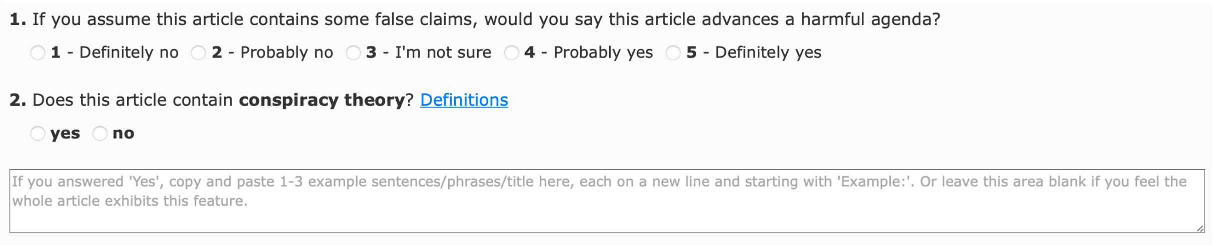
You will be presented with a series of excerpts from news articles and their titles. These articles likely contain some misinformation (false, misleading, or inaccurate information). However, this task does not involved fact-checking, so please do not look anything up on the internet. Just answer the questions based off of the article as presented.

First, read each article excerpt. You will then be asked whether you feel the article advances a harmful agenda. You can provide your answer on a scale from 1-5. We understand this is a subjective question, please just answer based off of your gut reaction to the article and then spend the bulk of your time labeling the features in Question 2. In thinking about "harm", it is important to think about the scale of impact. An article which simply changes someones opinion or spreads gossip is likely less harmful than one which promotes negative actions in the real-world or strong negative beliefs about individuals or groups.

In Question 2, you will be asked a question about a specific attribute of the writing style and content. If you answer 'yes', you should copy and paste 1-3 sentences/phrases/title from the article which best support your answer into the text box. You can optionally leave the text box blank if you feel the writing style throughout the whole article supports your answer. For each feature you are asked about, you can view a definition by expanding the panel next to the question.

Please try to spend no more than 5 minutes per article. If you feel you need to skip an article for any personal reason, you can just leave the whole form blank and move onto the next article.

Figure 6: The annotation quality experiment instructions.



1. If you assume this article contains some false claims, would you say this article advances a harmful agenda?

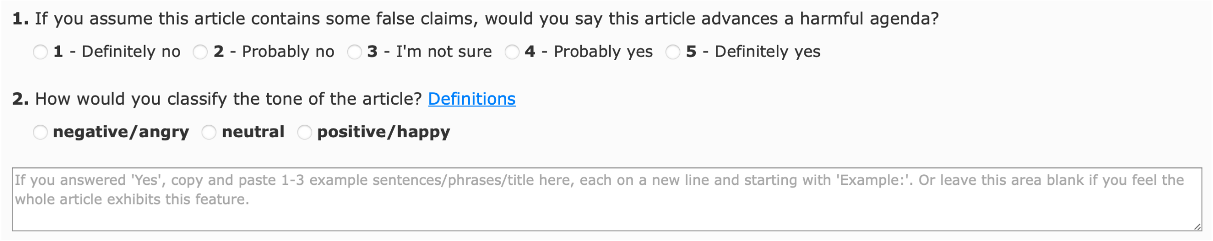
○ **1** - Definitely no ○ **2** - Probably no ○ **3** - I'm not sure ○ **4** - Probably yes ○ **5** - Definitely yes

2. Does this article contain **conspiracy theory**? Definitions

○ **yes** ○ **no**

If you answered 'Yes', copy and paste 1-3 example sentences/phrases/title here, each on a new line and starting with 'Example:'. Or leave this area blank if you feel the whole article exhibits this feature.

Figure 7: The questions asked in the annotation quality experiments for most feature labels.



1. If you assume this article contains some false claims, would you say this article advances a harmful agenda?

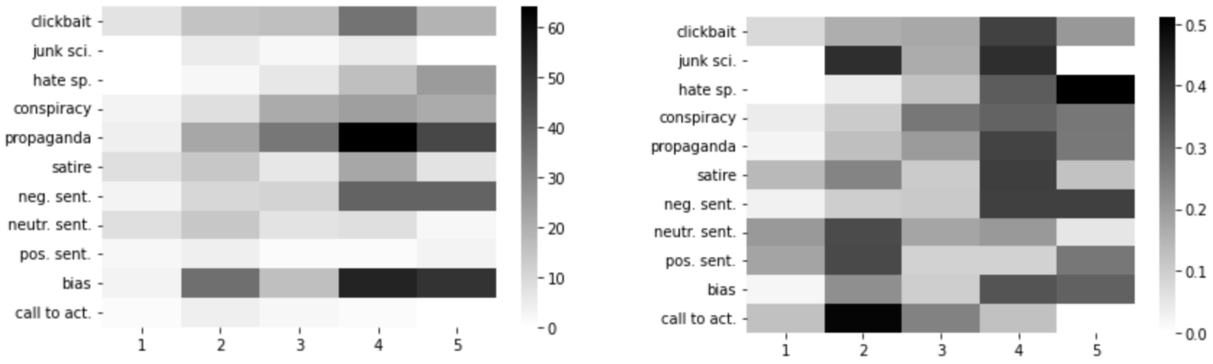
○ **1** - Definitely no ○ **2** - Probably no ○ **3** - I'm not sure ○ **4** - Probably yes ○ **5** - Definitely yes

2. How would you classify the tone of the article? Definitions

○ **negative/angry** ○ **neutral** ○ **positive/happy**

If you answered 'Yes', copy and paste 1-3 example sentences/phrases/title here, each on a new line and starting with 'Example:'. Or leave this area blank if you feel the whole article exhibits this feature.

Figure 8: The questions asked in the annotation quality experiments for tone-related labels.



(a) Counts of each harmful agenda score associated with each feature label.

(b) Fraction of the agenda scores associated with each feature label that fall into each bucket. Each row sums to 1.

Figure 9: The distribution of agenda scores associated with each feature label.

Agenda Score	Annotation Count
1	58
2	124
3	87
4	123
5	69

Table 9: Counts of the agenda scores labeled in NEWSAGENDAS. There are also 45 data points for which annotators chose not to label an agenda score but selected feature labels and evidence spans.

Feature	Weak	Annot.	Spans
Clickbait	83	110	158
Junk Science	13	15	19
Hate Speech	4	54	65
Conspiracy Theory	52	84	102
Propaganda	220	198	289
Satire	52	64	104
Negative Sentiment	–	113	103
Neutral Sentiment	–	42	–
Positive Sentiment	–	13	14
Political Bias	35	181	234
Call to Action	–	8	10

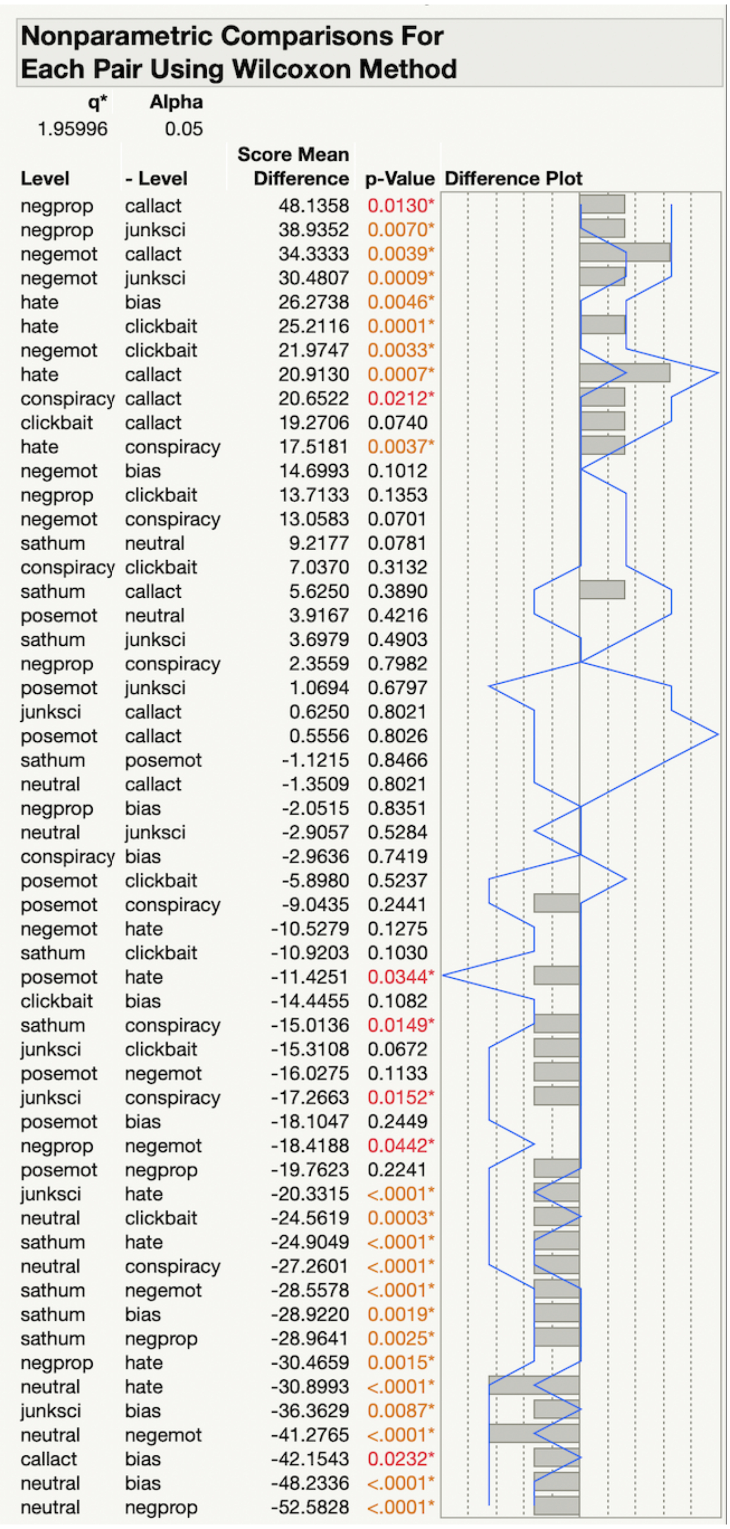
Table 10: Counts of weak labels, annotated gold labels, and evidence spans for each feature in NEWSAGENDAS.

were based on annotators' overall impression of a website, which may not capture all possible types of its articles. Evidence suggests that websites sharing junk science articles often share conspiracy articles, or articles possessing both features (more details in the next paragraph). Then, even if a website has "junk science" as its only label, some of its articles may still be "conspiracy." Therefore, articles from this website may not be proper negative examples for a conspiracy detector.

With this observation, we develop our criteria for negative examples. For a model that detects a specific label (referred to as the positive label), we quantify the positive label's overlap with other class labels using the overlap coefficient (Szymkiewicz–Simpson coefficient). The overlap between Label A and Label B is calculated as $\frac{|A \cap B|}{\min(|A|,|B|)}$, where A and B are the sets of websites whose multiple labels include Label A and Label B respectively. After exploratory experiments on the validation set, we adopted a threshold of 0.15 to filter out classes that overlap too much with the positive class. For example, the overlap coefficient of "junk science" and "conspiracy" is 0.5396, exceeding 0.15. Thus, excluding "conspiracy" articles from the negative examples can better train the "junk science" detector. The negative classes after applying this criterion can be found in Table 12. In addition to sampling from these selected negative classes, all negative samples must not have the positive label among their multiple labels. Since we have multiple negative classes, we include more negative examples than positive examples, depending on the availability of the former after applying the criteria. We adopt a standard class-weighted loss in training to handle class imbalance.

F Additional Results and Analysis

The full Wilcoxon pairwise comparisons (discussed in Section 4.2) are shown in Table 10.



Nonparametric Comparisons For Each Pair Using Wilcoxon Method

q*	Alpha
1.95996	0.05

Level	- Level	Score Mean Difference	p-Value	Difference Plot
negprop	callact	48.1358	0.0130*	
negprop	junksci	38.9352	0.0070*	
negemot	callact	34.3333	0.0039*	
negemot	junksci	30.4807	0.0009*	
hate	bias	26.2738	0.0046*	
hate	clickbait	25.2116	0.0001*	
negemot	clickbait	21.9747	0.0033*	
hate	callact	20.9130	0.0007*	
conspiracy	callact	20.6522	0.0212*	
clickbait	callact	19.2706	0.0740	
hate	conspiracy	17.5181	0.0037*	
negemot	bias	14.6993	0.1012	
negprop	clickbait	13.7133	0.1353	
negemot	conspiracy	13.0583	0.0701	
sathum	neutral	9.2177	0.0781	
conspiracy	clickbait	7.0370	0.3132	
sathum	callact	5.6250	0.3890	
posemot	neutral	3.9167	0.4216	
sathum	junksci	3.6979	0.4903	
negprop	conspiracy	2.3559	0.7982	
posemot	junksci	1.0694	0.6797	
junksci	callact	0.6250	0.8021	
posemot	callact	0.5556	0.8026	
sathum	posemot	-1.1215	0.8466	
neutral	callact	-1.3509	0.8021	
negprop	bias	-2.0515	0.8351	
neutral	junksci	-2.9057	0.5284	
conspiracy	bias	-2.9636	0.7419	
posemot	clickbait	-5.8980	0.5237	
posemot	conspiracy	-9.0435	0.2441	
negemot	hate	-10.5279	0.1275	
sathum	clickbait	-10.9203	0.1030	
posemot	hate	-11.4251	0.0344*	
clickbait	bias	-14.4455	0.1082	
sathum	conspiracy	-15.0136	0.0149*	
junksci	clickbait	-15.3108	0.0672	
posemot	negemot	-16.0275	0.1133	
junksci	conspiracy	-17.2663	0.0152*	
posemot	bias	-18.1047	0.2449	
negprop	negemot	-18.4188	0.0442*	
posemot	negprop	-19.7623	0.2241	
junksci	hate	-20.3315	<.0001*	
neutral	clickbait	-24.5619	0.0003*	
sathum	hate	-24.9049	<.0001*	
neutral	conspiracy	-27.2601	<.0001*	
sathum	negemot	-28.5578	<.0001*	
sathum	bias	-28.9220	0.0019*	
sathum	negprop	-28.9641	0.0025*	
negprop	hate	-30.4659	0.0015*	
neutral	hate	-30.8993	<.0001*	
junksci	bias	-36.3629	0.0087*	
neutral	negemot	-41.2765	<.0001*	
callact	bias	-42.1543	0.0232*	
neutral	bias	-48.2336	<.0001*	
neutral	negprop	-52.5828	<.0001*	

Figure 10: Pairwise comparisons using the Wilcoxon method across the set of features with respect to the agenda score. A *positive* Score Mean Difference with significant p-value implies that the articles with Label 1 are associated with higher agenda scores than articles with Label 2 (** $p < 0.01$, * $p < 0.05$). A *negative* Score Mean Difference with significant p-values implies the opposite. The final column indicates the Score Mean Difference. The agenda score has a bi-modal distribution, as expected in Likert scale type survey responses. Key for feature names - negprop:propaganda, callact:call to action, negemot:negative sentiment, junksci:junk science, hate:hate speech, bias:political bias, clickbait:clickbait, conspiracy:conspiracy theories, neutral:neutral sentiment, sathum:satire, posemot:positive sentiment.

Label	**Example Spans from NEWSAGENDAS**
Clickbait	*Could #RussianHackers have used a cloaking device to hide Wisconsin from Hillary?*
Junk Science	*Apple cider vinegar has so many benefits, but personally one of the reasons I like it best is because of the digestive and metabolism boosting benefits.*
Hate Speech	*They are a race of ugly dwarves, of diminutive stature, with hideous faces, evil beady eyes and stunted small minds.*
Conspiracy Theory	*The case sparked national debate over immigration reform and so-called Sanctuary Cities that shield illegals from deportation, of which San Francisco is one.*
Propaganda	*President Barack Obama made sure to shutter veterans parks in an effort to make the GOP look bad during the shutdown which occurred under his watch.*
Satire	*The former U.S. senator and former Democrat nominee for Vice President was charged with several felonies. Shockingly, felonious narcissism was not one of them.*
Negative Sentiment	*Once again, the party bereft of ideas and principle resorts to emotional obfuscation and accusation to advance their ideological prejudice.*
Neutral Sentiment	*A long lost Viking settlement known as 'Hop' is located in Canada, a prominent archaeologist has revealed.*
Positive Sentiment	*Newspapers, pamphlets and broadsheets provided nourishment to both spark the American Revolution and keep it alive.*
Political Bias	*Although this news may sound surprising, there are valid reasons for blacks to gravitate toward Trump.*
Call to Action	*We need your financial support to help reach those undecided voters, and if you would like to help, you can donate online right here.*

Table 11: Example evidence spans annotated in NEWSAGENDAS.

Class	**Negative Example Classes**
Clickbait	Conspiracy Theory, Hate Speech, Propaganda, Satire, Average
Junk Science	Hate Speech, Propaganda, Satire, Average
Hate Speech	Clickbait, Junk Science, Satire, Average
Conspiracy Theory	Clickbait, Satire, Average
Propaganda	Clickbait, Junk Science, Satire, Average
Satire	Clickbait, Conspiracy Theory, Hate Speech, Junk Science, Propaganda, Average

Table 12: All training articles belong to one of the 7 classes - Clickbait, Junk Science, Hate Speech, Conspiracy, Propaganda, Satire - or are *Average* articles, meaning likely truthful and informative. The class labels are from the FakeNewsCorpus, Proppy Corpus, and the Yang et al. (2017) satire dataset. We omit articles from websites that only have less informative labels such as *bias* or *political*.

GSAC: A Gujarati Sentiment Analysis Corpus from Twitter

Monil Gokani and **Radhika Mamidi**
Language Technologies Research Center (LTRC)
Kohli Center on Intelligent Systems
International Institute of Information Technology, Hyderabad
`monil.gokani@research.iiit.ac.in`
`radhika.mamidi@iiit.ac.in`

Abstract

Sentiment Analysis is an important task for analysing online content across languages for tasks such as content moderation and opinion mining. Though a significant amount of resources are available for Sentiment Analysis in several Indian languages, there do not exist any large-scale, open-access corpora for Gujarati. Our paper presents and describes the Gujarati Sentiment Analysis Corpus (GSAC), which has been sourced from Twitter and manually annotated by native speakers of the language. We describe in detail our collection and annotation processes and conduct extensive experiments on our corpus to provide reliable baselines for future work using our dataset.

1 Introduction

Sentiment Analysis is an NLP task that involves identifying the sentiment or the polarity (such as positive vs negative) of a piece of text (Pang et al., 2008). It has received significant attention in recent years, with ever-increasing internet access and social media usage, even in Indian languages such as Hindi (Akhtar et al., 2016; Patra et al., 2018) and Marathi (Kulkarni et al., 2021a) which are typologically and geographically close to Gujarati. However, there is hardly any work done in Gujarati itself.

Gujarati is a very prominent language in Western India, with over 55 million first-language speakers and a significant presence in the states of Gujarat, Maharashtra, and Rajasthan (Eberhard et al., 2023). It is also the official language of the state of Gujarat. Despite a large online community active on social media and a significant mainstream media presence, there is a lack of large-scale, publicly available resources for sentiment classification (see Section 2).

Hence, we describe a new, gold-standard, manually annotated Gujarati Sentiment Analysis Corpus (GSAC) for monolingual sentiment classification. The dataset is sourced from Twitter and labelled by native speakers. We describe our annotation process and also run extensive experiments on the dataset using feature-based and deep-learning architectures to establish a reliable baseline for GSAC and compare the performances of various model architectures. The dataset is available on GitHub.[1]

2 Related Work

Significant work has been done on coarse-grained and aspect-based sentiment analysis (SA) in various Indian languages. Datasets have been created for SA in Hindi (Akhtar et al., 2016; Patra et al., 2018), Telugu (Mukku and Mamidi, 2017), Marathi (Kulkarni et al., 2021b), Bengali (Islam et al., 2021; Patra et al., 2018) and Tamil (Jenarthanan et al., 2019), and Tamil and Malyalam (Chakravarthi et al., 2021). However, SA in Gujarati has been scarcely explored, and no standard, publicly available dataset exists.

One of the earliest works in SA in Gujarati was by Joshi and Vekariya (2017), who used a POS tag-based feature set for an SVM classifier on a small sample of 40 tweets. Since then, Gohil and Patel (2019) developed and experimented with a Gujarati SentiWordNet to classify tweets, creating a Twitter dataset with 1120 samples. Other approaches included scraping movie-review websites to create a dataset (Shah and Swaminarayan, 2021; Shah et al., 2022a), even translating reviews from English to Gujarati to expand the dataset (Shah and Swaminarayan, 2022; Shah et al., 2022b). Mehta and Rajyagor (2021) attempted classifying a set of 300 poems into nine different emotional categories using machine learning-based approaches. However, none of the datasets used in these experiments have been released to open access, which makes it difficult to reproduce any of these results or compare the performance of new models with

---

[1] https://github.com/MG1800/gsac

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 129–137
July 14, 2023 ©2023 Association for Computational Linguistics

Work(s)	Source	Size	Annotation	Open Access
(Joshi and Vekariya, 2017)	Twitter	40	Manual	No
(Mehta and Rajyagor, 2021)	Poems	300	Manual	No
(Gohil and Patel, 2019)	Twitter	1120	Manual	No
(Shah et al., 2022a), (Shah and Swaminarayan, 2021)	Movie Reviews	500	Manual	No
(Shah and Swaminarayan, 2022), (Shah et al., 2022b)	Movie Reviews (Gujarati + translated from English)	2085	Automated, based on website rating	No
GSAC	**Twitter**	**6575**	**Manual**	**Yes**

Table 1: Comparison of previous datasets on Gujarati Sentiment Analysis with our dataset - GSAC

them.

Gujarati was a part of the set of languages included in the training data for XLM-T (Barbieri et al., 2022), a highly multilingual effort for creating a unified Twitter-based language model for sentiment classification. However, Gujarati was not a part of the monolingual evaluation reported by the authors. Additionally, Gujarati has been included in some research on multilingual lexical level sentiment classification (Zhao and Schütze, 2019; Buechel et al., 2020).

Efforts in dataset creation for Sentiment Analysis have been varied. We mainly focused on Twitter datasets or datasets in Indian languages for reference when deciding our annotation process. Jenarthanan et al. (2019) created a Twitter-based emotion classification dataset in Tamil and English and used a set of emotion words as queries for collecting tweets, an approach that we also use for collecting our data. Mukku and Mamidi (2017) classify sentences from a news corpus into three sentiment categories - positive, negative, and neutral, similar to what we aim for, and hence are a good source of reference for annotation guidelines. We also refer to Muhammad et al. (2022), which is a more recent effort at creating a sentiment classification dataset for resource-poor languages, collecting and annotating a dataset for 4 African languages with multiple human annotators.

Table 1 compares our dataset to the existing SA datasets in Gujarati.

3 Dataset Creation

The dataset was created in two main steps - collecting and sampling the dataset from Twitter to create a subset for annotation and getting the data annotated by native speakers, which included creating the annotation guidelines and training them for the task.

3.1 Collection

We source our data from Twitter, which has a large active user base of Gujarati speakers. We scraped the initial dataset using Twitter API [2], which supports filtering the results for Gujarati using the language tag. We also used the API parameters to exclude retweets and quotes, to reduce the number of duplicates in our dataset. To ensure we had a desirable mix of sentiments in the dataset, the search queries were based on a hand-picked subset of sentiment words [3] based on a machine-translated English sentiment lexicon (Chen and Skiena, 2014). We chose a subset so as to remove words that were either not translated or translated incorrectly in the list, selecting ∼250 words. The start times are varied to ensure the tweets are spread out over time, with the final set having tweets ranging from August 2010 to February 2022. We then preprocessed, filtered, and sampled from this large dataset to generate subsets for each of our annotators to label. The complete process we followed is described below:

1. Create a list of prompts by hand-picking samples from machine-translated sentiment vocabulary.

2. Scrape tweets using these prompts using Twitter API, using the API parameters to ensure collected tweets are in Gujarati script, spread out over several years, and do not include any retweets or quotes.

3. Preprocess these tweets, normalising whitespaces and newlines, lower-casing, and replacing all user mentions and URLs with the tokens `@user` and `<url>` respectively.

---

[2] https://developer.twitter.com/en/docs/twitter-api

[3] https://www.kaggle.com/datasets/rtatman/sentiment-lexicons-for-81-languages

4. Drop any tweets with identical text or fewer than 10 tokens after preprocessing. This step eliminated a significant number of gibberish tweets that were not useful for the task, such as the one shown in row 4 of Figure 1.

5. Randomly sampled 10% of the tweets for each prompt to create a subset of approximately 22,000 samples from the larger set that retained the same distribution as the original set.

6. From this smaller representative subset, we randomly sampled 7,000 tweets for annotation based on the annotation resources available to us.

The statistics for this process are provided in Table 2. We labelled approximately 7000 tweets from the representative set, with the final dataset containing 6,575 tweets after dropping undesirable samples as described in Section 3.2.

Stage	**Count**
Initial set from scraping	320,978
Filtering out duplicates	247,226
Dropping tweets with <10 tokens	226,482
Representative Set after Sampling	22,630
Annotated	6,575

Table 2: Data collection statistics

3.2 Annotation

We first developed the annotation schema and tested it by annotating a small sample of the dataset ourselves. Once the dataset was finalised, we recruited four annotators and trained them over several rounds of labelling and discussion before providing them with independent subsets to annotate.

3.2.1 Annotation Schema

We classified each tweet in our dataset as `positive`, `negative`, or `neutral`. We also gave our annotators an `unfit` tag for tweets that they think cannot be used for the task. We define each of the labels as follows:

- `positive` - Tweets were classified as positive if they expressed a positive sentiment about some subject (a product or a movie, for example) or if they showed support for a subject, such as a person or a policy. Tweets about events inherently associated with positive sentiments (such as reporting a sports team's victory) are also labelled positive.

- `negative` - Tweets that expressed a negative opinion about a subject (such as criticising a policy or an official) were labelled negative. Tweets talking about events with an inherently negative connotation - such as reporting the death of a celebrity or the loss of a sports team, and tweets containing any kind of derogatory remarks or threats towards a subject were also labelled negative.

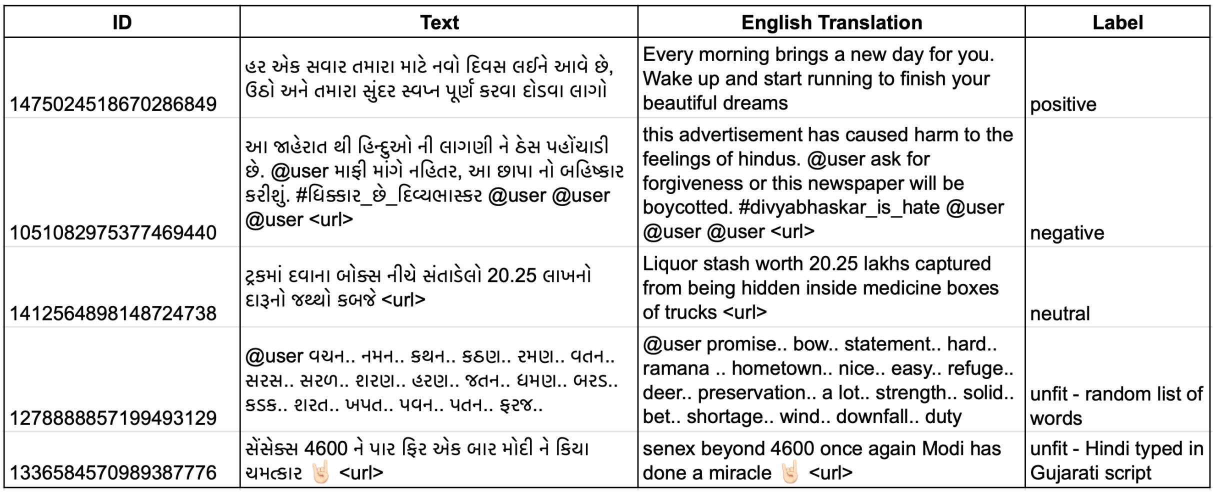
- `neutral` - Tweets were labelled as neutral in two cases - if they contained no sentiment about the subject or if they contained a mix of both positive and negative sentiment about a subject (such as praising one aspect but criticising another of a product).

- `unfit` - Tweets were marked unfit if the annotator could not assign one of the three labels to it. This happened in several cases, such as cases where it was a different language tweet that was typed in Gujarati script, or there was not enough context in the tweet to label it (if it required a media attachment to understand, for example). Any tweets marked unfit by any of the annotators were dropped from the dataset.

Figure 1 illustrates some of the tweets and their labels, along with an approximate English translation of the tweet.

3.2.2 Annotation Process

We manually annotated 7000 samples across four annotators. The annotators were linguistics students who were native speakers of Gujarati, aged between 19 and 23. The annotators were trained for the task over three rounds of annotation on small subsets of 50 tweets each, followed by a session of doubt clarification and discussion after every round. To measure the annotation quality, we calculate inter-annotator agreement using Fleiss' Kappa coefficient (Fleiss, 1971). Over the three rounds of training, it improved from 0.48 to 0.52 and finally to 0.58, which suggests moderately strong agreement. The tweets used for these training rounds were discarded and not included in the final dataset. Each annotator then labelled data in subsets of 500 samples.



ID	Text	English Translation	Label
1475024518670286849	હર એક સવાર તમારા માટે નવો દિવસ લઈને આવે છે, ઉઠો અને તમારા સુંદર સ્વપ્ન પૂર્ણ કરવા દોડવા લાગો	Every morning brings a new day for you. Wake up and start running to finish your beautiful dreams	positive
1051082975377469440	આ જાહેરાત થી હિન્દુઓ ની લાગણી ને ઠેસ પહોંચાડી છે. @user માફી માંગે નહિતર, આ છાપા નો બહિષ્કાર કરીશું. #ધિક્કાર_છે_દિવ્યભાસ્કર @user @user @user <url>	this advertisement has caused harm to the feelings of hindus. @user ask for forgiveness or this newspaper will be boycotted. #divyabhaskar_is_hate @user @user @user <url>	negative
1412564898148724738	ટ્રકમાં દવાના બોક્સ નીચે સંતાડેલો 20.25 લાખનો દારૂનો જથ્થો કબજે <url>	Liquor stash worth 20.25 lakhs captured from being hidden inside medicine boxes of trucks <url>	neutral
1278888857199493129	@user વચન.. નમન.. કથન.. કઠણ.. રમણ.. વતન.. સરસ.. સરળ.. શરણ.. હરણ.. જતન.. ધમણ.. બરડ.. કડક.. શરત.. ખપત.. પવન.. પતન.. ફરજ..	@user promise.. bow.. statement.. hard.. ramana .. hometown.. nice.. easy.. refuge.. deer.. preservation.. a lot.. strength.. solid.. bet.. shortage.. wind.. downfall.. duty	unfit - random list of words
1336584570989387776	સેંસેક્સ 4600 ને પાર ફિર એક બાર મોદી ને કિયા ચમત્કાર 🤘🏻 <url>	senex beyond 4600 once again Modi has done a miracle 🤘🏻 <url>	unfit - Hindi typed in Gujarati script

Figure 1: Some samples from the GSAC dataset

3.3 Statistics

Our final dataset contains a total of 6575 tweets after dropping the tweets labelled unfit. We divide the dataset into training, development, and test sets in a 70:10:20 ratio, respectively. Within the complete dataset, the `neutral` class has the highest representation, comprising about 45.12% of the total dataset, followed by `positive` at 30.05% and finally `negative` at 24.83%. Additional details about the class distribution are reported in Table 3.

The average word count for the combined dataset is 27.77, with a standard deviation of 13.86. The average word count (excluding whitespaces) is 136.07, with a standard deviation of 67.55, as shown in Table 4, which also reports the same values for each class. Figures 2 and 3 illustrate the class-wise and split-wise distribution of word counts in the dataset, respectively.

4 Experiments

We train two sets of models to test how different models perform on our dataset and to set baselines for it. The first set of models consists of feature vector-based models, which we train on two different variants based on different sets of features - Bag-of-Words and TF-IDF. The second set is a set of deep contextualised models, where we fine-tune various transformer-based pre-trained language models for classification on this dataset.

4.1 Feature Vector Models

Split	Positive	Neutral	Negative	Total Count
Train	1374	2100	1128	4602
Dev	201	287	163	651
Test	401	580	341	1322
Total	1976	2967	1632	6575

Table 3: Split-wise Class Distribution of Dataset

Split	Tokens	Characters
Positive	27.86 (11.65)	141.79 (58.92)
Neutral	27.25 (15.52)	132.39 (77.02)
Negative	28.60 (13.07)	135.62 (57.82)
Overall	**27.77 (13.86)**	**136.01 (67.55)**

Table 4: Mean Token and Character Counts for each label (brackets contain standard deviation)

We train five classifiers - Naive Bayes, Logistic Regression, Support Vector Machines, Random Forests, and a Multi-Layer Perceptron - each on two different feature vectors - Bag-of-Words and TF-IDF for a total of 10 models.

Bag-of-Words (BoW) or Count Vectorizer represents a document (in this case, a tweet) as a vector of the counts of each word present in the document. Even though it ignores word order, bag-of-words features can still be useful as feature vectors for tasks such as text classification (McCallum and Nigam, 2001).

TF-IDF (Term Frequency - Inverse Document Frequency) (Spärck Jones, 1972) is a method to represent documents that factors in the relative frequency of a word across documents by calculating a score based on two parameters - term frequency, which is the frequency of a term in the current document, and inverse document frequency - which is based on the frequency of the term across all documents.

The models we train for each of these are:

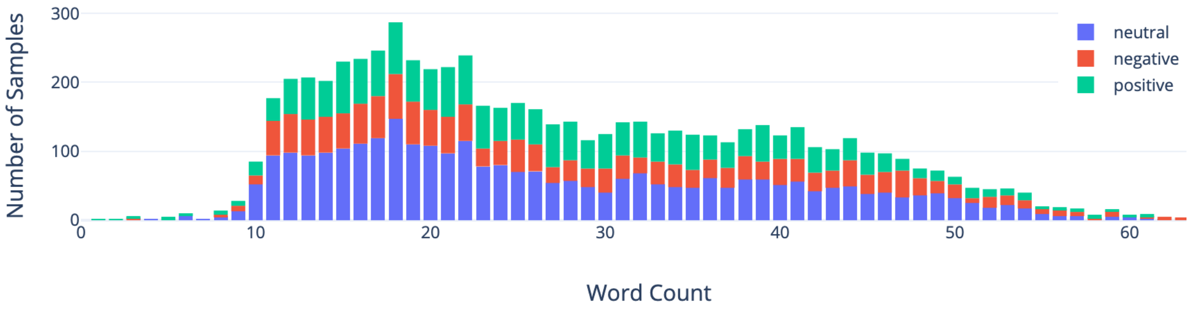


Figure 2: Class-wise distribution of word counts

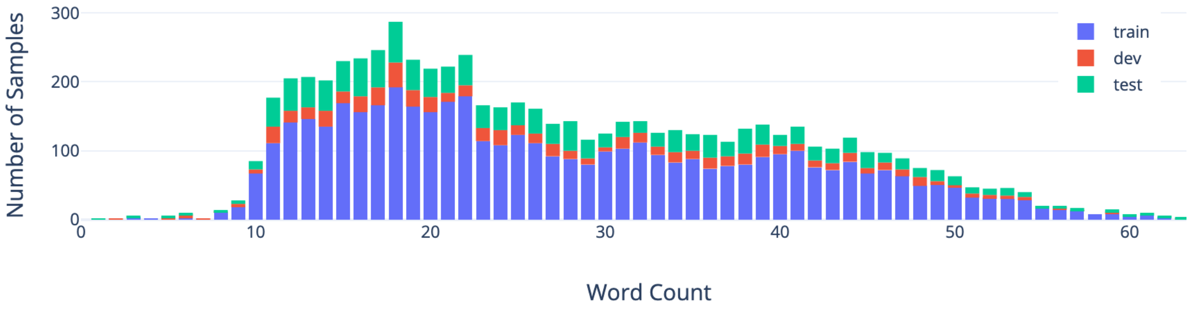


Figure 3: Split-wise distribution of word counts

- **Naive Bayes Classifier** - A Naive Bayes classifier is a simple classifier that estimates the probability of each label under the assumption of input features being conditionally independent, which has been shown to perform well on text classification (McCallum and Nigam, 2001). We train the classifier for 200 epochs or until convergence.

- **Logistic Regression** - Logistic regression (Cox, 1958) is a classification algorithm that estimates a logistic function to calculate the probability of an input feature belonging to a certain class. We train an LR classifier over 100 epochs or convergence using a one-vs-all approach.

- **Support Vector Machine** - A support vector machine (Cortes and Vapnik, 1995) is a classifier that tries to find the hyper-plane that most optimally divides the training data according to the labels. This is also trained using a one-vs-all approach, over 200 maximum epochs.

- **Random Forests** - Random Forests (Breiman, 2001) are a type of ensemble classifier that use a large number of decision trees (set to 100 for our model), each using a subset of the input features and training data, to estimate the most likely label for the given input.

- **Multi-Layer Perceptron** is a simple feed forward neural network (Rosenblatt, 1958; Rumelhart et al., 1986). Our model uses a single 100-dimension hidden layer, with a ReLU activation, for 300 maximum epochs.

We use the scikit-learn python library (Pedregosa et al., 2011) to create feature vectors from the text and train and test this set of models.

4.2 Deep Contextualised Models

Multilingual transformer-based language models trained on multiple languages such as BERT (Devlin et al., 2018) and RoBERTa (Conneau et al., 2019) have been shown to perform well on downstream tasks (Pires et al., 2019). We fine-tune the following language models on our dataset:

- **Multilingual BERT** - mBERT is a multilingual version of BERT (Devlin et al., 2018), and is a language model trained on the top 100 languages with the largest Wikipedia corpora, which includes Gujarati. We use the `bert_base_multilingual_uncased` version of BERT.

- **XLM-RoBERTa** - XLM-RoBERTa is a multilingual version of RoBERTa (Conneau et al., 2019), which is itself a more optimised version of BERT, trained on a larger dataset, and

Model	Precision	Recall	Accuracy	Weighted F1	Macro F1
Bag of Words					
Naive Bayes	0.59	**0.58**	**0.58**	**0.57**	**0.56**
Logistic Regression	0.55	0.55	0.55	0.55	0.54
SVM	0.55	0.52	0.52	0.49	0.46
Random Forests	**0.61**	**0.58**	**0.58**	0.55	0.53
MLP	0.52	0.52	0.52	0.52	0.51
TF-IDF					
Naive Bayes	**0.66**	0.52	0.52	0.43	0.38
Logistic Regression	0.58	**0.57**	**0.57**	**0.56**	**0.55**
SVM	0.57	0.56	0.56	0.54	0.53
Random Forests	0.59	0.55	0.52	0.50	0.50
MLP	0.50	0.50	0.50	0.50	0.49
Pretrained LMs					
mBERT	0.38	0.51	0.51	0.43	0.38
XLM-RoBERTa	0.41	0.52	0.52	0.43	0.39
XLM-T	0.64	0.62	0.63	0.64	0.63
GujaratiBERT	0.64	0.64	0.64	0.64	0.64
IndicBERT	**0.65**	**0.67**	**0.66**	**0.66**	**0.66**

Table 5: Results of the various models on the test set. **Bold** indicates best score for each set of models. **Underline** indicates best score across all models.

a modified training task. XLM-RoBERTa also includes Gujarati as a part of its training set. We fine-tune the `xlm-roberta-base` variant of the model.

- **XLM-T** - XLM-T (Buechel et al., 2020)is a variant of XLM-RoBERTa that was trained on a Twitter dataset consisting of 198M tweets in a large set of languages, including over 10,000 samples in Gujarati. It was further finetuned for Sentiment Classification on a set of 8 languages, which included Hindi, which is closely related typologically to Gujarati. We further fine-tune the `twitter-xlm-roberta-base-sentiment` variant of the model on HuggingFace.

- **GujaratiBERT** - GujaratiBERT (Joshi, 2022) is an mBERT (base variant) model that has been fine-tuned for Gujarati using publicly available monolingual Gujarati corpora. Since it is specifically fine-tuned for Gujarati, we expected it to perform better than mBERT and XLM-RoBERTa.

- **IndicBERT** - IndicBERT (Kakwani et al., 2020) is an ALBERT (Lan et al., 2019) model pre-trained on a combined corpus of 12 different Indian languages (including Gujarati), which has been shown to achieve state-of-the-art performance on multiple downstream tasks in several Indian languages on the IndicGLUE benchmark (Kakwani et al., 2020), including sentiment analysis in Hindi (Akhtar et al., 2016) and Telugu (Mukku and Mamidi, 2017). We fine-tune this model for classification on our dataset.

All of our transformer models are trained for 5 epochs, with a learning rate of 4e-5 and batch size of 8. We set up our training and testing scripts using the simpletransformers (Rajapakse, 2019) library, which is based on the transformers library from HugggingFace (Wolf et al., 2020).

5 Results

We report the detailed results for each model in Table 5. We make a few observations from observing the weighted and macro F1 scores for each model:

- We observe that GujaratiBERT and IndicBERT achieve the best performance compared to all other models. This could be because compared to the rest of the pretrained language models, these two models have been trained on a significantly higher amount of Gujarati data (during pretraining for IndicBERT, and during fine-tuning for GujaratiBERT).

- mBERT and XLM-RoBERTa perform very poorly compared to other pretrained language models. This could be because they are trained on a very large set of languages, due to which Gujarati might not have sufficient representation in the corpus and the model vocabulary causing it to underperform.
- XLM-T contained only ∼10,000 samples in Gujarati out of a total ∼198M samples in its training data. However, it still achieves comparable performance to GujaratiBERT and IndicBERT. This may be because the training data for XLM-T comes exclusively from the same domain as our dataset (Twitter), which suggests pretraining or fine-tuning models on similar domain data in multiple languages can help improve model performance in low-resource languages.
- Despite not achieving the same performance as XLM-T, GujaratiBERT, or IndicBERT, the Naive Bayes model using TF-IDF features achieves the highest precision out of all the models trained. Other statistical models (such as Random Forests and Naive Bayes on both feature sets) also achieve reasonably high average precision (>= 0.59) while taking significantly less computational resources and time.

6 Conclusion and Future Work

In this paper, we present the Gujarati Sentiment Analysis Corpus (GSAC), which contains over 6500 manually annotated tweets. To the best of our knowledge, it is the first significant publicly available corpus for this task in Gujarati. We also present our annotation schema and conduct extensive experimentation to establish baselines for this new dataset. We find that pre-trained language models that included Gujarati as a part of pre-training or fine-tuning achieve better performance on this dataset compared to other models, with IndicBERT achieving the best weighted and macro F1 scores. As a part of future work, we plan to explore methods to extend this dataset automatically by using this dataset as a seed dataset to label additional data (such as by bootstrapping) or by exploring other avenues of acquiring data, such as via machine translation of existing datasets in other languages such as English or Hindi.

7 Ethical Consideration

Sentiments in a dataset sourced from social media platforms can be susceptible to inherent bias due to public opinion being biased in favour of or against certain subjects, depending on external factors like demographics. During the collection and annotation process for our dataset, we switched our collection strategy from querying tweets for particular topics (events) during the initial stages to querying them using a sentiment lexicon because we observed that the topics we queried were frequently heavily biased towards either positive or negative sentiments. The privacy of platform users is another concern that is raised when collecting data from social media. To ensure that no identifying details about any Twitter user were presented to our annotators, we removed any identifying characteristics such as user mentions and URLs from the tweets, as well as the original Tweet IDs and used internally generated IDs for the annotation process. We also only release the Tweet IDs and corresponding labels in our dataset in compliance with Twitter's data-sharing policy.

References

Md Shad Akhtar, Ayush Kumar, Asif Ekbal, and Pushpak Bhattacharyya. 2016. A hybrid deep learning architecture for sentiment analysis. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 482–493, Osaka, Japan. The COLING 2016 Organizing Committee.

Francesco Barbieri, Luis Espinosa Anke, and Jose Camacho-Collados. 2022. XLM-T: Multilingual language models in Twitter for sentiment analysis and beyond. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 258–266, Marseille, France. European Language Resources Association.

Leo Breiman. 2001. *Machine Learning*, 45(1):5–32.

Sven Buechel, Susanna Rücker, and Udo Hahn. 2020. Learning and evaluating emotion lexicons for 91 languages. *CoRR*, abs/2005.05672.

Bharathi Raja Chakravarthi, KP Soman, Rahul Ponnusamy, Prasanna Kumar Kumaresan, Kingston Pal Thamburaj, John P McCrae, et al. 2021. Dravidianmultimodality: A dataset for multi-modal sentiment analysis in tamil and malayalam. *arXiv preprint arXiv:2106.04853*.

Yanqing Chen and Steven Skiena. 2014. Building sentiment lexicons for all major languages. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 383–389, Baltimore, Maryland. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

David R Cox. 1958. The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2):215–232.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

David M. Eberhard, Gary F. Simons, and Charles D. Fennig. 2023. *Ethnologue: Languages of the World*, 26 edition. SIL International, Dallas, Texas.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychol. Bull.*, 76(5):378–382.

Lata Gohil and Dharmendra Patel. 2019. A sentiment analysis of gujarati text using gujarati senti word net. *International Journal of Innovative Technology and Exploring Engineering*.

Khondoker Ittehadul Islam, Sudipta Kar, Md Saiful Islam, and Mohammad Ruhul Amin. 2021. SentNoB: A dataset for analysing sentiment on noisy Bangla texts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3265–3271, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Rajenthiran Jenarthanan, Yasas Senarath, and Uthayasanker Thayasivam. 2019. Actsea: Annotated corpus for tamil sinhala emotion analysis. In *2019 Moratuwa Engineering Research Conference (MERCon)*, pages 49–53.

Raviraj Joshi. 2022. L3cube-hindbert and devbert: Pre-trained bert transformer models for devanagari based hindi and marathi languages. *arXiv preprint arXiv:2211.11418*.

Vrunda C Joshi and Vipul M Vekariya. 2017. An approach to sentiment analysis on gujarati tweets. *Advances in Computational Sciences and Technology*, 10(5):1487–1493.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLPSuite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages. In *Findings of EMNLP*.

Atharva Kulkarni, Meet Mandhane, Manali Likhitkar, Gayatri Kshirsagar, and Raviraj Joshi. 2021a. L3CubeMahaSent: A Marathi tweet-based sentiment analysis dataset. In *Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 213–220, Online. Association for Computational Linguistics.

Atharva Kulkarni, Meet Mandhane, Manali Likhitkar, Gayatri Kshirsagar, and Raviraj Joshi. 2021b. L3cubemahasent: A marathi tweet-based sentiment analysis dataset. *CoRR*, abs/2103.11408.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR*, abs/1909.11942.

Andrew McCallum and Kamal Nigam. 2001. A comparison of event models for naive bayes text classification. *Work Learn Text Categ*, 752.

Bhavin Mehta and Bhargav Rajyagor. 2021. Gujarati poetry classification based on emotions using deep learning.

Shamsuddeen Hassan Muhammad, David Ifeoluwa Adelani, Sebastian Ruder, Ibrahim Said Ahmad, Idris Abdulmumin, Bello Shehu Bello, Monojit Choudhury, Chris Chinenye Emezue, Saheed Abdullahi Salahudeen, Aremu Anuoluwapo, Alípio Jeorge, and Pavel Brazdil. 2022. Naijasenti: A nigerian twitter sentiment corpus for multilingual sentiment analysis. *CoRR*, abs/2201.08277.

Sandeep Sricharan Mukku and Radhika Mamidi. 2017. ACTSA: Annotated corpus for Telugu sentiment analysis. In *Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems*, pages 54–58, Copenhagen, Denmark. Association for Computational Linguistics.

Bo Pang, Lillian Lee, et al. 2008. Opinion mining and sentiment analysis. *Foundations and Trends® in information retrieval*, 2(1–2):1–135.

Braja Gopal Patra, Dipankar Das, and Amitava Das. 2018. Sentiment analysis of code-mixed indian languages: An overview of sail$_code - mixedsharedtask@icon - 2017$.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

T. C. Rajapakse. 2019. Simple transformers. `https://github.com/ThilinaRajapakse/simpletransformers`.

F Rosenblatt. 1958. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol. Rev.*, 65(6):386–408.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1986. Learning internal representations by error propagation.

P. Shah, P. Swaminarayan, and Maitri Patel. 2022a. Sentiment analysis on film review in gujarati language using machine learning. *International Journal of Electrical and Computer Engineering (IJECE)*.

Parita Shah and Priya Swaminarayan. 2022. Machine learning-based sentiment analysis of gujarati reviews. *International Journal of Data Analysis Techniques and Strategies*.

Parita Shah, Priya Swaminarayan, and Maitri Patel. 2022b. Sentiment analysis on film review in gujarati language using machine learning. *International Journal of Electrical and Computer Engineering*, 12(1):1030.

Parita Vishal Shah and Priya Swaminarayan. 2021. Lexicon-based sentiment analysis on movie review in the gujarati language. *Int. J. Inf. Technol. Commun. Convergence*.

Karen Spärck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Mengjie Zhao and Hinrich Schütze. 2019. A multilingual BPE embedding space for universal sentiment lexicon induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3506–3517, Florence, Italy. Association for Computational Linguistics.

A Dataset for Explainable Sentiment Analysis in the German Automotive Industry

Andrea Zielinski Calvin Spolwind
Anna Grimm Henning Kroll
Fraunhofer ISI
Breslauer Str. 48
76139 Karlsruhe
`andrea.zielinski@isi.fraunhofer.de`

Abstract

While deep learning models have greatly improved the performance of many tasks related to sentiment analysis and classification, they are often criticized for being untrustworthy due to their black-box nature. As a result, numerous explainability techniques have been proposed to better understand the model predictions and to improve the deep learning models. In this work, we introduce InfoBarometer, the first benchmark for examining interpretable methods related to sentiment analysis in the German automotive sector based on online news. Each news article in our dataset is annotated with respect to the overall sentiment (i.e., *positive, negative and neutral*), the target of the sentiment (focusing on *innovation-related topics such as e.g. electromobility*) and the rationales, i.e., textual explanations for the sentiment label that can be leveraged during both training and evaluation. For this research, we compare different state-of-the-art approaches to perform sentiment analysis and observe that even models that perform very well in classification do not score high on explainability metrics like model plausibility and faithfulness. We calculated the polarity scores for the best method BERT and got a macro F1-score of 73.8. Moreover, we evaluated different interpretability algorithms (LIME, SHAP, Integrated Gradients, Saliency) based on explicitly marked rationales by human annotators quantitatively and qualitatively. Our experiments demonstrate that the textual explanations often do not agree with human interpretations, and rarely help to justify the models decision. However, global features provide useful insights to help uncover spurious features in the model and biases within the dataset. We intend to make our dataset public for other researchers.

1 Introduction

There is a wealth of economic (online) news so that even specialists find it challenging to analyze all information in a timely manner. Therefore, methods that seek to automatically identify the sentiment in textual data for calculating economic indices or trends have become prominent in recent years (Seki et al., 2022; Kalamara et al., 2022; Levenberg et al., 2014; Weiss and Nemeczek, 2022).

In particular, deep learning based approaches have yielded high performance (Poria et al., 2020; Rojas-Barahona, 2016; Hartmann et al., 2022), and their results even seem to correlate with survey-based indicators (Marszal et al., 2022; Katayama et al., 2019).

However, sentiment analysis[1] is challenging due to the domain-specific language and unavailability of labeled datasets needed for training or fine-tuning neural models (Araci, 2019; Mishev et al., 2020). It is also a complex task, as a document may contain multiple targets whose sentiments may be opposite. For example, in the sentence *Um gute Produkte und Leistungen zu liefern, hat die Firma hohe Investitionen getätigt. (In order to deliver good products and services, the company has made high investments.)*, the author expresses a positive sentiment towards *products and services* using the opinion words *gute (good)* and a negative sentiment towards *Investitionen (investments)* using *hohe (high)*. Apart from mixed sentiment, another challenge is that depending on the context, the sentiment may also shift to the opposite, e.g., *hoher Komfort/hoher Verlust (high comfort vs. high loss)*.

Since model interpretability is a critical requirement for many downstream applications, recently explainable artificial intelligence (XAI) techniques that make models explainable have been proposed (Danilevsky et al., 2020; Madsen et al., 2021), and also been adopted for sentiment analysis in general (El Zini et al., 2022) or with respect to financial news (Gite et al., 2021; Xing et al., 2020). In line with Arrieta et al. (2020), we consider that an AI

---

[1] In our work, we focus on document-level sentiment analysis, i.e. the goal to infer the overall opinion of a news article, which is assumed to convey a unique opinion towards a topic.

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 138–148
July 14, 2023 ©2023 Association for Computational Linguistics

system is explainable, if the task model is intrinsically interpretable or if it is complemented with an interpretable and faithful explanation. In this work, we focus on transparent inference through the generation of a post-hoc explanation for the final output prediction (Thayaparan et al., 2020).

It is well known that machine learning techniques suffer from an inadequate generalization capacity on real-world datasets, which are often scarce and imbalanced. While transfer learning and regularization techniques have been proposed to solve this issue (Pan and Yang, 2010; Srivastava et al., 2014), overfitting the training data is still a common problem, especially for non-English texts. As has been stated by various authors, explainable modelling can contribute to trustworthy AI systems, that go beyond quantitative performance metrics such as accuracy (Danilevsky et al., 2020; Wallace et al., 2020).

To this aim, we build up a corpus of German news articles that report recent innovations in the automotive field, which includes class labels for topics such as *e-mobility, autonomous driving, Diesel, hydrogen, synthetic fuel, and misc*, sentiment labels *i.e., positive, neutral, negative*, and human rationales for studying trustworthiness, inspired by recent work in this field (Zaidan and Eisner, 2008; Wang et al., 2021, 2022). We conduct experiments building on state-of-the art machine learning models for sentiment analysis, as well as interpretation methods (Molnar, 2022) using established evaluation metrics for interpretability (DeYoung et al., 2020; Atanasova et al., 2020). Human explanation for sentiment classification is essential for high-impact settings such as financial analysis where human rationales are required to support specialists' decisions. Basic research questions are

- **RQ1** How well can we automatically classify news articles as expressing a *positive, negative or neutral* opinion at the document-level and which approach gives the best performance. How well is human IAA for our corpus.

- **RQ2** Which sentiment detection and XAI method delivers interpretable results, highlighting words or phrases that have lead to the sentiment class. Is this also reflected by human rationales on positive or negative aspects?

Our main contributions are as follows:

- We present a benchmark for rationalized predictions, including baseline models and results. We quantitatively compare SVM, CNN, LSTM and BERT models in the context of sentiment analysis by performing an evaluation on our manually annotated corpus. Using local and global explanation methods, we are able to retrieve the tokens that are most indicative of the sentiment class label.

2 InfoBarometer Dataset

In this section, we provide the dataset collection strategy, the annotation procedure and statistics on our dataset.

2.1 Data Collection

We scraped German news articles related to recent innovative trends in the automotive sector for the time period Feb-2022 to Dec-2022 from online news providers[2], filtering the results by selecting innovation-related keywords, and then applying a precision-oriented topic classifier.

We keep a minimal layout with title and paragraphs, eliminating links and pictures in the news articles.

2.2 Annotation Procedure

The dataset was annotated using *tagtog*[3], a web-based tool, which was configured appropriately for our annotation task. Each news article contains three types of annotations: the sentiment polarity, the topic category, and the human rationales.

Regarding **sentiment polarity**, each news article is assigned one of the following polarity labels: *positive, negative, neutral* which reflects the overall sentiment label about a certain **topic category** that is prevalent in a given news article, and chosen from a predefined inventory of categories, i.e. *autonomous driving*, *electromobility*, *hydrogen*, *Diesel*, *Synfuel* and *misc* (see Appendix A). We further ask the annotators to highlight **rationales** as text spans, that could justify the final polarity annotation.

Annotation guidelines have been set up that clearly explain the goal of the annotation task, how to annotate tokens or spans and also include a definition for each topic category, following best-practice recommendations (Wiegreffe and Marasović, 2021).

---

[2] www.automobil-industrie-vogel.de, www.automobilwoche.de

[3] https://www.tagtog.com/

We provided multiple examples with topic and polarity classification as well as rationale annotations to help the annotators understand the task.

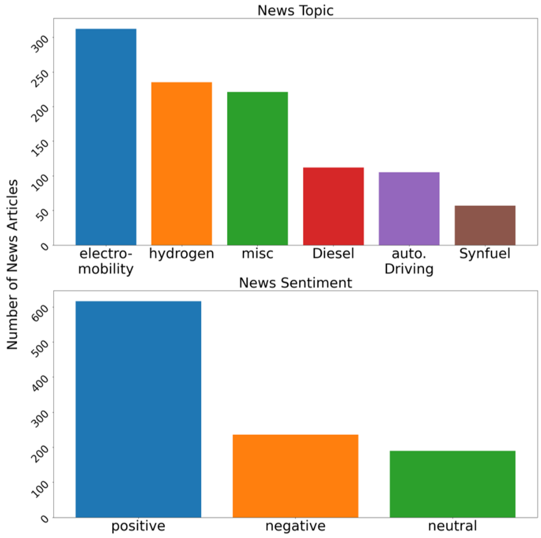


Figure 1: Dataset Statistics

Each news article was annotated by 3 annotators, experts in the automotive domain. In addition a master annotator consolidated annotations and decided on the final label and rationales, when annotators disagreed. Since classifying implicit sentiments is a challenging task, which demands expert knowledge and requires interpretation, we conducted an inter-annotator agreement study to determine whether our annotation guidelines were properly defined and resulted in consistent and reproducible annotations. To measure the interannotator agreement of the manual annotation (IAA), we calculate the overall F1-measures of the annotations, treating the master annotation as the gold standard, resulting in an average F1 score of 0.829 and 0.774 for text classification and sentiment annotation, respectively, which indicates a high agreement of the labeled data. The IAA is depicted in Table 1.

IAA	Topic Label f1	Sentiment f1
annotator1	0.8249	0.7989
annotator2	0.7716	0.6821
annotator3	0.8905	0.8396

Table 1: Inter-Annotator Agreement of the Sentiment and Topic Classification Task

2.3 Dataset Statistics

The dataset is composed of 1,039 news articles from the web. As can be seen in Figure 1, the extent to which German news media cover innovation-related technologies in the automotive sector differs a lot in terms of frequency of certain topics. Looking at the sentiment distribution, we see that they mostly receive positive coverage in German news media with 59.12%, while 22.65% are negative and 18.23% of all news is neutral. Given these results, we infer that news on innovation and new technologies are indeed overall positive for the German automotive industry. While sentiment polarity annotations express overall sentiments (positive/negative/neutral) in the news article, it might nonetheless include different opinionated expressions with different polarities, positive or negative, indicated by spans of rationales on each news item.

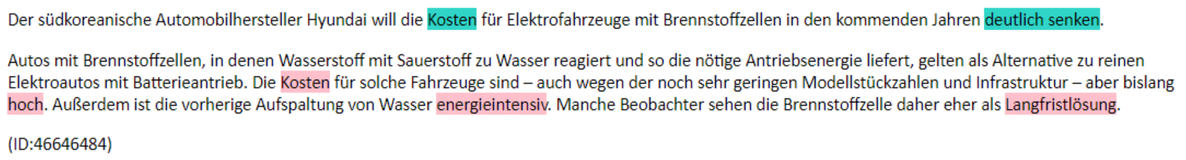
Figure 2 shows all rationales that have been selected for a specific news article, grounding the overall positive sentiment annotation for the whole article in the positive evidences *Steigerung (increase), gerettet (saved), mehr als verdoppelt (more than doubled)*. The aim was to annotate the news in an exhaustive way, which in this example includes also negative evidences (e.g. *Mangel an Halbleitern (lack of semiconductors), verfehlt (missed), Produktion gebremst (hampered production)*.

The InfoBarometer dataset has been split into training, validation, and testing set. The training set contains 833 news items, while the dev and test set contain 104 and 102 items, respectively. The average length for each news article is 226.21 tokens, consisting of 13.23 sentences on average.

For this dataset, the number of rationales per instance is 6.25 (9.97 tokens) on average, which varies slightly by sentiment class, also when taking the average percentage of rationales to input sequence length into account (see Table 2). Note that there is no perfect correlation between the length of an article and the length of the rationale based on our Benchmark Corpus (see Fig. 3). Only the test set was used in the plausibility evaluation.

2.4 Related Work

Many datasets with human-annotated rationales have been published for interpretability evaluation, in particular highlight-based rationales (DeYoung et al., 2020). For the task of sentiment analysis, the Movie Reviews Dataset (Zaidan et al., 2007) has lately been extended (DeYoung et al., 2020) to



Der südkoreanische Automobilhersteller Hyundai will die Kosten für Elektrofahrzeuge mit Brennstoffzellen in den kommenden Jahren deutlich senken.

Autos mit Brennstoffzellen, in denen Wasserstoff mit Sauerstoff zu Wasser reagiert und so die nötige Antriebsenergie liefert, gelten als Alternative zu reinen Elektroautos mit Batterieantrieb. Die Kosten für solche Fahrzeuge sind – auch wegen der noch sehr geringen Modellstückzahlen und Infrastruktur – aber bislang hoch. Außerdem ist die vorherige Aufspaltung von Wasser energieintensiv. Manche Beobachter sehen die Brennstoffzelle daher eher als Langfristlösung.

(ID:46646484)

Figure 2: Annotation of a German news article. Human Rationales are highlighted in blue (positive) and red (negative).

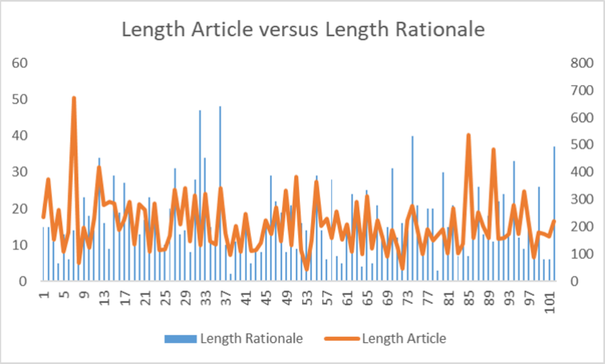


Figure 3: Dataset Statistics: The correlation between the length of an article and the length of the rationale based on our Benchmark Corpus.

Sentiment	Rationales	Len. Rationale	Len. Ratio
negative	6.75	16.46	7.27
neutral	5.52	14.90	6.59
positive	6.22	17.79	7.86

Table 2: The Rationales column presents the average number of rationales, the Length Rationale the average number of tokens per instance, while the Length Ratio column presents the average ratio of rationale to input sequence length measured in tokens.

build a comprehensive evaluation benchmark for interpretability (Zaidan and Eisner, 2008).

Wang et al. (2021) propose a novel evaluation dataset for sentence-level sentiment analysis for English. They provide highlight-based rationales to evaluate the robustness and interpretability (i.e., faithfulness and plausibility) of various algorithms (Lime, SHAP, attention) and deep learning frameworks, including LSTM and SKEP (Tian et al., 2020).

In comparison to these datasets, the InfoBarometer dataset also contains span-level rationale annotations, however, we annotated three polarity classes instead of 2, including the neutral class which either denotes the lack of sentiment towards a topic or may consist of an equal amount of positive and negative evidence in the news article. The included rationale annotations are comprehensive, since the annotators were asked to mark all text supporting the polarity label and we aggregated the rationales from multiple annotators if they agreed on the same class. In addition, approx. 20% of all article contain mixed sentiments with evidence for both the positive and negative polarity class.

3 Methodology

Due to the small size of our dataset, we think that there is high need for explainability, regarding model interpretability. Through our experiments, we like to investigate if explainability techniques can uncover biases within the dataset and check the trustworthiness of the ML models trained on the InfoBarometer dataset.

3.1 Classification Models

We consider three model families of text encoders with increasing complexity: a support vector machine (SVM) with a linear kernel (Pang et al., 2008), a CNN (Kim, 2014), LSTM (Hochreiter and Schmidhuber, 1997) and a pre-trained BERT model (Devlin et al., 2018)[4]. To build the sentiment classifier, we fine-tune BERT on our dataset to encode domain specific semantics and augment it with a prediction task to encode sentiment and topic information. In the CNN and LSTM models, the word inputs are initialized to 300-dimensional GloVe embeddings (Pennington et al., 2014)[5]. We tune all parameter using the validation set and use the best model for testing. While recently contextual language models like BERT performed best on a variety of benchmark datasets for document-sentiment analysis, also CNNs and RNNs have been applied successfully (Poria et al., 2020), as well as sequence models (Li et al., 2016), because they can encode long-range dependencies in the word sequence, which is fundamental to model

---

[4] German BERT model is from Huggingface https://huggingface.co/deepset/gbert-base

[5] GLOVE embeddings are from Deepset https://www.deepset.ai/german-word-embeddings

negation and polarity shifts. On the movie dataset[6] (Zaidan and Eisner, 2008), CNNs based on pre-trained embeddings showed highest performance in terms of accuracy, outperforming RNNs and LSTMs, because they are able to learn contextual semantic features that are relevant for the sentiment prediction task. Our baseline models are:

Convolutional Neural Networks: CNNs (Kim, 2014) are specially powerful in exploiting the local correlation and pattern of the data by their feature maps. Since parallelization is possible, they are more efficient than LSTMs and yield a good performance for classification tasks with little fine tuning (Nedjah et al., 2022).

Long-short time memory : LSTM (Schmidhuber et al., 1997) is a popular recurrent neural network architecture for modeling sequential data, and can capture long term dependencies. They have the characteristics of selectivity and memory cells and solve the gradient vanishing problem.

Bidirectional Encoder Representations from Transformers (BERT) by (Devlin et al., 2018), enriched with the deep bidirectional word representations released by HuggingFace (Wolf et al., 2019). Key aspects of the BERT model include multi-head self-attention as well as a Transformer layer.

See Appendix A.2 for our hyperparameter settings.

3.2 Explainability Models

We focus on post-hoc feature attribution explanation methods, probing the model parameters and/or input-output pairs of an already trained black-box model. We use Captum[7], an open source library for model interpretability built on PyTorch for our experiments, including explanation methods that are *gradient-based*, i.e. InputXGradient (Shrikumar et al., 2016), Saliency (Simonyan et al., 2013) *perturbation-based*, i.e. Shapley Value Sampling (Shapley, 1953; Castro et al., 2009), and Lime (Ribeiro et al., 2016) as a *model simplification* method. Saliency maps are used to visualize which parts of the input are responsible for the prediction. In the case of gradient-based methods, we measure the importance of a feature using the derivative with respect to that feature. We apply the L2 norm to obtain the magnitude of a gradient vector, which becomes a saliency of each token. As the output, we take the loss with the top prediction as the ground truth class (Han et al., 2020).

4 Experimental Setup

We execute experiments on topic classification and sentiment analysis for four different ML approaches. The model predictions as well as the XAI models' explanations are compared to ground truth data that has been annotated by domain experts. For an overview of the proposed approach see Figure 4.

4.1 Classification Results

Following prior works, we conduct experiments with all described sentiment classification models.

Sentiment Model	Macro F1	Accuracy
SVM_linear	0.5320	0.6476
CNN with GLOVE	0.3824	0.6095
LSTM with GLOVE	0.4013	0.5048
Fine-tuned BERT	0.7376	0.7809

Table 3: Accuracy of sentiment analysis models (in terms of classification accuracy and macro f1), evaluated on the news datasets with 1039 articles devided into train/dev/test.

Topic Classifier Model	Macro F1	Accuracy
SVM_linear	0.3600	0.4857
CNN with GLOVE	0.4981	0.6381
LSTM with GLOVE	0.3248	0.5143
Fine-tuned BERT	0.7904	0.8190

Table 4: Accuracy of the topic classification models (in terms of classification accuracy and macro f1), evaluated on the news datasets with 1039 articles devided into train/dev/test.

4.2 Computational Efficiency

We also compare the computational efficiency of our ML models and XAI techniques (see Section 5.3) that are critical in a setting which requires timely decision support. We recorded the computational time to generate sentiment and class predictions on a computer cluster with 2 AMD EPYC 7742 64-Core Processors 2.25 GHz, 192 GB RAM, x64 NVIDIA A100-PCIE 40GB.

The computational time for classifying our test dataset is shown in Table 8. The speed for testing is relatively low compared to the time for training the model, so that all of them can be used in a real-time interactive system.

---

[6] https://www.tensorflow.org/datasets/catalog/movie_rationales

[7] https://captum.ai/

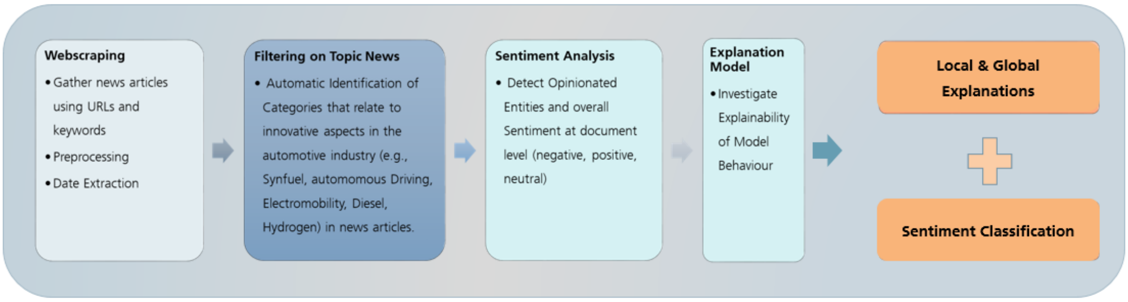

Webscraping
• Gather news articles using URLs and keywords
• Preprocessing
• Date Extraction
Filtering on Topic News
• Automatic Identification of Categories that relate to innovative aspects in the automotive industry (e.g., Synfuel, automomous Driving, Electromobility, Diesel, Hydrogen) in news articles.
Sentiment Analysis
• Detect Opinionated Entities and overall Sentiment at document level (negative, positive, neutral)
Explanation Model
• Investigate Explainability of Model Behaviour
Local & Global Explanations
Sentiment Classification

Figure 4: Overview of the proposed approach

Topic Classifier Model	Train sec	Test sec
CNN with GLOVE	3228.06	27.85
LSTM with GLOVE	703.44	22.83
Fine-tuned BERT	1062.33	21.54

Table 5: Computational efficiency. We report the mean speed in seconds for topic and sentiment classification, for training on train/dev and testing on the test dataset

4.3 Explainability Results

Global Explanation We use SHAP to compute the global features that influence the overall performance of our ML models (Lundberg and Lee, 2017). A key advantage of SHAP is that feature contributions are all expressed in terms of the outcome variable, providing a same scale to compare the importance of each feature against each other. Figure 5 shows what features are important to the model overall. Top salient features in our test set for the negative and positive class are *stop, decrease, bad, crises, expensive* and *preferred, solve, build up, possible, modern, funding*, respectively.

Local Explanation An example of a local explanation from our dataset is shown in Figure 6. We can see that the sentence has been predicted correctly to be positive. However, our explainability methods partly disagree on the interpretation of the same prediction made by the same BERT model. XAI methods depict either *investiert (invest), synthetisch (synthetic)* or *Bundesregierung (government)* as most salient feature that contribute to a positive sentiment assessment.

We observe that the model is relying on spurious features, like *electric* and *autonomous*. Furthermore, any bias in the data, e.g. mentions of nationalities like *Chinese*, need to be eliminated before deployment.

4.4 Evaluation of Explainability

4.4.1 Metrics

Plausibility: In the context of this work, we generate token-level explanations, selecting a list of the top k most salient tokens. For each instance, the model must generate an explanation defined as a subset of zero or more tokens from the instance. The longer the instance, the more explanation tokens are selected. We use IOU (Intersection-Over-Union) F1-score and Token F1 score to measure plausibility (DeYoung et al., 2020). We compute the score at the token level and do not consider continuous sub-strings (spans), since this metrics is considered too imprecise (Wang et al., 2022).

Faithfulness: For debugging, it is important to produce explanations that are faithful, i.e., accurately reflect the features considered important by the model (Jacovi and Goldberg, 2020). The XAI method should faithfully reveal information about the model's inner working. A common approach is to iteratively mask salient features from the input and measure the average drop in the model's performance (DeYoung et al., 2020). In this work, we follow the approach of Atanasova et al. (2020), which relies on producing several perturbations by masking [0, 10, 20,.., 100%] of the input tokens in order of decreasing saliency, and then computing the area under the threshold-performance.

5 Results - Performance Comparison for XAI Methods

We compare LIME, Saliency, InputXGradient and SHAP with respect to the evaluation criteria plausibility (human agreement), faithfulness and runtime complexity.

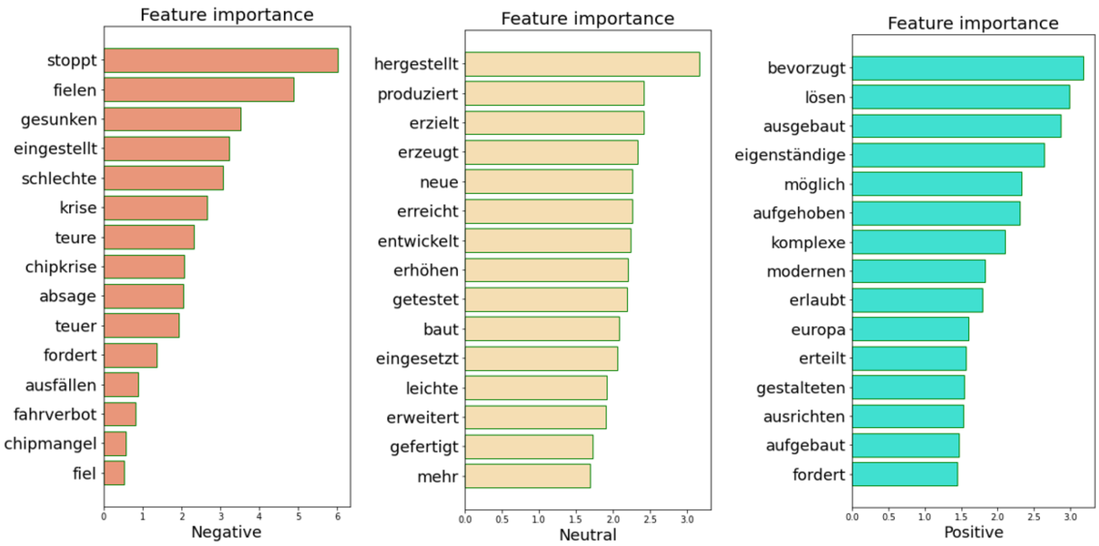


Figure 5: Global Feature Importance Score per Class generated by SHAP and the BERT Sentiment Classifier.

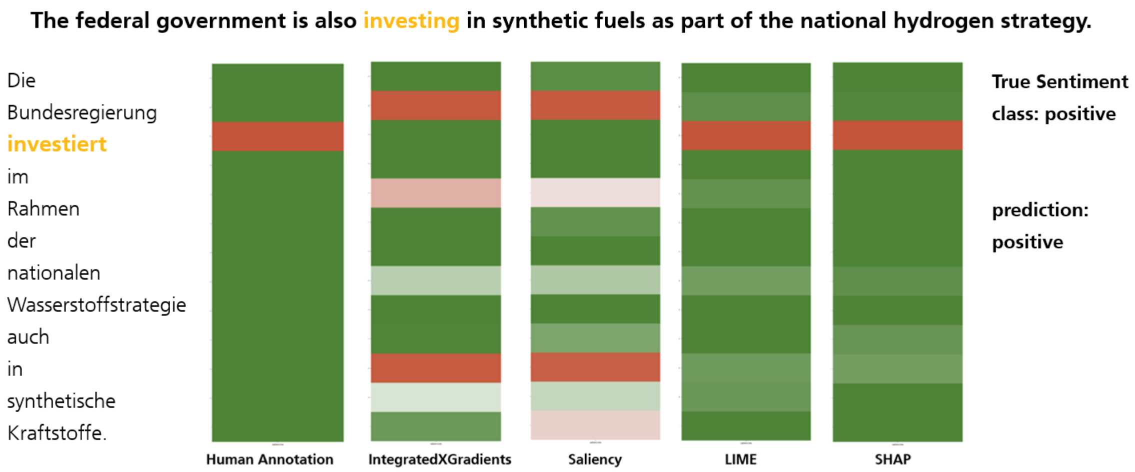


Figure 6: Explainability information generated by different XAI methods for the BERT Sentiment analysis model.

5.1 Results for Human Agreement Performance

Table 6 shows the Human Agreement performance of each explanation algorithm on our test dataset. Regarding plausibility, there is only modest agreement of explanations generated by XAI methods versus human rationales, independent of the ML model. Since rationales are generally very short, the model has a high chance of missing the main evidence in the text. Simpler models like CNN performed slightly better, so we observe a negative correlation between model performance and agreement with human rationales.

XAI methods usually produce a list of top-k features, ordered according to their saliency. However, hyperparameter setting and choice of the optimal cutoff threshold of the ranked feature list impacts the output results. To this aim, we calculated plausibility with a moving threshold t in the range [0.01, 0.2], varying the number of tokens being extracted as salient features, and selecting thresholds that result in the highest F1 score. Related work generally determines the threshold value heuristically based on the length ratio of the gold annotations or based on the saliency weights, as suggested by Yu et al. (2019). Interestingly, the results differ from the heuristically determined threshold based on the length ratio (see Table 2), and suggest that better results could be obtained by choosing a higher threshold, i.e, *negative* mean: 11.7%; std: 4.7, *neutral* mean: 12.3%; std: 5.5, *positive* mean: 11.4%; std: 5.7.

Model	Token F1	IOU
CNN [LIME]	0.066	0.036
LSTM [LIME]	0.062	0.034
BERT [LIME]	0.069	0.037
CNN [IG]	0.044	0.023
LSTM [IG]	0.068	0.037
BERT [IG]	0.039	0.021
CNN [Saliency]	0.062	0.033
LSTM [Saliency]	0.068	0.037
BERT [Saliency]	0.044	0.023
CNN [SHAP]	0.074	0.041
LSTM [SHAP]	0.071	0.039
BERT [SHAP]	**0.078**	**0.043**

Table 6: Evaluation Plausibility of the explainability techniques measured with Intersection over Union (IOU) and Token F1 Score with the gold human annotations.

5.2 Results for Faithfulness

In our use case, we require faithful explanations that are actually relevant to the model's prediction and inner workings. The sufficiency of rationales, based on the AUC score is shown in Table 7. A faithful rationale should display low sufficiency values, as the prediction should be highly influenced by the tokens selected as most salient. As can be seen, simpler model architectures achieve the highest faithfulness scores in terms of sufficiency, with CNN performing best.

	CNN	LSTM	BERT
LIME	**27.29**	37.84	52.66
IG	29.92	37.44	46.15
Saliency	29.32	36.85	46.71
SHAP	27.80	32.66	46.46

Table 7: Comparison of the Faithfulness of the explainability methods measured by AUC for thresholds $\in [0, 10, 20, .., 100]$ defined as the average difference of the AUC before and after masking the top k% words with zero padding. Lower scores are better.

5.3 Computational Efficiency

We also compare the computational efficiency of our XAI techniques. The wait time for an explanation should not be a bottleneck for the task workflow. We recorded the computational time to generate salient features (including visualization heatmaps) for our test dataset. As shown in Table 8, the computational time for generating explanations is particularly high for SHAP in combination with LSTM or BERT.

Explainability Model	BERT	LSTM	CNN
LIME	1549	4744	308
IG	498	40	63
Saliency	486	42	59
SHAP	13362	53917	1245

Table 8: Computational efficiency. We report the mean speed in seconds for generating a saliency map for the test dataset

6 Final Discussion

In this paper, we introduced a new benchmark corpus compiled from online news articles, annotated by 3 domain experts for document-level sentiment. Moreover, it contains multiple rationales that provide evidence for the annotators choice of the overall sentiment. Since many articles have a mixed sentiment, including borderline cases that are difficult to classify, highlighting positive as well as negative aspects mentioned in one single article, yields increased transparency.

We used the corpus as a benchmark for the German language, where resources for studying explainability are scarce. We investigated several ML architectures for the task, in combination with different post-hoc explainability methods. Since there is no single solution that is best suited to every use case, our analysis allows identifying the strengths and limitations of each method.

Our findings indicate that BERT yields the best performance in terms of sentiment classification accuracy. In combination with SHAP, it offers a global view of feature importance, which helps detecting spurious features and bias. We think that end users will profit from XAI methods which allow to get an aggregated view of feature importance for a particular topic category, or based on a specific time frame. However, due to the high dimension of our data, local explanations are overall not very plausible, regardless of the underlying ML model and explainability method. Moreover, the BERT model is less faithful than CNN and LSTM, due to high complexity of the model. For our use case, the computational time for generating explanations with LIME, IG or Saliency would be acceptable in a real-time application, except for SHAP which suffers from a long computational time.

In future work, we seek to identify the training data points responsible for model misclassifications and find training instances that show bias through influence functions (Koh and Liang, 2017), and investigate the impact of the pretrained embeddings and model.

7 Acknowledgements

We thank our students Ina Weidinger and Vincent Hahne for supporting us in the annotation task for the InfoBarometer dataset as well as for helping us to set up the annotation guidelines.

References

Dogu Araci. 2019. Finbert: Financial sentiment analysis with pre-trained language models. *arXiv preprint arXiv:1908.10063*.

Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. 2020. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115.

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2020. A diagnostic study of explainability techniques for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3256–3274, Online. Association for Computational Linguistics.

Javier Castro, Daniel Gómez, and Juan Tejada. 2009. Polynomial calculation of the shapley value based on sampling. *Computers & Operations Research*, 36(5):1726–1730.

Marina Danilevsky, Kun Qian, Ranit Aharonov, Yannis Katsis, Ban Kawas, and Prithviraj Sen. 2020. A survey of the state of explainable ai for natural language processing. *arXiv preprint arXiv:2010.00711*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jay DeYoung, Sarthak Jain, Nazneen Fatema Rajani, Eric Lehman, Caiming Xiong, Richard Socher, and Byron C. Wallace. 2020. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, Online. Association for Computational Linguistics.

Julia El Zini, Mohamad Mansour, Basel Mousi, and Mariette Awad. 2022. On the evaluation of the plausibility and faithfulness of sentiment analysis explanations. In *Artificial Intelligence Applications and Innovations: 18th IFIP WG 12.5 International Conference, AIAI 2022, Hersonissos, Crete, Greece, June 17–20, 2022, Proceedings, Part II*, pages 338–349. Springer.

Shilpa Gite, Hrituja Khatavkar, Ketan Kotecha, Shilpi Srivastava, Priyam Maheshwari, and Neerav Pandey. 2021. Explainable stock prices prediction from financial news articles using sentiment analysis. *PeerJ Computer Science*, 7.

Xiaochuang Han, Byron C. Wallace, and Yulia Tsvetkov. 2020. Explaining black box predictions and unveiling data artifacts through influence functions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5553–5563, Online. Association for Computational Linguistics.

Jochen Hartmann, Mark Heitmann, Christian Siebert, and Christina Schamp. 2022. More than a feeling: Accuracy and application of sentiment analysis. *International Journal of Research in Marketing*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9:1735–1780.

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable NLP systems: How should we define and evaluate faithfulness? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4198–4205, Online. Association for Computational Linguistics.

Eleni Kalamara, Arthur Turrell, Chris Redl, George Kapetanios, and Sujit Kapadia. 2022. Making text count: economic forecasting using newspaper text. *Journal of Applied Econometrics*, 37(5):896–919.

Daisuke Katayama, Yasunobu Kino, and Kazuhiko Tsuda. 2019. A method of sentiment polarity identification in financial news using deep learning. *Procedia Computer Science*, 159:1287–1294.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Conference on Empirical Methods in Natural Language Processing*.

Pang Wei Koh and Percy Liang. 2017. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1885–1894. PMLR.

Abby Levenberg, Stephen Pulman, Karo Moilanen, Edwin Simpson, and Stephen Roberts. 2014. Predicting economic indicators from web text using sentiment composition. *International Journal of Computer and Communication Engineering*, 3(2):109–115.

Jiwei Li, Will Monroe, and Dan Jurafsky. 2016. Understanding neural networks through representation erasure.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30.

Andreas Madsen, Siva Reddy, and Sarath Chandar. 2021. Post-hoc interpretability for neural nlp: A survey. *arXiv preprint arXiv:2108.04840*.

Anna Marszal et al. 2022. What news can really tell us? evidence from a news-based sentiment index for financial markets analysis. Technical report, Narodowy Bank Polski, Economic Research Department.

Kostadin Mishev, Ana Gjorgjevikj, Irena Vodenska, Lubomir T Chitkushev, and Dimitar Trajanov. 2020. Evaluation of sentiment analysis in finance: from lexicons to transformers. *IEEE access*, 8:131662–131682.

Christoph Molnar. 2022. *Interpretable Machine Learning*, 2 edition.

Nadia Nedjah, Igor Santos, and Luiza de Macedo Mourelle. 2022. Sentiment analysis using convolutional neural network via word embeddings. *Evolutionary Intelligence*, 15(4):2295–2319.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

Bo Pang, Lillian Lee, et al. 2008. Opinion mining and sentiment analysis. *Foundations and Trends® in information retrieval*, 2(1–2):1–135.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, and Rada Mihalcea. 2020. Beneath the tip of the iceberg: Current challenges and new directions in sentiment analysis research. *IEEE Transactions on Affective Computing*.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Model-agnostic interpretability of machine learning. *arXiv preprint arXiv:1606.05386*.

Lina Maria Rojas-Barahona. 2016. Deep learning for sentiment analysis. *Language and Linguistics Compass*, 10(12):701–719.

Jürgen Schmidhuber, Sepp Hochreiter, et al. 1997. Long short-term memory. *Neural Computing*, 9(8):1735–1780.

Kazuhiro Seki, Yusuke Ikuta, and Yoichi Matsubayashi. 2022. News-based business sentiment and its properties as an economic index. *Information Processing & Management*, 59(2):102795.

L Shapley. 1953. Quota solutions op n-person games1. *Edited by Emil Artin and Marston Morse*, page 343.

Avanti Shrikumar, Peyton Greenside, Anna Shcherbina, and Anshul Kundaje. 2016. Not just a black box: Learning important features through propagating activation differences. *arXiv preprint arXiv:1605.01713*.

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Mokanarangan Thayaparan, Marco Valentino, and André Freitas. 2020. A survey on explainability in machine reading comprehension. *ArXiv*, abs/2010.00389.

Hao Tian, Can Gao, Xinyan Xiao, Hao Liu, Bolei He, Hua Wu, Haifeng Wang, and Feng Wu. 2020. Skep: Sentiment knowledge enhanced pre-training for sentiment analysis.

Eric Wallace, Matt Gardner, and Sameer Singh. 2020. Interpreting predictions of NLP models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, pages 20–23, Online. Association for Computational Linguistics.

Lijie Wang, Hao Liu, Shuyuan Peng, Hongxuan Tang, Xinyan Xiao, Ying Chen, Hua Wu, and Haifeng Wang. 2021. Dutrust: A sentiment analysis dataset for trustworthiness evaluation. *arXiv preprint arXiv:2108.13140*.

Lijie Wang, Yaozong Shen, Shuyuan Peng, Shuai Zhang, Xinyan Xiao, Hao Liu, Hongxuan Tang, Ying Chen, Hua Wu, and Haifeng Wang. 2022. A fine-grained interpretability evaluation benchmark for neural nlp. *arXiv preprint arXiv:2205.11097*.

Daniel Weiss and Fabian Nemeczek. 2022. A media-based innovation indicator: Examining declining technological innovation systems. *Environmental Innovation and Societal Transitions*, 43:289–319.

Sarah Wiegreffe and Ana Marasović. 2021. Teach me to explain: A review of datasets for explainable nlp. *ArXiv*, abs/2102.12060.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Frank Xing, Lorenzo Malandri, Yue Zhang, and Erik Cambria. 2020. Financial sentiment analysis: an investigation into common mistakes and silver bullets. In *Proceedings of the 28th international conference on computational linguistics*, pages 978–987.

Mo Yu, Shiyu Chang, Yang Zhang, and T. Jaakkola. 2019. Rethinking cooperative rationalization: Introspective extraction and complement control. In *Conference on Empirical Methods in Natural Language Processing*.

Omar Zaidan and Jason Eisner. 2008. Modeling annotators: A generative approach to learning from annotator rationales. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 31–40, Honolulu, Hawaii. Association for Computational Linguistics.

Omar Zaidan, Jason Eisner, and Christine Piatko. 2007. Using “annotator rationales” to improve machine learning for text categorization. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 260–267, Rochester, New York. Association for Computational Linguistics.

A Appendix

A.1 Query Synonyms

q1: ["Elektromobilität","Elektroauto","Stromer", "Elektrofahrzeug", "BEV","PHEV", "Electromobility", "e-mobility", "e-car", "electric car","emobility", "e-mobilität", "Emobilität"]

q2: ["autonomes Fahren","autonomes Auto","autonomes Fahrzeug","selbstfahrendes Auto", "selbstfahrendes Fahrzeug","fahrerloses Fahren","fahrerloses Auto","fahrerloses Fahrzeug", "autonomous driving","Robotaxi", "autonomes Taxi", "autonomes Shuttle", "autonome Mobilitätsdienstleistungen","robotaxi"]

q3: ["Wasserstoff", "Hydrogen", "FCEV", "Brennstoffzelle", "Wasserstofffahrzeug", "Wasserstoffauto", "Brennstoffzellenfahrzeug", "Brennstoffzellenauto", "fuel cell", "hydrogen", "Power-to-X"]

q4: ["Diesel"]

q5: ["SynFuels", "SynFuel", "Synthetische Kraftstoffe", "Syn-Fuels", "EFuels", "E-Fuels", "alternative Kraftstoffe"]

A.2 Implementation Details

A.2.1 SVM

We utilize a linear svm model operating on bag of word features weigthed by tf.idf, and lowercasing all words. To optimize the models, we employ full-batch gradient descent with L1 regularization on the weight matrices.

A.2.2 CNN

For CNN, we employ an embedding dimension of 300, which is initialized by the GloVe embeddings. The batch size is 64, with a 0.1 dropout rate. We utilize the AdamW optimizer with a learning rate of 5e-5. Window sizes $\in [2, 3, 4, 5]$ were selected, with 100 out channels. Regarding stride, we keep the hyperparameter at the default value 1. CNN models are trained using an early stop over the validation set and up to 50 training epochs.

A.2.3 LSTM

For LSTM, we employ an embedding dimension of 300, which is initialized by the GloVe embeddings. The batch size is 64, and the dropout rate 0.1. We use a learning rate of 5e-5 along with the AdamW optimizer. 4 LSTM layers were selected, with a hidden layer size of 300. LSTM models are trained using an early stop over the validation set and up to 50 training epochs.

A.2.4 Transformer

As a base for all our experiments we use the German BERT-BASE model which consists of 12 layers, a hidden state size of 768 dimensions per token amounting to a total of 110 million parameters. The parameters of this model are initialized using bert-base-german-cased, which has been released by deepset.ai. We trained the model with a learning rate of 5e-5. We chose the best model using early stopping with the best number of epochs determined by using the validation splits.

Examining Bias in Opinion Summarisation Through the Perspective of Opinion Diversity

Nannan Huang Lin Tian Haytham Fayek Xiuzhen Zhang
RMIT University, Australia
{amber.huang, lin.tian2}@student.rmit.edu.au
haytham.fayek@ieee.org
xiuzhen.zhang@rmit.edu.au

Abstract

Opinion summarisation is a task that aims to condense the information presented in the source documents while retaining the core message and opinions. A summary that only represents the majority opinions will leave the minority opinions unrepresented in the summary. In this paper, we use the stance towards a certain target as an opinion. We study bias in opinion summarisation from the perspective of opinion diversity, which measures whether the model generated summary can cover a diverse set of opinions. In addition, we examine opinion similarity, a measure of how closely related two opinions are in terms of their stance on a given topic, and its relationship with opinion diversity. Through the lens of stances towards a topic, we examine opinion diversity and similarity using three debatable topics under COVID-19. Experimental results on these topics revealed that a higher degree of similarity of opinions did not indicate good diversity or fairly cover the various opinions originally presented in the source documents. We found that BART (Lewis et al., 2020) and ChatGPT can better capture diverse opinions presented in the source documents.

1 Introduction

The aim of opinion summarisation is to reduce the amount of information in the source text while maintaining the core message and opinions expressed therein. It can be in the form of a summary of a product review (Alam et al., 2016; Bražinskas et al., 2020; Chu and Liu, 2019), online discourse using platforms like Twitter, Reddit (Fabbri et al., 2021; Bilal et al., 2022) or other types of text with opinions. There are two major types of models, extractive (Meng et al., 2012; Ku et al., 2006; Erkan and Radev, 2004; Liu, 2019) and abstractive (Lewis et al., 2020; Raffel et al., 2020; Zhang et al., 2020). The extractive models extract the key information by selecting the most salient sentences in the source documents. Whereas the abstractive summarisation models generate new phrases that reflect the key information in the source documents. Applications for this activity include tracking consumer opinions, assessing political speeches, or internet conversations, among many others.

A summarisation model's output will reflect any biases present in the data used for training the model. Moreover, summarisation models are used in many applications where the fairness of the outputs is critical. For instance, opinion summarisation models can be used to summarise product reviews and social media posts. If these models produce biased summaries, they have the risk of being misused as a tool to influence and manipulate people's opinions. Therefore, it is essential to look into the fairness of the models.

Earlier studies of bias in opinion summarisation have mainly evaluated biases in summarisation by comparing whether the information is selected from different social groups equally, using attributes including gender, dialect, and other societal properties (Dash et al., 2019; Blodgett et al., 2016; Keswani and Celis, 2021; Olabisi et al., 2022). Such representation is fair from the perspective of information input, leaving the other side of information consumption under studied. It is equally important to look at fairness from end-users' perspective. Shandilya et al. (2020) pointed out that fair representation from the information input perspective does not always imply fair representation from end-users' perspective. From the standpoint of end-users, it is more important for summaries to cover the various opinions presented in the source documents so that the various opinions in the source documents can be heard equally (Shandilya et al., 2020). In this study, we examine bias in opinion summarisation from end-users' perspective by comparing the output of different modern summarisation models.

Blodgett et al. (2020) noted that many research work on bias in natural language processing lacks a precise description and definition of *bias* in their

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 149–161
July 14, 2023 ©2023 Association for Computational Linguistics

study, making it challenging for readers to understand. Our working definitions of the key terms are as follows. According to the *stance triangle theory* (Du Bois, 2007), a stance is composed of three components: the object, the subject, and the attitude, which shape sociocultural value systems and opinion expression. While studies such as Misra et al. (2016) and Reimers et al. (2019) utilise arguments with similarity and clustering techniques to capture similar opinions, our study takes a different approach based on the stance triangle theory. In our study, an *opinion* is a personal stance about a particular target, and a stance is a position or attitude that an individual takes about a given target. Our definition of *bias in opinion summarisation* is when the summary focuses more on certain opinions than on the diversity of opinions presented in the source documents. Note that it is generally agreed that diverse opinions should be taken into account regardless of their quantitative value in order to include more diverse opinions when using sentiment information (Angelidis and Lapata, 2018; Siledar et al., 2023). Hence, our focus is on measuring the diversity of opinions rather than quantity. This is measured through *opinion diversity* which assesses opinion equivalence relative to the source documents. It measures the opinions in the source documents that the generated summary contains. In addition, we compare opinion similarity between the source and generated documents and further examine the relationship between opinion diversity and opinion similarity. *Opinion similarity* is a measure of how closely related two opinions are in terms of their stance on a given topic. We use BERT (Devlin et al., 2019) to compare the semantic closeness in the embedding space. We aim to understand whether models perform well in capturing overall opinions that are less biased by covering diverse opinions. We examine opinions on three COVID-19 topics using stances on these topics.

In our study, we aim to answer the following questions:

1. How well can summarisation models present various opinions in a document from the perspective of stance towards a topic?
2. Does a greater degree of opinion similarity with the source documents suggest a lack of bias in summarisation models?

2 Related Work

2.1 Opinion summarisation

Opinion summarisation has received significant attention in recent years, with extractive models such as Hybrid-TFIDF (Inouye and Kalita, 2011) a frequency-based summarisation method, having great performance for summarising social media data like tweets. Recent studies have introduced the concept of key point analysis (Bar-Haim et al., 2020a,b, 2021), which uses extractive models to identify key arguments from the source documents and match them to the main opinions' associated key points. Abstractive opinion summarisation models, such as Copycat (Bražinskas et al., 2020) and MeanSum (Chu and Liu, 2019), are designed to address the problem of summarising product or business reviews. MeanSum (Chu and Liu, 2019) is based on LSTM, while Copycat (Bražinskas et al., 2020) uses a variational autoencoder (VAE) to generate latent vectors of given reviews.

2.2 Biases in opinion summarisation

Existing studies of bias in opinion summarisation have focused on the perspective of using sensitive attributes of social media users and categorising them under different social groups (Dash et al., 2019). These attributes include social identities like gender, race and political leaning information. Other studies focused on the perspective of dialect used in text and whether the generated summaries cover such dialects (Blodgett et al., 2016; Keswani and Celis, 2021; Olabisi et al., 2022).

One limitation of these approaches is that they require obtaining sensitive hidden attributes from the text producers. Due to other factors including privacy, this is not always practicable. Another limitation, as pointed out in Shandilya et al. (2020) is the assumption of fairness in using information from different social groups equally and this is from the input information perspective. Fairness should also be evaluated from the perspective of end-users. Shandilya et al. (2020) discovered in their study that, from the perspective of end-users, it is more important to present diverse information in the source documents. We argue that considering fairness from the perspective of end-users is equally important since they are the users of the final product and make informed decisions based on the generated summaries.

Therefore, in our work, we focus on the fair representation of information generated by summari-

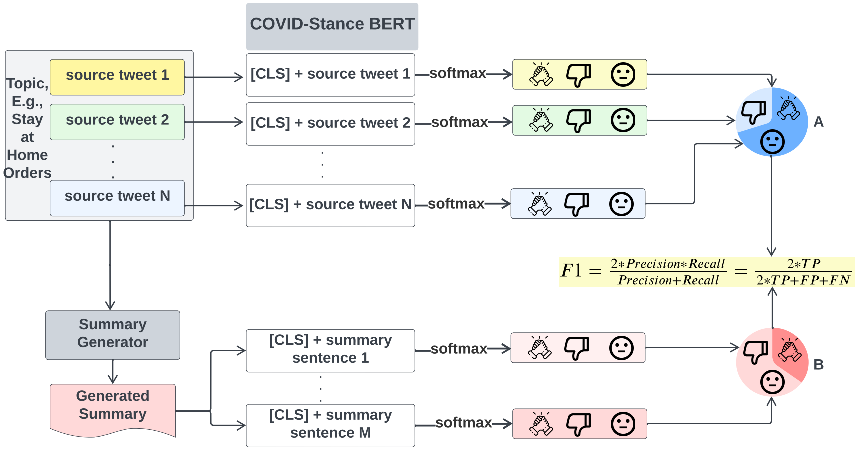


Figure 1: The process of calculating opinion diversity: COVID-Stance-BERT is applied to each source tweet and aggregated into a single set of opinions. The same process is applied to the generated summaries at the sentence level. The F1 score is applied among the two sets of opinions, where the true positive (TP) measures opinions captured by the model generated summary that is also presented in the source documents; false positive (FP) measures opinions captured by the model generated summary that is not presented in the source documents; false negative (FN) measures opinions not captured by the model generated summary that is presented in the source documents.

sation models from end-users' perspective. We study bias in opinion summarisation by examining whether the summary focuses more on certain opinions than on the diverse opinions presented in the source documents.

3 Methodology

As previously stated, we capture opinion in our study by utilising stance and its target. When the generated summaries reflect opinions that diverge from those of the source texts, the summary is considered biased. This could take the form of concentrating on a narrow range of opinions or going beyond what was expressed in the source documents. It should be noted that the biased information in the source text is not the main focus of our study; rather, we are interested in how opinions are presented and whether the opinions in the generated summaries match those in the source documents.

We formulate our problems in three steps. We first use a pre-trained language model to capture opinions from the input sequences. The opinion diversity is then calculated using the F1 score between the set of stances in the source tweets and the generated summary for each cluster (discussed below) under each topic. Finally, we compare the cosine similarity between the source tweet cluster and the summary at the sentence level to measure opinion similarity using model representation.

Let $\mathcal{C} = \{c_0, c_1, c_2, ..., c_l\}$ be a set of clusters. These clusters are derived from the three main topics ("CDC", "Stay at Home Orders" and "Wearing a Face Mask"). For each c, we have a set of source tweets $\mathcal{T}$, defined as $\mathcal{T} = \{t_0, t_1, t_2, ..., t_n\}$. For each c, we have a set of generated summaries $\mathcal{S} = \{\mathrm{s}_0, \mathrm{s}_1, \mathrm{s}_2, ..., \mathrm{s}_l\}$, where each summary s consists of a lists of sentences, defined as $\mathrm{s} = \{e_0, e_1, e_2, ..., e_q\}$, where e refers to the textual content of each input sentence.

3.1 Capturing Opinions

We train COVID stance classification models related to several COVID-19 topics or targets using a publicly available dataset from Glandt et al. (2021) on COVID-19 related stance detection. The dataset consists of four different COVID-19 related topics and targets. See Appendix A.1 for further detail on the summary of the data distribution. Similar to Glandt et al. (2021), using the further pre-trained BERT model (bert-large-uncased)[1] with the COVID-19 tweet corpus (Müller et al., 2020), we fine-tune the model with stance labeled data from Glandt et al. (2021). Thus, our COVID-Stance-BERT is a further pre-trained BERT fine-tuned with standard cross entropy loss to do a three-class classification of stance labels (support,

---

[1] `https://huggingface.co/bert-large-uncased`

against, and neutral). Each tweet t_i is associated with a ground-truth label $d \in \mathcal{D}$, where $\mathcal{D}$ represents the label set (support, against or neutral; 3 classes).

$$v_i = \text{BERT}([\text{CLS}] \oplus x_i) \tag{1}$$

$$\hat{d}_i = \text{softmax}(Wv_i + b) \tag{2}$$

We call these models COVID-Stance-BERT in the remainder of the paper. The average accuracy and macro F1 scores across three targets are 0.8208 and 0.8026 respectively. Similar levels of accuracy and F1 scores were obtained across these topics compared to Glandt et al. (2021). The detailed result is reported in Appendix A.2.

3.2 Opinion Diversity

The overall process is presented in Fig 1. In order to determine whether different opinions stated in the source documents can be captured using summarisation models, we apply COVID-Stance-BERT to the source documents to produce a collection of opinions represented in the input tweets and the generated summaries. The majority of the source tweets have only a single sentence. We therefore treat each sentence in the generated summaries as a tweet and apply the same COVID-Stance-BERT to retrieve its opinions. We apply the in-domain stance detection model on both the source documents and the summarised sentences across the tweet clusters under different topics. That is, a stance detection model fine-tuned to the target of "Stay at Home Orders" is applied to the collection of tweets and the generated summaries towards the "Stay at Home Orders" topic.

Once the prediction is done on both the source documents and the generated summaries, we compare the sets of opinions and examine if the generated summaries cover the various opinions presented in the source document. We compute the F1 score by comparing the two sets of stances under each discussed topic across all clusters and use them to represent how well the summarisation model captures the stances in relation to the input documents.

The detailed calculation can be found as follows: we apply COVID-Stance-BERT stance prediction to calculate the opinion diversity on both the set of source tweets and all generated summaries. For each generated summary $s_j \in \mathcal{S}$, we adopt sentence splitting function[2] as $s_j = \{ss_0, ss_1, ..., ss_m\}$ For each sentence e in generated summary s_j, we take the associated stance label d_e, formally,

$$\begin{aligned} d_t &= \text{BERT}(\text{emb}([CLS], t_p)), \\ d_{ss} &= \text{BERT}(\text{emb}([CLS], ss_q)), \end{aligned} \tag{3}$$

where t_p represents the text of the source tweet and s_q is the summary, emb() the embedding function and d_t and d_{ss} are the stance predictions based on the $[CLS]$ token produced by our COVID-Stance-BERT.

Once the above is completed, we get the set of non-repeated opinion(s) in both the source documents and the generated summaries. We use the F1 score, which measures the harmonic mean of opinion precision and opinion recall, to evaluate the performance of opinion diversity, where the opinion precision measures the proportion of important opinions in the generated summary. The opinion recall measures the degree of salient opinion in the source documents that the generated summary contains. A higher F1 score indicates the model generated summary can better cover the various opinions presented in the source documents. The true positive (TP), false positive (FP), and false negative (FN) are measured as follows:

- TP = opinions captured by the model generated summary that is also presented in the source documents.
- FP = opinions captured by the model generated summary that is not presented in the source documents.
- FN = opinions not captured by the model generated summary that is presented in the source documents.

More detail on illustrations of various scenarios of opinion precision and recall and their associated F1 scores can be found in Appendix A.3. We report the average across all clusters under each topic as the overall opinion diversity for each model.

3.3 Opinion Similarity

The overall process of evaluating opinion similarity is visualised in Fig 2. To aggregate all the source tweets in a cluster to get a source representation

---

[2] `https://www.nltk.org/api/nltk.tokenize.html`

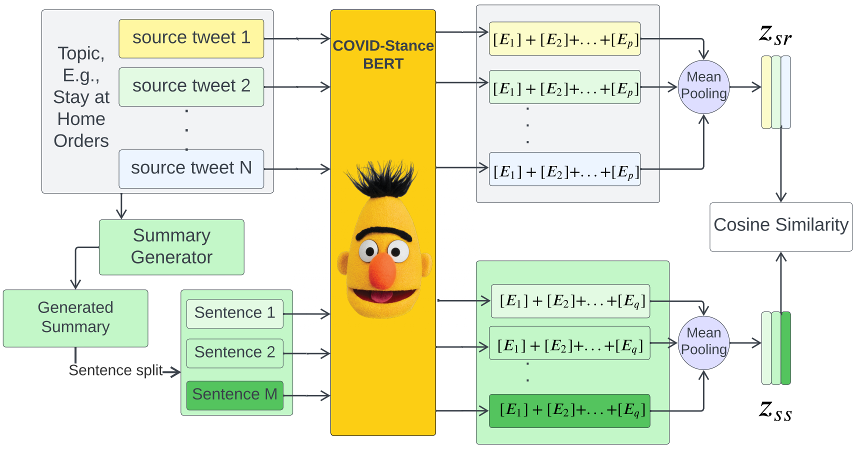


Figure 2: The process of calculating opinion similarity. For the source tweets, a single representation is obtained using mean-pooling approach based on the mean representations of all input tokens. For the generated summaries, the same process is applied at the sentence level. The cosine similarity is then applied to these representations to measure opinion similarity.

(z_{sr}), we adopt the mean-pooling approach based on the mean representations of all input tokens:

$$z_{sr} = \frac{1}{|n|} \sum_{i \in n} h_i^c, \tag{4}$$

where h_i^c is the mean stance representation derived from source tweet t_i.

To extract the representation for each generated summary (z_{ss}), we applied the mean-pooling approach over the average of all token representations from the last layer of our language model encoder as shown in Eq. 5:

$$z_{ss} = \frac{1}{|m|} \sum_{i \in m} v_i^c, \tag{5}$$

where v_i^c is the mean stance representation based on each token representation from a given summary s_i.

For each generated summary, we compute the cosine similarity to the average source representation in the cluster, with the following equation:

$$\cos(z_{sr}, z_{ss}) = \frac{z_{sr} z_{ss}}{\|z_{sr}\| \|z_{ss}\|}. \tag{6}$$

A higher cosine similarity score between the source and the summary indicates a model is better at capturing the overall opinions and semantic information presented in the source documents.

In summary, we investigate bias in opinion summarisation models from the standpoint of opinion diversity. The idea is to look at whether summarisation models can capture the various opinions presented in the source documents. In addition, from the perspective of opinion similarity, we evaluate how closely related the generated summary and the source documents are in terms of their stance and semantic information on a given topic. Combining the results of opinion diversity and opinion similarity, we aim to understand whether summaries that express opinions that are overall similar to the source documents also indicate they are less biased.

4 Experiments and Results

4.1 Data

In this study, we use the COVID-19 tweet clusters dataset provided by Bilal et al. (2022). The dataset contained tweet clusters that are coherently opinionated, coherently non-opinionated, and incoherent subsets. Each topic contains a different number of clusters. A cluster of tweets contains a collection of tweets obtained during a particular timestamp on a specific topic. In Bilal et al. (2022) each cluster was used for generating the final summary. Hence, in our work, we focus on the cluster level when generating summaries for each topic. For example, under the topic "Wearing a Face Mask" there can be multiple clusters obtained at different times; each cluster of tweets could be discussing whether wearing a face mask is a good idea, obtained at a different time of the year.

In our experiment, we are only considering tweet

Table 1: The statistic of the number of tweet clusters and the average number of tweets in each cluster under different topics.

Topic	No. Clusters	Ave No. Tweets
CDC	78	21.77
Stay at Home Orders	48	20.54
Wearing a Face Mask	52	22.42

clusters that are coherently opinionated, with a similar discussion of targets as the ones mentioned in Glandt et al. (2021). This is to ensure that all tweets contain opinions and, at the same time, to utilise the stance detection model in an in-domain setting to evaluate the opinions expressed in these clusters. We obtained coherently opinionated clusters including "Stay at Home Orders", "Wearing a Face Mask" and a highly related topic, "CDC/Centres for Disease Control and Prevention", where the clusters of tweets centred around the aforementioned topics are mainly focusing on the discussion and the expression of opinions towards them during the COVID-19 pandemic. The in-domain stance detection models were then applied to the clusters apart from "CDC/Centres for Disease Control and Prevention" where we applied the stance detection model that was trained on the "Fauci" topic since they are both public figures.

The overall data distribution after our selection can be found in Table 1. There are 78, 48, and 52 clusters of tweets; and on average, 21.77, 20.54, and 22.42 tweets under each cluster for the "CDC", "Stay at Home Orders", and "Wearing a Face Mask" topics, respectively.

4.2 Baseline Models

Several summarisation models are used to generate summaries in our experiments, including extractive summarisation models TextRank (Mihalcea and Tarau, 2004), LexRank (Erkan and Radev, 2004) and Hybrid-TFIDF (Inouye and Kalita, 2011); and abstractive summarisation models BART (Lewis et al., 2020), Pegasus (Zhang et al., 2020) and T5 (Raffel et al., 2020), a summarisation model for review Copycat (Bražinskas et al., 2020) and a recently released Large Language Model(LLM) - ChatGPT[3]. Following Bilal et al. (2022) we limit the abstract summarisation models word limit to the generated summary within [90%, 110%] of the gold standard length; and the average token length of the gold standard length for ChatGPT. For extractive models only allows selecting sentences we limit to the average number of sentences of the gold standard length. We apply all the models mentioned in a zero-shot setting. More in-depth discussion on each of the models is below:

- **BART** (Lewis et al., 2020) is an encoder-decoder model with a bidirectional encoder and a left-to-right decoder. Pretrained using a novel in-filling technique by replacing a span of text with a single mask token. Making it useful for language generation tasks We use the BART large model, pre-trained on CNN/Daily Mail [4].

- **Pegasus** (Zhang et al., 2020) is a model that employs the Transformer encoder and decoder, self-supervised learning and pre-training on predicting the removed sentences, and tokens similar to the masked language model. We use the Pegasus model pre-trained on CNN/Daily Mail [5].

- **T5** (Raffel et al., 2020) is an encoder-decoder model pretrained on a multi-task setting using both supervised and unsupervised settings where the tasks are converted into a set of input-output text pairs. This allows it to understand a large variety of relationships between texts. We use the T5 base model pre-trained on CNN/Daily Mail [6].

- **ChatGPT** OpenAI's ChatGPT, a recently released Large Language Model (LLM), was developed by employing reinforcement learning from human feedback (RLHF) (Christiano et al., 2017) to train a GPT-3.5 series model. We employ OpenAI's ChatGPT API (gpt-3.5-turbo-0301) for our experiments. We adjusted the maximum tokens to the average gold standard token length. We prompt ChatGPT as below "Summarise the following tweets: You should write it in tweet style. You should use no more than 4 sentences. Tweets: [Source tweets]".

---

[3] `https://chat.openai.com/`

[4] `https://huggingface.co/facebook/bart-large-cnn`

[5] `https://huggingface.co/google/pegasus-cnn_dailymail`

[6] `https://huggingface.co/flax-community/t5-base-cnn-dm`

Table 2: Results of opinion diversity (Opi Div) and opinion similarity (Opi Sim) for various models under different discussed topics. The best results are bolded, and the ranking of the models is provided inside the brackets.

Events	CDC		Stay at Home Orders		Wearing a Face Mask	
Models	Opi Div	Opi Sim	Opi Div	Opi Sim	Opi Div	Opi Sim
BART	**0.7449 (1)**	0.8503 (4)	0.7681 (2)	0.8373 (7)	**0.8147 (1)**	0.8412 (6)
Pegasus	0.5265 (7)	0.8745 (3)	0.7576 (3)	0.8775 (3)	0.3692 (5)	0.8768 (3)
T5	0.6346 (3)	0.8451 (5)	0.7417 (5)	0.8407 (6)	0.4692 (4)	0.8327 (7)
ChatGPT	0.7282 (2)	0.8818 (2)	**0.8014 (1)**	0.8515 (5)	0.6006 (3)	0.8498 (5)
Copycat	0.5265 (7)	0.6725 (8)	0.7014 (8)	0.7288 (8)	0.6737 (2)	0.7177 (8)
TextRank	0.5338 (6)	0.8370 (6)	0.7417 (5)	0.8519 (4)	0.2615 (8)	0.8828 (2)
LexRank	0.5530 (5)	0.8208 (7)	0.7569 (4)	0.8817 (2)	0.3590 (7)	0.8607 (4)
Hybrid TFIDF	0.5697 (4)	**0.8914 (1)**	0.7063 (7)	**0.8923 (1)**	0.3667 (6)	**0.8965 (1)**

- **Copycat** (Bražinskas et al., 2020) is a Variational Autoencoder model overcomes issues with limited training data in the context of review summarisation using self-supervision and latent representation that represents the general opinions expressed in the source reviews. We use the model provided by the authors [7].

- **TextRank** (Mihalcea and Tarau, 2004) is a graph-based model that extracts the most important sentence from the input document based on the weight determined by the edges connected to the words or phrases.

- **LexRank** (Erkan and Radev, 2004) is a graph-based model represents each sentence in the document as a node. Edges between vertices are calculated using cosine similarity, and the importance of a sentence is determined by the number of connected edges. The model extracts the most important sentence from the document based on the connectivity matrix.

- **Hybrid-TFIDF** (Inouye and Kalita, 2011) is a graph-based model similar to TextRank (Mihalcea and Tarau, 2004) and LexRank (Erkan and Radev, 2004). Each word in the sentence is represented using the TF-IDF score, and similarity between sentences is computed to build edges between sentence vertices. Similar to other graph-based models, sentences with the most connected edges are deemed most important.

4.3 Opinion Diversity by Models

To answer the first question, we first gather opinions from the source input tweets by running COVID-Stance-BERT on each tweet in each cluster under different topics. Similarly, the same model is used to collect opinions captured by different summarisation models as discussed in Section 3.2. Once the sets of opinions from the source document and the generated summaries are obtained, we compare whether the opinions presented in the source document are also the opinions captured by the models. This is to answer the question of whether models can fairly preserve the various opinions being presented in the source documents.

We compare the three different discussed topics listed in Table 1 using the clusters within each topic. Summaries were first obtained using the models mentioned in Section 4.2 for each cluster. Then, using the opinion sets obtained for both the source documents and the generated summaries, we compare whether the summaries have high opinion diversity by comparing the set of stance(s) captured by the summarisation models with the set of stance(s) presented in the source documents. See Appendix A.4 for details of the stance detection example and an example calculation of opinion diversity.

The results of the opinion diversity of different models under different discussion topics are listed in Table 2. We are reporting the F1 score for opinion diversity. The results of opinion precision and recall can be found in Appendix A.5. BART (Lewis et al., 2020) has the best opinion diversity for both the "CDC" and "Wearing a Face Mask" topics, while it has a competitive result for the "Stay at Home Orders". ChatGPT has the best opinion diversity for "Stay at Home Orders" and is

---

[7] `https://github.com/abrazinskas/Copycat-abstractive-opinion-summarizer`.

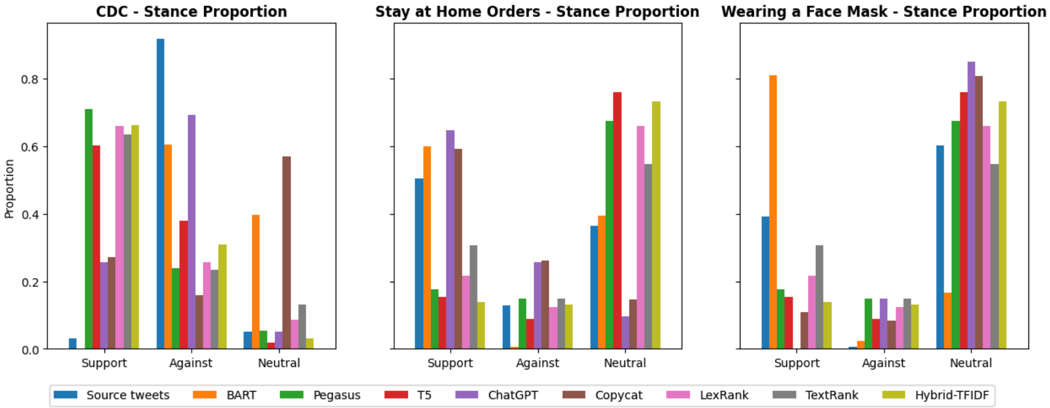


Figure 3: Stance distribution across various discussion topics for source documents and model generated summaries.

relatively competitive for the other topics.

Overall speaking, BART has the best performance in capturing diverse opinions and followed by ChatGPT. It is worth noticing, however, that previous studies found that ChatGPT tends to generate lengthy summaries when using the default parameters (Wang et al., 2023; Yang et al., 2023) and it is important to provide guidance in the prompt. The fact that we limit its maximum tokens and ask it to generate a fixed number of sentences may hurt its performance. For example, in the generated summary presented in Appendix A.5, ChatGPT generated an unfinished sentence.

4.4 Opinion Similarity by Models

The second question we aimed to answer is whether a model that generates summaries with high overall opinion similarity to the source documents would also be less biased by covering various opinions. To answer this question, we used the average input token representation obtained using the COVID-Stance-BERT model, for both the source documents and also the generated summaries.

For the source documents, the average input token representation for each tweet was obtained using COVID-Stance-BERT, followed by applying mean-pooling to aggregate them into a single representation. For the summary, a similar approach was applied to the generated summary at the sentence level. We obtained average input token representations for each sentence in the generated summary, and then a mean pooling was applied over the sentence representations to obtain a single representation for the generated summary. Once the representations are obtained for both the source documents and the model generated summaries. We compare the stance similarity between source documents against different model generated summaries using the cosine similarity between the source representation and the summary representation.

Table 2 shows the cosine similarity for measuring how similar the overall stance presented in the generated summaries is to the source documents. From the result, we can see that the model with the overall highest similarity of opinions across different topics is Hybrid-TFIDF (Inouye and Kalita, 2011). We suspect this can be due to the superior performance of Hybrid-TFIDF (Inouye and Kalita, 2011) in summarising microblogging text. This is most likely because Twitter posts are not like typical documents and are unstructured, disconnected, and brief (Inouye and Kalita, 2011). As a result, when compared to the source documents, it maintains a similar overall stance.

4.5 Opinion Distribution

We computed the stance distribution across the discussion topics using the proportion of stances in the source documents and different summarisation models. The outcome is depicted in Fig. 3. From the distribution result, we can see that BART (Lewis et al., 2020) can better preserve the stance distribution for both the "CDC" and "Wearing a Face Mask" topics, whereas ChatGPT can better preserve the stance distribution for the "Stay at Home Orders" topic. While most models could pick up various stances and present a similar stance distribution compared to the source documents for the "Stay at Home Orders" topic. We suspect this is due to the fact that this topic has no obvious minority opinion, making it less challenging for models to cover diverse opinions. The above aligns with what we found in Section 4.3 where BART (Lewis et al., 2020) and ChatGPT outperformed the other models by covering more diverse opinions. We believe one of the possible reasons could be due to BART's impressive multi-document summarisation capabilities (Chen and Yang, 2020; Johner et al., 2021) and ChatGPT's good performance in multiple downstream NLP tasks.

In conclusion, based on the results in Section 4.3 and 4.4 we found that when the generated summary has a higher degree of similarity in terms of overall opinion, that does not indicate it is fair in terms of covering more diverse opinions. In combination with the result from Section 4.5 we observed that the model's ability to capture various opinions on different topics is case-dependent. We found that when no obvious minority stance is presented in the source documents, most models could capture various opinions in the source document.

5 Conclusion

In this work, we proposed a new way to examine bias in opinion sumamrisation from the perspective of presenting various opinions in the summary. We investigated various summarisation models for the COVID-19 event under three topics, using stance towards a target as a representation of opinions. In addition, we also examined overall stance similarity using model representation. We found that BART (Lewis et al., 2020) and ChatGPT are better at capturing diverse opinions when generating a summary. Whereas Hybrid-TFIDF has the highest similarity across the three discussed topics for overall opinion similarity. Based on the result, we found that higher opinion similarity does not indicate that the model presents diverse opinions. While both attributes are important for evaluating bias in opinion summarisation we suggest future studies look into introducing metrics that can evaluate summaries from both perspectives.

References

Md Hijbul Alam, Woo-Jong Ryu, and SangKeun Lee. 2016. Joint multi-grain topic sentiment: modeling semantic aspects for online reviews. *Information Sciences*, 339:206–223.

Stefanos Angelidis and Mirella Lapata. 2018. Summarizing opinions: Aspect extraction meets sentiment prediction and they are both weakly supervised. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3675–3686.

Roy Bar-Haim, Lilach Eden, Roni Friedman, Yoav Kantor, Dan Lahav, and Noam Slonim. 2020a. From arguments to key points: Towards automatic argument summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4029–4039.

Roy Bar-Haim, Lilach Eden, Yoav Kantor, Roni Friedman, and Noam Slonim. 2021. Every bite is an experience: Key point analysis of business reviews. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3376–3386.

Roy Bar-Haim, Yoav Kantor, Lilach Eden, Roni Friedman, Dan Lahav, and Noam Slonim. 2020b. Quantitative argument summarization and beyond: Cross-domain key point analysis. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 39–49.

Iman Munire Bilal, Bo Wang, Adam Tsakalidis, Dong Nguyen, Rob Procter, and Maria Liakata. 2022. Template-based abstractive microblog opinion summarization. *Transactions of the Association for Computational Linguistics*, 10:1229–1248.

Su Lin Blodgett, Solon Barocas, Hal Daumé III, and Hanna Wallach. 2020. Language (technology) is power: A critical survey of "bias" in NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5454–5476, Online. Association for Computational Linguistics.

Su Lin Blodgett, Lisa Green, and Brendan O'Connor. 2016. Demographic dialectal variation in social media: A case study of African-American English. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1119–1130, Austin, Texas. Association for Computational Linguistics.

Arthur Bražinskas, Mirella Lapata, and Ivan Titov. 2020. Unsupervised opinion summarization as copycat-review generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5151–5169.

Jiaao Chen and Diyi Yang. 2020. Multi-view sequence-to-sequence models with conversational structure for abstractive dialogue summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4106–4118.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Eric Chu and Peter Liu. 2019. Meansum: a neural model for unsupervised multi-document abstractive summarization. In *International Conference on Machine Learning*, pages 1223–1232. PMLR.

Abhisek Dash, Anurag Shandilya, Arindam Biswas, Kripabandhu Ghosh, Saptarshi Ghosh, and Abhijnan Chakraborty. 2019. Summarizing user-generated textual content: Motivation and methods for fairness in algorithmic summaries. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–28.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

John W Du Bois. 2007. The stance triangle. *Stancetaking in discourse: Subjectivity, evaluation, interaction*, 164(3):139–182.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Alexander Fabbri, Faiaz Rahman, Imad Rizvi, Borui Wang, Haoran Li, Yashar Mehdad, and Dragomir Radev. 2021. ConvoSumm: Conversation summarization benchmark and improved abstractive summarization with argument mining. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6866–6880, Online. Association for Computational Linguistics.

Kyle Glandt, Sarthak Khanal, Yingjie Li, Doina Caragea, and Cornelia Caragea. 2021. Stance detection in covid-19 tweets. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1596–1611.

David Inouye and Jugal K Kalita. 2011. Comparing twitter summarization algorithms for multiple post summaries. In *2011 IEEE Third international conference on privacy, security, risk and trust and 2011 IEEE third international conference on social computing*, pages 298–306. IEEE.

Timo Johner, Abhik Jana, and Chris Biemann. 2021. Error analysis of using bart for multi-document summarization: A study for english and german language. In *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, pages 391–397.

Vijay Keswani and L Elisa Celis. 2021. Dialect diversity in text summarization on twitter. In *Proceedings of the Web Conference 2021*, pages 3802–3814.

Lun-Wei Ku, Yu-Ting Liang, Hsin-Hsi Chen, et al. 2006. Opinion extraction, summarization and tracking in news and blog corpora. In *AAAI spring symposium: Computational approaches to analyzing weblogs*, volume 100107, pages 1–167.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Yang Liu. 2019. Fine-tune bert for extractive summarization. *arXiv preprint arXiv:1903.10318*.

Xinfan Meng, Furu Wei, Xiaohua Liu, Ming Zhou, Sujian Li, and Houfeng Wang. 2012. Entity-centric topic-oriented opinion summarization in twitter. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 379–387.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

Amita Misra, Brian Ecker, and Marilyn Walker. 2016. Measuring the similarity of sentential arguments in dialogue. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 276–287, Los Angeles. Association for Computational Linguistics.

Martin Müller, Marcel Salathé, and Per E Kummervold. 2020. Covid-twitter-bert: A natural language processing model to analyse covid-19 content on twitter. *arXiv preprint arXiv:2005.07503*.

Olubusayo Olabisi, Aaron Hudson, Antonie Jetter, and Ameeta Agrawal. 2022. Analyzing the dialect diversity in multi-document summaries. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6208–6221, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Nils Reimers, Benjamin Schiller, Tilman Beck, Johannes Daxenberger, Christian Stab, and Iryna Gurevych. 2019. Classification and clustering of arguments with contextualized word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 567–578, Florence, Italy. Association for Computational Linguistics.

Anurag Shandilya, Abhisek Dash, Abhijnan Chakraborty, Kripabandhu Ghosh, and Saptarshi Ghosh. 2020. Fairness for whom? understanding the reader's perception of fairness in text summarization. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 3692–3701.

Tejpalsingh Siledar, Jigar Makwana, and Pushpak Bhattacharyya. 2023. Aspect-sentiment-based opinion summarization using multiple information sources. In *Proceedings of the 6th Joint International Conference on Data Science & Management of Data (10th ACM IKDD CODS and 28th COMAD)*, pages 55–61.

Jiaan Wang, Yunlong Liang, Fandong Meng, Zhixu Li, Jianfeng Qu, and Jie Zhou. 2023. Cross-lingual summarization via chatgpt. *arXiv preprint arXiv:2302.14229*.

Xianjun Yang, Yan Li, Xinlu Zhang, Haifeng Chen, and Wei Cheng. 2023. Exploring the limits of chatgpt for query or aspect-based text summarization. *arXiv preprint arXiv:2302.08081*.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

A Appendix

A.1 Dataset

The dataset provided by Glandt et al. (2021) consists of four different stance targets related to COVID-19. The detail of the data distribution can be found in Table 3.

Table 3: COVID-related stance detection data distribution in training, validation and test subsets per target provided in Glandt et al. (2021)

Target	Train	Val	Test
Anthony S. Fauci, M.D.	1464	200	200
Keeping Schools Closed	790	200	200
Stay at Home Orders	972	200	200
Wearing a Face Mask	1307	200	200

A.2 COVID-Stance-BERT Performance

Table 4: Performance of the COVID-Stance-BERT models for stance detection on the targets in the dataset provided by Glandt et al. (2021). The performance is reported in terms of accuracy (Acc) and macro F1 score (F1).

Target	Acc	F1
Anthony S. Fauci, M.D.	0.7714	0.7557
Stay at Home Orders	0.8652	0.8340
Wearing a Face Mask	0.8257	0.8180

A.3 Illustrations of Opinion Diversity

Different illustrations of opinion diversity calculation can be found in Table 5. The opinion precision measures the proportion of important opinions in the generated summary. The opinion recall measures the degree of salient opinion in the source documents that the generated summary contains.

A.4 Stance Examples and Opinion Diversity Calculations

Table 6 provided an example of the stance expressed in the source tweets and the generated summaries using a cluster of tweets under the topic of "Stay at Home Orders". In this example, the source tweets presented three different stances towards the topic. For the generated summaries, Pegasus (Zhang et al., 2020) presented only a neutral stance towards the topic, while ChatGPT successfully captured both support and against stances. We provide how opinion diversity can be measured using the provided example and the calculation detail can be found in Table 7. Applying F1 to these sets of stances yields the results shown in Table 6; from this, we can see that ChatGPT in this example has a higher opinion diversity score.

A.5 Additional Results

Table 8 contains results of opinion precision and recall for the three discussed topics. Based on the result, we found that BART (Lewis et al., 2020) has the best precision score across the three discussed topics. ChatGPT has relatively strong recall scores. BART is better at capturing important opinions, while ChatGPT is relatively better at capturing salient opinions.

Table 5: Illustration of different opinion precision and recall scenarios.

Source documents: **Opinion A**; **Opinion B**			
Model generated summary:	Opinion Precision	Opinion Recall	F1
Good precision, weak recall: **Opinion A**	1.00	0.50	0.67
Good precision, good recall: **Opinion A**; **Opinion B**	1.00	1.00	1.00
Weak precision, weak recall: **Opinion A**; Opinion C	0.50	0.50	0.50
Bad precision, bad recall: Opinion C	0.00	0.00	0.00

Table 6: An example of source tweets and generated summaries with different opinions towards the "Stay at Home Orders" topic. In the source tweets, users expressed support, against and neutral stances towards the topic, forming three different opinions. In the generated summaries, ChatGPT covered two opinions, whereas Pegasus (Zhang et al., 2020) covered a single opinion. In this scenario, ChatGPT has better opinion diversity.

Source tweets	**Stance in source tweets**
'Hey, #billmaher, having people stay home is not "fear," it's *public health.* @USER' '@USER It goes against every fibre of my being to stay home but I have to because my dads high risk. If you die from covid you can't support the movement long-term. Do what you can from home if you're worried'	Support
'@USER Zero protesters but we all are ordered to stay home, and this is a "free" country. What a disgrace to the veterans that died for "free".'	Against
'The UFCW told their members to "please stay home" and expressed their sympathies to members "who lost their workplaces." And this is in a union that has tens-of-thousands of Black members working in essential low-wage jobs in retail and meat packing. Cowards.'	Neutral
Generated summaries	**Stance in generated summaries**
ChatGPT 'People are upset about being told to stay home during the pandemic, but it's important for public health.' 'Some are frustrated that they can't attend church or receive sacraments, while others are protesting and attending funerals.' 'There are concerns about the spread of COVID-19 and the impact on vulnerable individuals.' 'Some are critical of those who are not taking'	Support Against
Pegasus 'I stay home to protect family, friends, strangers .' '...and this is in a union that has tens-of-thousands of black members working in essential low-wage jobs in retail and meat packing.' '...and this is in a state that has much harsher lasting restrictions depending on the needs of each state...look it up during a'	Neutral

Table 7: Illustration of opinion diversity calculation using examples provided in Table 6

Source tweets: Stance-A, Stance-B, Stance-C
System summary: ChatGPT: Stance-A, Stance-B Pegasus: Stance-C
Opinion diversity: ChatGPT: Precision = 2/2; Recall = 2/3; F1 = 0.8 Pegasus: Precision = 1/1; Recall = 1/3; F1 = 0.5

Table 8: Results of opinion precision and opinion recall for various models under different discussed topics.

Events	CDC		Stay at Home Orders		Wearing a Face Mask	
Models	Precision	Recall	Precision	Recall	Precision	Recall
BART	**0.8462**	0.7286	**1.0000**	0.6354	**0.9808**	**0.7308**
Pegasus	0.6111	0.5256	0.9931	0.6424	0.4071	0.3558
T5	0.7137	0.6560	0.9826	0.6285	0.4872	0.4615
ChatGPT	0.7863	**0.7521**	0.9931	**0.6979**	0.8077	0.5000
Copycat	0.6303	0.5192	0.9792	0.5868	0.8782	0.5929
TextRank	0.6303	0.5235	0.9931	0.6319	0.2981	0.2468
LexRank	0.6517	0.5577	**1.0000**	0.6389	0.3942	0.3397
Hybrid TFIDF	0.6496	0.5791	0.9792	0.5833	0.4038	0.3494

Fluency Matters! Controllable Style Transfer with Syntax Guidance

Ji-Eun Han[1] **and Kyung-Ah Sohn**[1,2,*]
[1]Department of Artificial Intelligence, Ajou University
[2]Department of Software and Computer Engineering, Ajou University
`{hanji0514, kasohn}@ajou.ac.kr`
* Corresponding author

Abstract

Unsupervised text style transfer is a challenging task that aims to alter the stylistic attributes of a given text without affecting its original content. One of the methods to achieve this is controllable style transfer, which allows for the control of the degree of style transfer. However, an issue encountered with controllable style transfer is the instability of transferred text fluency when the degree of the style transfer changes. To address this problem, we propose a novel approach that incorporates additional syntax parsing information during style transfer. By leveraging the syntactic information, our model is guided to generate natural sentences that effectively reflect the desired style while maintaining fluency. Experimental results show that our method achieves robust performance and improved fluency compared to previous controllable style transfer methods.

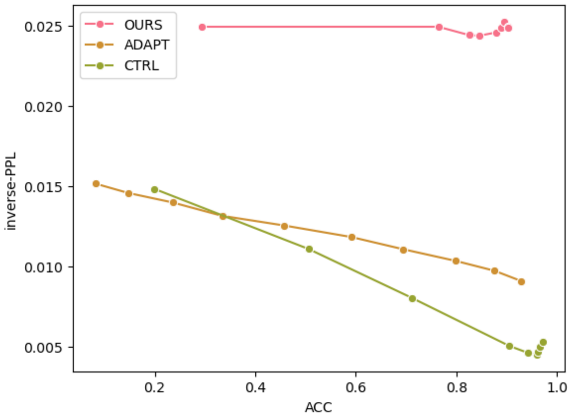


Figure 1: Comparison of inverse-perplexity between controllable style transfer models as the style transfer degree changes. A higher inverse-perplexity score indicates better fluency of generated text. Each dot represents the inverse-perplexity corresponding to style control degrees from 1 to 10. The x-axis represents the transfer accuracy.

1 Introduction

Text style transfer has been garnering increasing interest in the field of natural language generation. Its applicability spans a wide range of tasks, including data augmentation (Chen et al., 2022), stylistic writing for marketing purposes (Kaptein et al., 2015; Jin et al., 2020), and natural chatbot response generation (Kim et al., 2019).

Text style transfer aims to modify a given text to represent a target style attribute. Key considerations for this task include ensuring that the generated text: (i) reflects the desired style attribute, (ii) preserves style-irrelevant content, and (iii) generates a sentence that seems natural to humans. Target style attributes can include various styles such as sentiment, formality, politeness, offensiveness, and genre. In this work, we primarily focus on sentiment as the target style attribute.

Some approaches, such as those by Jhamtani et al. (2017), Carlson et al. (2018), and Wang et al. (2019b), train their models using parallel datasets consisting of pairs of source text and transferred text. However, collecting human-generated transferred text can be both time-consuming and costly. As a result, mainstream research has primarily focused on unsupervised methods that rely solely on source text.

Unsupervised methods for text style transfer can be broadly categorized into two approaches: disentanglement and entanglement. Hu et al. (2018), Shen et al. (2017), and John et al. (2018) proposed models that disentangle content and style in the latent space. However, content and style cannot be entirely separated. As a result, rather than separating content and style, an alternative approach was proposed that uses entangled latent representation.

In the entanglement approach, style information is used to overwrite the latent representation of the source text, resulting in the text reflecting the target style. Multiple approaches have been proposed to achieve this, including the use of back-translation loss (Sennrich et al., 2016a) or a combination of reconstruction, cycle loss, and style classification

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 162–171
July 14, 2023 ©2023 Association for Computational Linguistics

loss in the method proposed by Dai et al. (2019).

Expanding upon the entanglement approach, Wang et al. (2019a) and Kim and Sohn (2020) attempted to control the degree of style during the transfer process. The advantage of these models lies in their ability to generate diversely transferred sentences with varying degrees of style. However, although these models transfer sentences effectively, the generated sentences often lack naturalness. To address this issue, we have endeavored to improve the fluency of our model by incorporating additional syntax information.

Figure 1 highlights that our method substantially outperforms the models proposed by Kim and Sohn (2020) (ADAPT) and Wang et al. (2019a) (CTRL) in terms of perplexity, a metric used to measure the fluency of generated sentences. It should be noted that in the figure, we have inverted the perplexity score, meaning that a higher score indicates better fluency. The comparison reveals that as both the style degree and the accuracy of the transferred style increase, the inverted perplexity score declines in both models. However, our model maintains stable perplexity scores. This suggests that the incorporation of syntax parses helps to preserve the syntactic structure of transferred sentences across diverse levels of accuracy. We found that the other models tend to prioritize generating more tokens containing the target style to enhance accuracy, regardless of fluency. As a result, the generated sentences become less fluent as the style degree increases.

To enhance the fluency of controllable text style transfer, we extract syntax parses from constituency parse trees and encode them into syntactic embeddings. After encoding, we concatenate these embeddings with semantic and style embeddings.

Our experimental results on two datasets demonstrate that our method outperforms several text style transfer baselines. Specifically, our model shows remarkable performance in relation to perplexity. Furthermore, we present an ablation study and qualitative analysis. We also evaluate the syntax preservation capability among controllable models to validate the effectiveness of incorporating syntax parses. Our contributions are suggested as follows:

- We propose a novel approach to enhance the fluency of the controllable text style transfer task. We place emphasis on the fluency of the generated text, ensuring that it sounds natural as if written by a human. By incorporating additional syntax information as a model input, we effectively improve the model's fluency regardless of the transfer strength.
- We validate the effectiveness of our approach by conducting experiments utilizing automatic evaluation metrics. Moreover, we analyze our method with respect to syntax preservation and fluency. The results show that our method helps the model comprehend the syntactic structure of the input sentences and serves as a constraint, steering the model towards generating more natural text.
- We present text-level outputs and compare them to outputs from controllable text style transfer baselines, demonstrating that our model generates fluent sentences while preserving both the syntactic structure and content integrity of the input text.

2 Related Work

Entangle-based text style transfer

One of the approaches employed in unsupervised text style transfer is entanglement. Rather than dividing the latent representation of an input text into content and style components, the entanglement approach directly integrates the input text's latent representation with target style information. Subramanian et al. (2019) use back-translation loss (Sennrich et al., 2016a) to enable learning in two steps: first, the model transfers the input sentence x reflecting the target style s', and second, it reconstructs the output from the previous step with the original style s. Dai et al. (2019) train their model with both reconstruction and cycle loss. Additionally, a style classifier is used to incorporate a style classification loss during training.

Controllable style transfer

Controllable style transfer involves adjusting the magnitude of style transfer strength in the transferred text. Wang et al. (2019a) proposed the Fast-Gradient-Iterative-Modification algorithm to modify the latent representation of the input text to follow the target style. A modification weight is used to control the transfer strength.

Similarly, Kim and Sohn (2020) use the modification weight and train style embeddings to control the style transfer strength. Two style embeddings – positive and negative – are trained in training time. By multiplying these embeddings by the

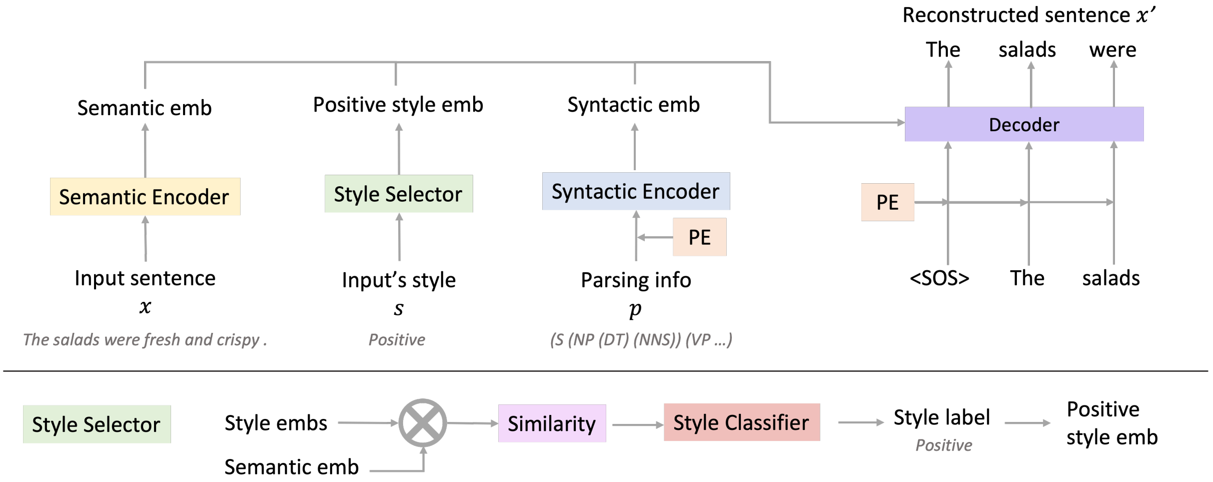


Figure 2: The architecture of our proposed model, consisting of four components: semantic encoder, syntactic encoder, style selector, and decoder. The *upper* figure shows the overall model architecture. The input sentence x, its style label s, and syntax parse p are provided to the model. The semantic encoder, style selector, and syntactic encoder each output an embedding. A concatenated latent representation of the three embeddings—semantic, syntactic, and style—is then fed to the decoder, which generates the reconstructed sentence x'. The *bottom* figure shows Style Selector which selects the style embedding of input x.

modification weight, the model can generate style-controlled text.

Our model adopts the approach by Kim and Sohn (2020) but with the additional step of incorporating style embeddings alongside semantic and syntactic embeddings. The integration of all three types of embeddings, along with the additional syntax information, enables our model to generate more sophisticatedly controlled and natural text.

Syntax-guided generation

Syntax-guided generation generally uses additional syntax information, particularly in machine translation and paraphrasing. In both tasks, syntax information is typically derived from constituency parse trees. After the parse tree has been extracted, it is linearized and then provided to the model along with the input text.

In machine translation, Yang et al. (2020) predict soft target templates and use them to provide syntactical guidance during the translation procedure. Sun et al. (2021) and Huang and Chang (2021) utilize syntax templates to generate syntactically controlled paraphrases that conform to these templates. Sun et al. (2021) use a ranker and retriever to select target parse templates and then generate texts according to the templates. Huang and Chang (2021) train a parse generator to generate diverse syntax templates.

Previous research has explored the importance of syntax in text style transfer. Hu et al. (2021) demonstrated that previous style classifiers were incapable of learning syntax and could worsen models' performance, especially in formality transfer. They employed Graph Convolutional Networks (GCNs) to extract syntactic information and used it to train both syntax-classifier and syntax-encoder. Rather than relying on GCNs for incorporating syntactic information, our approach extracts syntax information from the constituency parse trees. Subsequently, we combine the encoded linearized parse information with semantic and style embeddings.

3 Proposed Method

We formulate the syntax-guided text style transfer as follows: given an input text x, its corresponding style label s, and syntax parse p as model inputs, we train our model using an autoencoder to reconstruct x while preserving the style s. Training the model based on reconstruction is necessary in an unsupervised setting due to the lack of a parallel dataset. The actual style transfer takes place during inference time.

3.1 Model Architecture

Figure 2 shows our overall model architecture. Our model consists of four key components: i) a *semantic encoder* that encodes the input text x; ii) a *syntactic encoder* that encodes the input text's syntax parse p; iii) a *style selector* that chooses the appropriate style embedding se for the input text

x; iv) a *decoder* that generates either reconstructed sentences or transferred sentences.

Semantic encoder. The semantic encoder converts the input text x into a semantic embedding z_{sem}. We represent each token in the input text as $x_1, x_2, ..., x_n$, where n is the number of tokens in x. The semantic encoding process is expressed as follows:

$$z_{sem} = (z_1^{sem}, z_2^{sem}, ..., z_n^{sem}) = Enc_{sem}((x_1, x_2, ..., x_n))$$

where Enc_{sem} represents the semantic encoder. We do not use positional encoding from Transformer (Vaswani et al., 2017) for the semantic embedding, but we apply it to the syntactic encoder. This leads to a semantic embedding that is less affected by word order and thus mainly captures the meaning of the text. In other words, the semantic embedding without positional encoding functions similarly to a bag of words representation. Previous studies have shown that bag of words representation can be effective in various tasks. For example, Xu et al. (2010) demonstrated that generating abstract summaries using only keywords in a bag of words is feasible. In addition, Tao et al. (2021) showed that neural models can successfully reconstruct sentences from an unordered bag of words.

Syntactic encoder. The goal of the syntactic encoder is to produce a syntactic embedding z_{syn} by taking the linearized syntax parse $p = \{p_1, p_2, ..., p_k\}$ as input. This can be expressed as follows:

$$z_{syn} = (z_1^{syn}, z_2^{syn}, ..., z_k^{syn}) = Enc_{syn}((p_1, p_2, ..., p_k))$$

To ensure that the syntax parse includes the information about the order of words, we utilize a Transformer encoder with positional encoding.

Style selector. We define two types of style embeddings: positive and negative. The style selector predicts the style of the input text x and then selects the appropriate style embedding. The process involves three phases, represented at the bottom of Figure 2. In the first phase, we calculate the similarities between each style embedding and the semantic embedding of x. To accomplish this, we use the dot product. In the second phase, we predict the style label of x by utilizing a style classifier C_θ. Finally, in the third phase, we select the final style embedding of x. This is achieved by leveraging the predicted style label in the second phase to select the proper style embedding of x.

Decoder. To generate the reconstructed text x', we concatenate the semantic, syntactic, and style embeddings of the input text x, and feed the resulting concatenated embedding to the Transformer decoder. The decoder then generates x' autoregressively. This process can be represented as follows:

$$\begin{aligned} x' &= (x'_1, x'_2, ..., x'_m) \\ &= Dec(concat(z_{sem}, z_{syn}, z_{style})) \end{aligned}$$

3.2 Training

Since we do not have access to a parallel dataset for this task, we train our model in an unsupervised manner by combining the reconstruction loss from a Transformer-based autoencoder with a style classification loss from a style classifier.

Reconstruction loss. We employ a Transformer-based autoencoder. We calculate the reconstruction loss by comparing the reconstructed sentence to the original sentence. The reconstruction loss is represented as follows:

$$L_{res} = \sum_{i=1}^{n} logP(x'_i = x_i | \bar{x}, p_x, s_x, x'_1, ..., x'_{i-1})$$

where $\bar{x}$ represents an unordered list containing all tokens in the input text x, while p_x represents the syntax parse, and s_x represents the style label of x. Additionally, x'_i represents the generated i-th token, with $x'_1, ..., x'_{i-1}$ being the previously autoregressively generated tokens. By considering the relationships between the semantic, syntactic, and style embeddings, our model gains the ability to reconstruct the input text x.

Style classification loss. In the second phase of the style selector, we utilize a style classifier denoted as C_θ to predict the sentiment of the input text. The classifier is comprised of simple linear layers. Style embeddings SE_i contain two embeddings: a positive embedding and a negative embedding in this task. The similarity between the semantic embedding z_{sem} and each style embedding in SE_i is given as an input of the classifier. Since the gold label is already provided in the training data, we calculate the loss by comparing the predicted label to the gold label y. This procedure is based on the following loss function:

$$L_{style}(C_\theta(Sim(z_{sem}, se_i)), y) = -\sum_{i=1}^{k} \bar{q}_i log(q_i)$$

where C_θ denotes the style classifier, Sim is the similarity calculation performed via dot product and se_i refers to one of the style embeddings in SE_i. $\bar{q}_i$ represents the true style label probability distribution, while q_i represents the predicted style label probability distribution. By optimizing this L_{style} loss function, we train the style embeddings.

Joint training loss. Reconstructing the input text is influenced by the style embedding since the style embedding is concatenated with the semantic and syntactic embeddings. Therefore, we train the autoencoder and the style classifier together using the joint loss as follows:

$$L = L_{res} + L_{style}$$

This approach allows the model to learn to reconstruct the input text while also considering the style information.

3.3 Inference

During inference, the semantic embedding is adjusted to perform style transfer. We use the style embedding that was learned during training. The style transfer operation is represented as follows:

$$z'_{sem} = z_{sem} + w \cdot se'_i$$

where z_{sem} is the semantic embedding, w is a style transfer weight, and se'_i represents the style embedding of the target style. The hyperparameter w controls the degree of style transfer. Following the adjustment of z'_{sem}, it is concatenated with the syntactic and style embeddings before being input into the decoder.

4 Experiment

4.1 Dataset

We evaluate our model with Yelp and Amazon datasets, which are commonly used in unsupervised text style transfer. Table 1 presents the number of data samples for the train, validation, test split in Yelp and Amazon datasets. Each dataset contains human transferred references.

Dataset	Train	Valid	Test
Yelp	443,259	1,000 / style	500 / style
Amazon	554,997	1,000 / style	500 / style

Table 1: Details of Yelp and Amazon datasets.

Yelp. The dataset consists of restaurant reviews on Yelp. The reviews include scores that range from 1 to 5. Each sentence is labeled with the sentiment, either positive or negative according to the score. Sentences with scores of 1 and 2 are labeled as negative, and 4 and 5 are labeled as positive. We use the preprocessed version of the dataset from Li et al. (2018).

Amazon. The dataset contains product reviews from Amazon. The same labeling scheme as the Yelp dataset is used. We use the dataset from He and McAuley (2016).

4.2 Evaluation Metric

We evaluate the performance of our model by comparing it to previous works using three commonly used metrics.

Accuracy measures how well the transferred sentences conform to the target style. To calculate the accuracy, we use a fasttext classifier (Joulin et al., 2016) that is trained on each training dataset. A higher accuracy indicates better model performance.

Content preservation metric evaluates the model's ability to maintain the meaning of the input text, regardless of its stylistic attributes. We measure this using the BLEU score (Papineni et al., 2002), which quantifies how much the transferred sentences overlap with human-written sentences. A higher BLEU score indicates greater similarity between the two sentences. To compute the BLEU-2 score, we utilize the nlg-eval[1](Sharma et al., 2017) package.

Fluency shows how natural the transferred text is. We use perplexity (PPL) as a measure of fluency. In our work, GPT-2 language model (Radford et al., 2019) is used. The GPT-2 model is fine-tuned with the training data of each dataset, and it calculates the 3-gram PPL score.

4.3 Baseline Models

To evaluate the effectiveness of our model, we compare it with several unsupervised text style transfer models. These models can be categorized into two groups based on their ability to control the degree of style transfer.

Uncontrollable models

1) Cross-Align (Shen et al., 2017): this model disentangles style and the content of the input text using a variational autoencoder. It uses an alignment approach to match the input and the transferred text. **2) StyleEmb** (Fu et al., 2017): this model also disentangles the latent into the style and

---

[1] https://github.com/Maluuba/nlg-eval

Model		Yelp			Amazon		
		ACC↑	BLEU↑	PPL↓	ACC↑	BLEU↑	PPL↓
Human reference		73.4	100.0	42.3	42.7	100.0	71.3
(1)	Cross-Align	74.5	21.5	66.9	82.9	8.6	27.5
	StyleEmb	8.8	33.9	61.6	44.5	24.6	114.3
	DeleteAndRetrieve	79.0	16.0	69.4	50.2	42.4	83.3
	Style transformer	84.9	**42.3**	164.0	62.0	42.3	104.6
	RACoLN	87.4	42.2	55.8	**90.1**	**52.1**	100.2
	PromptAndRerank 0-shot	52.2	21.4	65.4	43.8	32.5	91.7
	PromptAndRerank 4-shot	61.2	30.2	57.7	50.0	30.4	68.5
(2)	Controllable-transfer	71.1	35.5	124.2	55.0	36.0	109.6
	Adaptive-StyleEmb	**87.6**	33.9	101.3	74.1	34.19	90.6
	Ours	82.5	18.8	**40.9**	76.8	22.44	**26.8**

Table 2: Evaluation results conducted on the Yelp and Amazon datasets. We divided the models into two groups: (1) uncontrollable models, (2) controllable models. We selected the style transfer weight for models in (2) based on the geometric mean of the accuracy and BLEU score.

the content part using an adversarial network. It uses style embeddings that control the generated styles. **3) DeleteAndRetrieve** (Li et al., 2018): this model first removes the stylistic attributes in the input text and transfers the input by replacing those attributes with retrieved target attribute markers. The model is based on recurrent neural networks. **4) Style transformer** (Dai et al., 2019): unlike other models mentioned above, it overwrites the latent representations with target stylistic attributes. The model architecture is based on Transformer. **5) RACoLN** (Lee et al., 2021): this model is implemented using a gated recurrent unit architecture, and it utilizes a reverse attention mechanism to preserve the content of the input text during style transfer. **6) PromptAndRerank** (Suzgun et al., 2022): pre-trained language models are utilized to generate transferred text. We use the zero-shot and few-shot results from EleutherAI's GPT-J-6B using curly brackets as delimiters.

Controllable models
1) Controllable-transfer (Wang et al., 2019a): it modifies the latent representation of the input text iteratively until the desired degree of style transfer is achieved. **2) Adaptive-StyleEmb** (Kim and Sohn, 2020): it controls the style of the input text by adding style embeddings learned during training to the input latent representation. For these two models, we used pretrained models provided by the authors to get the model outputs.

4.4 Implementation Details

We apply byte pair encoding (Sennrich et al., 2016b) for tokenization and utilize the Stanford CoreNLP parser (Manning et al., 2014) to obtain constituency parses. The maximum token length of the input sentences is 40 and the max token length of linearized syntax parses is 180. Word embeddings are initialized using GloVe (Pennington et al., 2014). The encoder and decoder architecture of our model is implemented with standard Transformer architecture (Vaswani et al., 2017) with its default parameters. We employ the Adam optimizer with a learning rate of 1e-4 and a weight decay of 1e-5. The word dropout probability is set to 0.4. The training process is carried out for 10 epochs.

5 Results

5.1 Quantitative Evaluation

Table 2 presents the results of our quantitative evaluation on the Yelp and Amazon datasets. For controllable text style transfer models, there are multiple output candidates that can be generated by varying the style transfer weight. To select the best candidate, we choose the output with the highest geometric mean of the accuracy and BLEU score.

Our model demonstrates competitive accuracy on the Yelp dataset compared to both controllable and uncontrollable text style transfer models. Notably, our model achieves the lowest PPL score among all the compared models, with a score of 40.9, which is close to the PPL of human reference 42.3. On the Amazon dataset, our model achieves the highest accuracy among controllable models and also shows the lowest PPL score. Overall, these results suggest that our proposed method effectively improves the fluency of transferred text while maintaining high accuracy, although there is

	Negative→Positive
Input	other than that , food here is pretty **gross** .
Controllable-transfer	other than that , food is here pretty **fun makes you delicious** .
Adaptive-StyleEmb	other than that , food here is pretty **good and enjoy warm** .
Ours	other than that , food here is pretty **good** .
	Positive→Negative
Input	the service is **friendly and attentive**.
Controllable-transfer	the service **was not less but then disappointed had the wait fries**.
Adaptive-StyleEmb	the service is **then rude and had old fill that is your worse**.
Ours	the service is **slow and rude**.

Table 3: Comparison of transferred outputs at the text-level in controllable models. Bolded text indicates differences from the input text.

room for improvement in terms of the BLEU score.

5.2 Qualitative Evaluation

In Table 3, we compare transferred outputs from the controllable style transfer models. To select the optimal style transfer weight for the controllable text style transfer models, which is a hyperparameter, we use the same criterion used in 5.1, selecting the weight that shows the highest geometric mean of the accuracy and BLEU score. In the first sample, where a negative sentence is transferred into a positive one, our model is able to convert the token *gross* to *good* while preserving the content of the sentence. In contrast, the other compared models, suggested by Wang et al. (2019a) and Kim and Sohn (2020), generate some tokens that are not present in the original input sentence, such as *fun makes you delicious* and *good and enjoy warm*.

In the second sample, where a positive sentence is transferred into a negative one, our model is able to effectively transfer the sentiment of the input sentence by converting *friendly and attentive* into *slow and rude* while maintaining the naturalness and fluency of the sentence. The other compared models are also able to transfer the input into a negative sentiment. However, their outputs are less natural and fluent compared to ours.

These results indicate that our proposed method is highly effective in transferring the sentiment of the input text to the target style, while ensuring that the content and fluency of the transferred text are maintained.

5.3 Ablation Study

To further demonstrate the importance of incorporating syntax parse information, we conduct an ablation study. Table 4 shows the impact of concatenating syntactic and style embedding on the three evaluation metrics of the transferred text. Our proposed model (Model 1) outperformed the other models across all three metrics. When we excluded the syntactic embedding (Model 2), the resulting transferred text was less fluent, as evidenced by an increase of approximately 5 points in PPL. Similarly, when we removed both syntactic and style embeddings, the performance of the model dropped significantly, particularly in terms of accuracy and PPL. Conversely, adding the syntactic embedding to Model 4 (Model 3) resulted in a substantial decrease in PPL. These results underscore the crucial role of syntax parsing information in generating fluent and natural transferred text.

	Syn emb	Style emb	ACC	BLEU	PPL
(1)	O	O	**82.5**	**18.8**	**40.9**
(2)	X	O	57.3	13.7	45.8
(3)	O	X	21.0	14.6	41.3
(4)	X	X	16.0	16.1	58.6

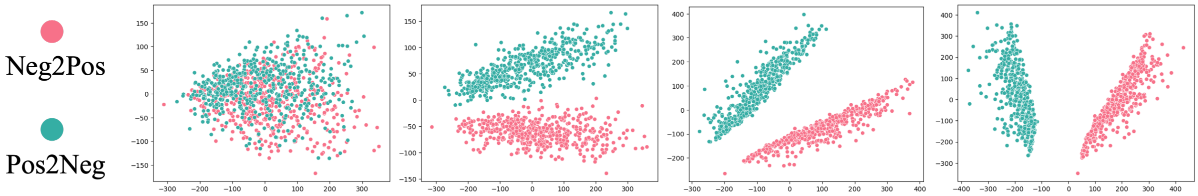
Table 4: Ablation study of the impact of concatenating syntax and style embeddings. We set the style transfer weight w of each model with the geometric mean of accuracy and BLEU score.

Model	Original	Human-transferred
Ours	**92.0**	**67.4**
Controllable-transfer	71.4	56.6
Adaptive-StyleEmb	71.4	56.9

Table 5: Syntax similarity of each model using a metric based on weighted ROUGE scores. It compares the linearized syntax parses of generated sentences to those of reference sentences. Our approach was compared to two controllable text style transfer baselines on the Yelp dataset to demonstrate its ability to preserve syntax while transferring style.

5.4 Syntax Preservation

As demonstrated in 5.2, the controllable style transfer models tend to generate more tokens than the input text in order to incorporate more stylistic



(a) $w = 0$ (b) $w = 1$ (c) $w = 2$ (d) $w = 3$

Figure 3: Semantic embedding visualization differing style weight w.

attributes. While this approach can contribute to higher accuracy, it may compromise the fluency of the output. Therefore, we conducted an experiment to assess the syntax preservation capabilities.

Syntax preservation is determined by the similarity between the syntax parse of the source text and that of the reference text. We employ the syntax parse similarity measure using weighted ROUGE scores (Lin, 2004) proposed in Sun et al. (2021).

$$S(p_{src}, p_{ref}) = a * ROUGE1 + b * ROUGE2 + c * ROUGEL$$

We set $a = 0.2$, $b = 0.3$, $c = 0.5$, following previous work. We applied the style transfer to the test set and compared the transferred output with two types of references: the original test set and human-transferred references. Table 5 demonstrates that the output generated by our model is considerably more similar to both references. This finding suggests that concatenating syntax parses aids the model in retaining the syntactic structures even though the sentiment has transferred.

5.5 Syntax-guided Reconstruction Ability

We evaluate the impact of syntax information on the reconstruction ability of our model, which is trained using reconstruction loss. To assess the pure reconstruction ability, we exclude style information from all models.

Model	ACC	Self-BLEU	PPL
Ours	**3.0**	**90.9**	**36.9**
Controllable-transfer	4.1	78.7	53.4
Adaptive-StyleEmb	5.4	71.5	63.2

Table 6: Impact of syntax parses on model's reconstruction ability evaluated on Yelp dataset.

The results presented in Table 6 highlight the impact of additional syntax information on the reconstruction ability of our model. To evaluate this ability, we use the self-BLEU metric which measures the similarity between the original input text and the reconstructed text, where a higher score indicates better reconstruction ability. Conversely, for accuracy, a lower score indicates better reconstruction ability since it is the accuracy for style transfer. The PPL is calculated using GPT-2 language model. Our findings indicate that incorporating additional syntax parses not only enhances transfer capability but also improves reconstruction ability.

5.6 Embedding visualization

We visualize semantic embeddings in Figure 3 using PCA (Wold et al., 1987) after they are transferred using learned style embeddings. Red dots represent positive sentences that were originally negative, while green dots indicate negative sentences that were originally positive. At w=0, the two colors of dots are entangled. However, as the transfer weight increases, these embeddings gradually separate. At w=3, it is evident that the embeddings are completely transferred and distinctly separated. This implies that the style transfer weight effectively controls the degree of transfer.

6 Conclusion

In this paper, we proposed a controllable, syntax-guided text style transfer model. We improved the fluency of transferred sentences, irrespective of the style transfer strength, by incorporating syntax parses and concatenating their embeddings with semantic and style embeddings. Our approach outperformed previous controllable models on two datasets in terms of consistent PPL scores and natural sentence generation while preserving context. However, our model yielded lower BLEU scores compared to other controllable style transfer models. Future work aims to improve content preservation capabilities while maintaining performance across varying style transfer weights.

Limitation

Our proposed method demonstrates stable perplexity even as the style transfer weight changes, but it yields a lower BLEU score compared to other controllable style transfer models. We hypothesize that the lower BLEU score may be attributed to the fact that the BLEU score calculation is based on just one human-written transferred sentence option per source sentence. This lower score could be a result of our model generating diverse sentences that do not necessarily overlap with the provided human-written references.

Ethics Statement

There are several ethical considerations that must be taken into account when developing a text style transfer model. One important consideration is the risk of the generated text being used to spread hate speech or misinformation. It is also crucial to ensure that the model does not exhibit bias towards a particular demographic, which could result in harmful outcomes. Another potential ethical concern is the misuse of the model for malicious purposes, such as generating negative comments or fake news. These issues need to be addressed to ensure that the development and use of the model align with ethical principles and values.

Acknowledgements

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2022R1A2C1007434), and also by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2023-2018-0-01431) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation).

References

Keith Carlson, Allen Riddell, and Daniel Rockmore. 2018. Evaluating prose style transfer with the bible. *Royal Society Open Science*, 5(10):171920.

Shuguang Chen, Leonardo Neves, and Thamar Solorio. 2022. Style transfer as data augmentation: A case study on named entity recognition. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1827–1841, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ning Dai, Jianze Liang, Xipeng Qiu, and Xuanjing Huang. 2019. Style transformer: Unpaired text style transfer without disentangled latent representation.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2017. Style transfer in text: Exploration and evaluation.

Ruining He and Julian McAuley. 2016. Ups and downs. In *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee.

Zhiqiang Hu, Roy Ka-Wei Lee, and Charu C. Aggarwal. 2021. Syntax matters! syntax-controlled in text style transfer.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2018. Toward controlled generation of text.

Kuan-Hao Huang and Kai-Wei Chang. 2021. Generating syntactically controlled paraphrases without using annotated parallel pairs.

Harsh Jhamtani, Varun Gangal, Eduard Hovy, and Eric Nyberg. 2017. Shakespearizing modern language using copy-enriched sequence-to-sequence models.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, Lisa Orii, and Peter Szolovits. 2020. Hooks in the headline: Learning to generate headlines with controlled styles.

Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. 2018. Disentangled representation learning for non-parallel text style transfer.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification.

Maurits Kaptein, Panos Markopoulos, Boris de Ruyter, and Emile Aarts. 2015. Personalizing persuasive technologies: Explicit and implicit personalization using persuasion profiles. *International Journal of Human-Computer Studies*, 77:38–51.

Heejin Kim and Kyung-Ah Sohn. 2020. How positive are you: Text style transfer using adaptive style embedding. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2115–2125, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Soomin Kim, Joonhwan Lee, and Gahgene Gweon. 2019. Comparing data from chatbot and web surveys: Effects of platform and conversational style on survey response quality. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 1–12, New York, NY, USA. Association for Computing Machinery.

Dongkyu Lee, Zhiliang Tian, Lanqing Xue, and Nevin L. Zhang. 2021. Enhancing content preservation in text style transfer using reverse attention and conditional layer normalization.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: A simple approach to sentiment and style transfer.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Baltimore, Maryland. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units.

Shikhar Sharma, Layla El Asri, Hannes Schulz, and Jeremie Zumer. 2017. Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation. *CoRR*, abs/1706.09799.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment.

Sandeep Subramanian, Guillaume Lample, Eric Michael Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2019. Multiple-attribute text style transfer.

Jiao Sun, Xuezhe Ma, and Nanyun Peng. 2021. AESOP: Paraphrase generation with adaptive syntactic control. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5176–5189, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mirac Suzgun, Luke Melas-Kyriazi, and Dan Jurafsky. 2022. Prompt-and-rerank: A method for zero-shot and few-shot arbitrary textual style transfer with small language models.

Chongyang Tao, Shen Gao, Juntao Li, Yansong Feng, Dongyan Zhao, and Rui Yan. 2021. Learning to organize a bag of words into sentences with neural networks: An empirical study. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1682–1691.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Ke Wang, Hang Hua, and Xiaojun Wan. 2019a. Controllable unsupervised text attribute transfer via editing entangled latent representation.

Yunli Wang, Yu Wu, Lili Mou, Zhoujun Li, and Wenhan Chao. 2019b. Harnessing pre-trained neural networks with rules for formality style transfer. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3573–3578, Hong Kong, China. Association for Computational Linguistics.

Svante Wold, Kim Esbensen, and Paul Geladi. 1987. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52.

Songhua Xu, Shaohui Yang, and Francis Lau. 2010. Keyword extraction and headline generation using novel word features. In *Proceedings of the AAAI conference on artificial intelligence*, volume 24, pages 1461–1466.

Jian Yang, Shuming Ma, Dongdong Zhang, Zhoujun Li, and Ming Zhou. 2020. Improving neural machine translation with soft template prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5979–5989, Online. Association for Computational Linguistics.

ChatGPT for Suicide Risk Assessment on Social Media: Quantitative Evaluation of Model Performance, Potentials and Limitations

Hamideh Ghanadian[1], **Isar Nejadgholi**[2], **Hussein Al Osman**[3]
[1,2,3]University of Ottawa, Ottawa, Canada
[2]National Research Council Canada, Ottawa, Canada
[1,3]`{Hghan053, Hussein.alosman}@uottawa.ca`
[2]`isar.nejadgholi@nrc-cnrc.gc.ca`

Abstract

This paper presents a novel framework for quantitatively evaluating the interactive ChatGPT model in the context of suicidality assessment from social media posts, utilizing the University of Maryland Reddit suicidality dataset. We conduct a technical evaluation of ChatGPT's performance on this task using Zero-Shot and Few-Shot experiments and compare its results with those of two fine-tuned transformer-based models. Additionally, we investigate the impact of different temperature parameters on ChatGPT's response generation and discuss the optimal temperature based on the inconclusiveness rate of ChatGPT. Our results indicate that while ChatGPT attains considerable accuracy in this task, transformer-based models fine-tuned on human-annotated datasets exhibit superior performance. Moreover, our analysis sheds light on how adjusting the ChatGPT's hyperparameters can improve its ability to assist mental health professionals in this critical task.

1 Introduction

According to the World Health Organization (WHO)[1], more than 700,000 people die due to suicide every year. For every suicide, there are many more people who attempt suicide. Furthermore, suicide is the fourth leading cause of death among 15-29 year-olds. According to The US Centers for Disease Control and Prevention (CDC) [2], the rate of suicides per 100,000 increased from 13.5 in 2020 to 14.0 in 2021.

Social media platforms are becoming a common way for people to express their feelings, suffering, and suicidal tendencies. One of the most effective methods recommended by the WHO for preventing suicide is to obtain information from social media and report suicidal ideation to healthcare providers to enable early identification, assessment, and follow-up with affected individuals. Hence, social media provides a significant means for obtaining information to identify individuals who are at risk of committing suicide, allowing for timely detection and intervention (Abdulsalam and Alhothali, 2022)

In recent years, there has been a growing interest in using Natural Language Processing (NLP) techniques for suicide prevention (Fernandes et al., 2018; Bejan et al., 2022). Researchers have developed suicide detection systems to analyze and interpret social media data, including text data. By detecting linguistic markers of distress and other risk factors, these systems can help identify individuals with a risk of suicidality and provide early interventions to prevent such incidents (Vioules et al., 2018). NLP techniques, therefore, offer a promising avenue for suicide prevention efforts, enabling more proactive and effective interventions to support those in need.

This paper investigates the strengths and limitations of ChatGPT, an advanced language model created by OpenAI (Radford et al., 2021), as a tool for suicidal ideation assessment from social media posts. The ChatGPT API provides access to a powerful natural language processing tool that can generate human-like text, answer questions, and perform a variety of other language-related tasks. With ChatGPT, developers can build conversational interfaces, Chatbots, and virtual assistants to interact with users and provide informative responses. However, some studies have highlighted the potential risks and ethical concerns associated with the use of ChatGPT and other language models in sensitive domains, such as mental health and suicide prevention (Zhuo et al., 2023). Therefore, it is crucial to carefully evaluate the use of ChatGPT in such settings to better appreciate its potential and limitations.

Our two research questions to assess the

---

[1] The World Health Organization
[2] The US Centers for Disease Control and Prevention

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 172–183
July 14, 2023 ©2023 Association for Computational Linguistics

reliability of ChatGPT in the suicide prevention task are as follows:

- **RQ1: Can ChatGPT assess the level of suicidality indicated in a written text?**

- **RQ2: Is ChatGPT's performance comparable with transformer-based models trained on a human-annotated dataset?**

The main contributions of this study are as follows:

- Our study examines ChatGPT's performance in evaluating the level of suicidality in Reddit posts using Zero-Shot and Few-Shot Learning techniques. We also compare the results of ChatGPT with two transformer-based models, ALBERT and DistilBERT. Our findings suggest that ChatGPT has potential for suicide risk assessment in Zero-Shot learning, but ALBERT outperforms ChatGPT in this task.

- Our study examines how changing temperature parameters affect ChatGPT's ability on suicide risk assessment. We found that the rate of inconclusive responses generated by ChatGPT is closely linked to changes in the temperature parameter, particularly in the Zero-Shot setting.

- Based on our findings, we can infer that, at lower temperature values, ChatGPT refrains from making a decision for a greater number of instances; however, it exhibits higher accuracy on the subset of instances for which it makes a decision.

These contributions provide a comprehensive evaluation of ChatGPT's performance in this critical application domain and highlight it's potential to be used to assist suicide prevention experts. Our code is available at GitHub[3].

2 Background and Related Work

In this section, we review the related work in suicide ideation detection as well as the generative language technology of the ChatGPT model.

2.1 Suicidal ideation detection and assessment

There are a vast number of research techniques that investigate suicidal ideation and its cause. For instance, clinical methods examine the resting state of heart rate (Sikander et al., 2016) and event-related initiators such as depression (Jiang et al., 2015) as suicidal indicators. Traditional methods use questionnaires, electronic health records, and face-to-face interviews to assess the potential risk of suicide (Chiang et al., 2011).

Several studies indicated the impact of social network reciprocal connectivity on users' suicidal ideation. Hsiung (2007) analyzed the changes in user behavior following a suicide case that occurred within a social media group. Jashinsky et al. (2014) highlighted the geographic correlation between the suicide mortality rates and the occurrence of risk factors in tweets. Colombo et al. (2016) focused on analyzing tweets that contained suicidal ideation, with a particular emphasis on the users' behavior within social network interactions that resulted in a strong and reciprocal connectivity, leading to strengthened bonds between users.

In recent years, NLP researchers have started to analyze users' posts on social media websites to gain an insight into language usage and linguistic clues of suicidal ideation (Chowdhary, 2020; Babulal and Nayak, 2023; Lekkas et al., 2023). Using NLP techniques, suicide-related keyword dictionaries and lexicons are manually built to enable keyword filtering (Varathan and Talib, 2014). The related analysis contains lexicon-based filtering (Sarsam et al., 2021), topic modeling within suicide-related posts (Seah and Shim, 2018), transformer-based models, and unsupervised learning (Linthicum et al., 2019). In line with this field of research, we examine the use of the ChatGPT model for this task, where no labeled data (Zero-Shot setting) or a small labeled dataset (Few-Shot setting) is available.

2.2 ChatGPT

ChatGPT is a state-of-the-art artificial intelligence (AI) Chatbot developed by OpenAI (Radford et al., 2021) that has gained widespread attention for its ability to generate human-like text. The original GPT model was trained on a massive corpus of text data, including books, articles, and web pages, using an unsupervised learning approach. The model's performance on a range of language tasks has since been surpassed by

---

[3] `https://github.com/Hamideh-ghanadian/ChatGPT_for_Suicide_Risk_Assessment_on_Social_Media`

newer models, including GPT-2 (Radford et al., 2019) and GPT-3 (Brown et al., 2020), which have larger training datasets and more sophisticated architectures. However, the ChatGPT model has been fine-tuned on large datasets of conversation data, including social media posts, customer support interactions, and chatbot logs (Dwivedi et al., 2023). ChatGPT differs from prior models as it employs Reinforcement Learning from Human Feedback (RLHF). Unlike supervised learning methods that depend on pre-existing training data, RLHF generates a response to a given input, which is evaluated by a human reviewer. The feedback obtained from the evaluator is used to train the model using reinforcement learning, with the objective of maximizing the reward received (Lambert et al., 2022).

Several recent studies have explored the effectiveness of ChatGPT in a variety of settings, including chatbots and virtual assistants. One study created a corpus named Human ChatGPT Comparison Corpus (HC3) by collecting a set of question-and-answer datasets covering various domains such as finance, medicine, and psychology (Guo et al., 2023). They conducted a comparative analysis between the responses generated by ChatGPT and those provided by humans to investigate the distinguishing features of ChatGPT's responses. In Jeblick et al. (2022) ChatGPT was employed to produce a simplified version of a radiology report, which was then evaluated for quality by radiologists. Another study investigated the proficiency of ChatGPT in answering questions related to the United States Medical Licensing Examination (USMLE) Step 1 and Step 2 exams (Gilson et al., 2022). They found that ChatGPT performed similarly to a third-year medical student.

Bang et al. (2023) proposes a framework for evaluating interactive ChatGPT language learning models using publicly available datasets. They evaluated ChatGPT using 23 datasets covering 8 different NLP tasks, such as summarization, machine translation, sentiment analysis, question answering, etc. They reported that ChatGPT outperforms large language models with Zero-Shot Learning on most tasks and even outperforms fine-tuned models on some tasks.

In this study, we analyze the performance of ChatGPT in predicting suicidal ideation on social media and identifying possible errors that may occur during the process.

3 Dataset

We utilize the University of Maryland Reddit Suicidality Dataset(UMD) (Zirikly et al., 2019; Shing et al., 2018), which is collected from the Reddit platform. Reddit is an online website and forum for anonymous discussion on a wide variety of topics. It is made up of millions of collective forums or groups called subreddits, including the *Depression*[4] and *SucideWatch*[5] subreddits.

The UMD dataset is a collection of Reddit posts and comments created by individuals who expressed suicidal thoughts or behaviors. The dataset contains over 100,000 posts and comments collected from various subreddits, including those related to mental health and suicide prevention, such as "*r/SuicideWatch*". The data was collected over a period of several years and includes the content of the post and comments as well as the location and timing of the posts.

UMD has been repeatedly used by researchers to develop and test natural language processing algorithms and machine learning models to identify and analyze patterns in online communication related to suicide risk (Coppersmith et al., 2018). Ji et al. (2022) proposed a method for improving text representation through the incorporation of sentiment scores based on lexicon analysis and latent topics. Additionally, they introduce the use of relation networks for the detection of suicidal ideation and mental disorders, leveraging relevant risk indicators. Ji et al. (2021) utilized two pretrained masked language models, MentalBERT and MentalRoBERTa, specifically designed to support machine learning in the mental healthcare research field. The authors assess these domain-specific models along with various pretrained language models on multiple mental disorder detection benchmarks. The results show that utilizing language representations pretrained in the mental health domain enhances the performance of mental health detection tasks, highlighting the potential benefits of these models for the mental healthcare research community.

This dataset contains annotations at the user level, utilizing a four-point scale to indicate the severity of the suicide risk: (a) *No risk*, (b) *Low risk*, (c) *Moderate risk*, and (d) *High risk*. According

---

[4] Depression subreddit

[5] SuicideWatch subreddit

to Zirikly et al. (2019), the dataset is divided into three subsets, each containing annotations for a distinct task. In this study, we utilized the subset designated for Task A. This task focuses on risk assessment and involves simulating a scenario in which an individual is suspected to require assistance based on online activity, such as posting to a relevant forum or discussion (e.g., r/SuicideWatch). The objective of the task is to evaluate the individual's risk level based on their online activity. This task requires minimal data, with each user typically having posted no more than a few times on SuicideWatch.

Data Preprocessing: In this study, we only use a subset of the UMD dataset. This subset of the dataset is designed for a specific task (Task A) and includes posts from 21,518 users and is subdivided into 993 labeled users and 20,525 unlabelled users. Out of the 993 labeled users, 496 have at least posted once on the SuicideWatch subreddit. The remaining 497 users are control users (i.e., they have not posted on SuicideWatch or any mental health-related subreddits). Since the provided labels are user-level labels, we aggregated all the posts of each user into a single data point, through the concatenation of all the posts made by a particular user. The dataset is divided into 80% training and 20% testing subsets. The ChatGPT evaluation was conducted solely on the testing subset, comprising 172 instances with proportional representation for each label. Table 1 presents the class sizes of the data subset used in this project.

	No Risk	**Low Risk**	**Moderate Risk**	**High Risk**
UMD Dataset	**26.73 %**	**15.27**	**30.69 %**	**27.28 %**
# of Users	**196**	**112**	**225**	**200**
Training subset	**27.45 %**	**16.39 %**	**31.90 %**	**24.24 %**
# of Users	**154**	**92**	**179**	**136**
Testing subset	**24.41 %**	**11.62 %**	**26.74 %**	**37.20 %**
# of Users	**42**	**20**	**46**	**64**

Table 1: The description of the subset of the UMD Dataset for TASK A defined in Zirikly et al. (2019)

4 Methodology

This paper evaluates the ability of ChatGPT to predict the level of suicidal ideation on the UMD dataset (Zirikly et al., 2019; Shing et al., 2018) and compares it with two fined-tuned classifiers.

4.1 Fine-Tuned Classifiers

We used pre-trained transformer-based language models to train two text classifiers. Transformers are a class of deep learning models, first introduced by Vaswani et al. (2017) in 2017. Researchers build state-of-the-art NLP models using transformer-based architectures because they can be quickly trained on large datasets and studies have shown that they are better at modeling long-term dependencies in natural language text. (Wolf et al., 2020). Moreover, the growth of pre-trained transformer-based structures has made it easier to adapt a high-capacity model trained on a large text to downstream tasks (Devlin et al., 2018; Howard and Ruder, 2018).

We utilize ALBERT [6] and DistilBERT[7] language models and fine-tune them with the UMD dataset to build the classifiers. For implementation, we employed the Huggingface library (Wolf et al., 2019), an open-source library and data science platform that provides tools to build, train and deploy ML models.

The ALBERT model was proposed by Lan et al. (2019) as a variation of BERT that is optimized in terms of memory consumption and training speed. In other words, ALBERT is a more lightweight version of BERT that maintains its high level of accuracy, making it a powerful tool for various NLP applications. The DistilBERT model was proposed by Sanh et al. (2019) which has %40 fewer parameters than BERT and runs %60 faster while preserving over %95 of BERT's performances.

We used the Trainer[8] class from Huggingface transformers[9] for feature-complete training in PyTorch. The hyperparameters were selected based on the default values commonly used in similar studies. The final hyperparameters used in our experiments were Learning Rate= $2e^{-5}$, Batch Size = 4, Dropout Rate = 0.1, and Maximum Sequence Length = 512.

4.2 ChatGPT

The language model utilized by ChatGPT is *gpt-3.5-turbo*[10], which is one of the most advanced language models developed by OpenAI. Chat models accept a sequence of messages as an input and produce a message generated by the model as an output. Although the chat format is primarily

---

[6] AlBERT

[7] DistilBERT

[8] Trainer

[9] Huggingface Transformers

[10] `https://platform.openai.com/docs/models/gpt-3-5`

intended for conversations spanning multiple turns, it is also equally useful for single-turn tasks that do not involve any conversations. We used the *OpenAI Python library*[11] to access the *ChatCompletion* functionality of the *gpt-3.5-turbo* model through its API.

Zero-Shot and Few-Shot Learning: Zero-Shot Learning and Few-Shot Learning are two techniques in Natural Language Processing (NLP) that are used to develop models that can perform tasks without explicit training data (Puri and Catanzaro, 2019). In this project, we evaluate the ability of ChatGPT for Zero-Shot Learning to predict the severity of the suicidality of text for which it has not been explicitly trained. Moreover, we aim to investigate ChatGPT's ability to predict the severity of suicidality, in a Few-Shot Learning setting, where a limited amount of training data is available. In this approach, the model is trained on a small set of examples of the task and then tested on a more extensive set of examples.

Input Message: The primary input for the system is the "message" parameter, which must be an array consisting of message objects. This object includes a "role" (either "system", "user", or "assistant") and a "content" (the message content). A conversation can consist of a single message or can extend over multiple pages. We provide a single message to the system which describes the definitions of suicide severity assessment.

The following example shows the input message for Zero-Shot Learning. In order to be consistent with the instructions given to human annotators of the UMD dataset, we obtained the definitions of each risk category from Shing et al. (2018). These definitions have been adapted from Corbitt-Hall et al. (2016) and specify a four-way categorization scheme of risk.

```
res=openai.ChatCompletion.create(
model="gpt-3.5-turbo",
messages=[{"role": "system", "content":
"""specify the level of the suicidality
    of the text based on the following
    criteria:

0= I do not see evidence that this
    person is at risk for suicide.

1= There may be some factors here that
    could suggest risk, but I do not
    really think this person is at much
    of a risk of suicide.

2= I see indications that there could be
    a genuine risk of this person
    making a suicide attempt.

3= I believe this person is at high risk
    of attempting suicide in the near
    future. Only say number"""},


{"role": "user","content":"""DESIRED
    INPUT TEXT"""}],
temperature=0.1)
```

For Few-Shot Learning with ChatGPT, we use prompt engineering. The prompt consists of two examples for each category (eight in total) drawn from the training dataset to the input message and followed by the same assessment question. For prompt engineering, we drew inspiration from a short course on ChatGPT Prompt Engineering[12] offered by *DeepLearning.AI*. We initiated the prompt construction process with a simple initial prompt and iteratively refined it through multiple rounds of trial and error. This iterative approach allowed us to gradually evolve the prompt, making necessary adjustments based on the observed outcomes and performance of the model. The complete implementation including Zero-Shot Learning, Few-Shot Learning and the fine-tuned classifiers is available on GitHub[13].

Temperature Parameter: The Temperature value in ChatGPT is a parameter that controls the randomness and creativity of the model's responses. To produce a response to a given input message, the model generates a probability distribution over all possible next words or tokens in the response. The temperature parameter affects the probability distribution over the possible tokens at each step of the generation process.

A high temperature value (close to 1) will result in more diverse and unpredictable responses, as the model samples from less likely tokens in the distribution. This can result in more creative and surprising responses but may also increase the likelihood of generating nonsensical or irrelevant text. On the other hand, a low temperature value (e.g. 0.1) will result in more conservative and predictable responses, as the model chooses the most likely tokens in the distribution. This can result in more coherent and on-topic responses but

---

[11] https://github.com/openai/openai-python

[12] ChatGPT Prompt Engineering for Developers

[13] https://github.com/Hamideh-ghanadian/ChatGPT_for_Suicide_Risk_Assessment_on_Social_Media

may be more repetitive or less attractive. The temperature parameter in the ChatGPT allows users to control the balance between creativity and coherence in the model's responses based on their specific needs and preferences.

Inconclusiveness Rate: We define an additional metric, the *Inconclusiveness rate* for further evaluation of ChatGPT in this task. This parameter refers to the proportion of test cases that do not yield a definitive or conclusive result. In other words, it is the rate at which the evidence or information is inconclusive to support a clear decision. To calculate the inconclusiveness rate, after ChatGPT assessed the suicidality risk level of instances, we count all the cases where the ChatGPT reports inconclusive results. An example of an inconclusive response generated by ChatGPT is, *"As an AI, I cannot provide an assessment of the suicidal risk level for this instance"*. Then we divide the number of inconclusive instances by the total number of instances in the test dataset and report this metric as a percentage.

5 Results

In this section, we present the results of our study in accordance with the research questions presented in section 1 based on the test set described in section 3. For evaluation, we report four widely-used metrics in this task, accuracy, precision, recall, and F-score to provide a comprehensive and informative evaluation of the performance of the classification models (Sokolova and Lapalme, 2009). For ChatGPT, we also report the *Inconclusiveness rate* described in Section 4.

5.1 RQ1: Can ChatGPT assess the level of suicidality indicated in a written text?

Zero-Shot Learning: In this section, we present the results of ChatGPT for suicidal ideation prediction with the Zero-Shot Learning approach. The goal of our project is to evaluate the performance of ChatGPT in assessing the level of suicidality of a written text. Furthermore, we use five different temperature values to evaluate the impact of temperature on generated response, and report the inconclusiveness rate of ChatGPT at each temperature. The rest of the metrics are used to evaluate the performance of ChatGPT for the instances in that ChatGPT was able to generate a conclusive answer. Table 2 presents the performance of the ChatGPT in five different temperature values.

Temperature	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**Inconclusiveness Rate**
0.1	**0.88**	**0.57**	**1**	**0.73**	**2.91 %**
0.3	**0.67**	**0.33**	**1**	**0.50**	**2.32 %**
0.5	**0.67**	**0.22**	**0.67**	**0.33**	**1.71 %**
0.7	**0.64**	**0.27**	**1**	**0.43**	**1.16 %**
1	**0.54**	**0.21**	**1**	**0.35**	**0 %**

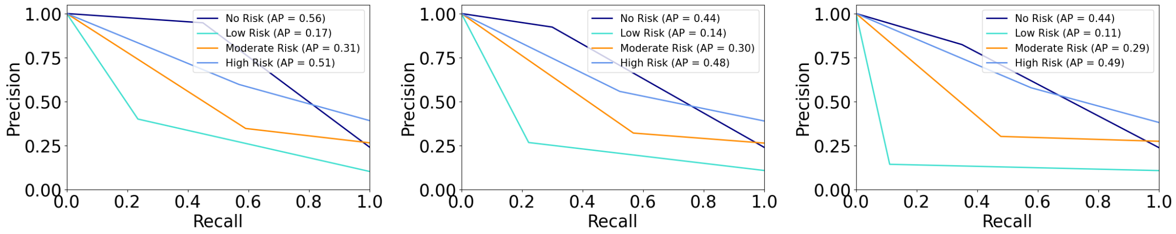
Table 2: Performance and inconclusiveness rate of ChatGPT for Zero-Shot Learning in five different temperature values. The row with the highest F1-score is highlighted.

As presented in Table 2, a higher temperature will result in a more decisive output but with a greater risk of generating errors. Conversely, a lower temperature will result in more indecisiveness, but with a lower risk of errors, i.e., the highest F1-score is achieved with a temperature of 0.1. We observed that ChatGPT's inconclusiveness rate (inability to assess the level of suicidality of instances) is 2.91% for a temperature of 0.1, which is the highest rate for all temperature values. As shown in Table 2, as the temperature value increases, the inconclusiveness rate and F1-score decrease.

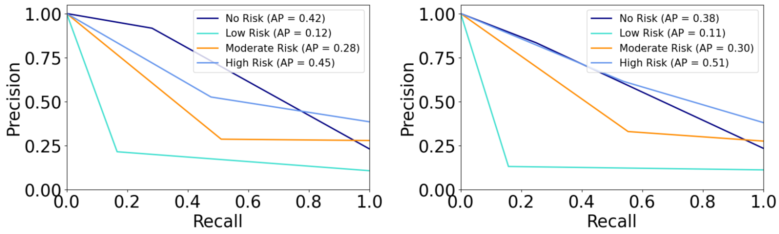
For further evaluation, we present the Precision-Recall (PR) graph of the model at each temperature. The PR graph displays the trade-off between precision and recall for different thresholds used to classify instances. Figure 1 shows the PR curve of ChatGPT for each class. Moreover, it shows the impact of increasing temperature values on predicting the suicidality of the text in each class. As the temperature increases, the area under the PR graph declines. In other words, the graph shows lower values for both precision and recall measures. Moreover, Figure 1 shows that the Average Precision (AP) of ChatGPT in predicting the *No Risk* and *High Risk* classes is higher, compared to the two middle classes, *Low Risk* and *Moderate Risk*.

The bar chart depicted in Figure 2 illustrates which classes are more challenging for ChatGPT for suicidality assessment. Figure 2 shows that at the temperature of 0.1, 3 out of 5 inconclusive instances belong to *Moderate risk* and 2 out of 5 instances belong to *Low Risk* categories.

Few-Shot Learning: We use prompt engineering to implement Few-Shot Learning with ChatGPT. The prompt consists of a few examples from the training dataset, and the model is trained to assess the suicidality level of the text based on the given



(a) Temp=0.1 (b) Temp=0.3 (c) Temp=0.5



(d) Temp=0.7 (e) Temp=1

Figure 1: Precision-Recall graph of the ChatGPT at different temperature values in Zero-Shot setting

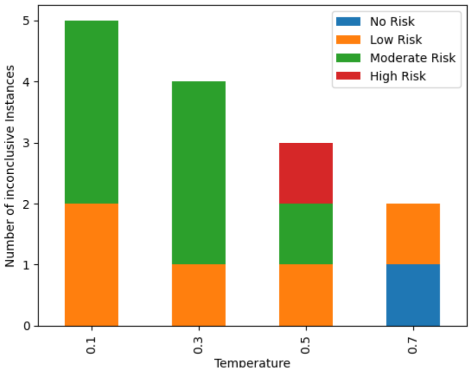


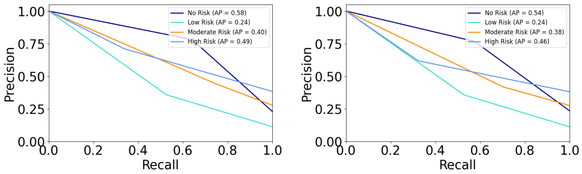
Figure 2: Number of instances for which ChatGPT refrains from making a decision, at different temperature values and for classes *No Risk*, *Low Risk*, *Moderate Risk*, and *High Risk*

criteria. Similar to Zero-Shot Learning, we begin by providing the definitions of each risk category in the prompt, followed by eight training examples and their corresponding labels, with each example and label being separated by a comma and placed in individual paragraphs. The prompt concludes with a request for ChatGPT to provide an assessment based on the given criteria. Table 3 presents the results of ChatGPT in Few-Shot settings at different temperature values.

Temperature	Accuracy	Precision	Recall	F1-Score	Inconclusiveness Rate
0.1	**0.81**	**0.67**	**0.77**	**0.71**	**0.58 %**
0.3	**0.81**	**0.67**	**0.77**	**0.71**	**0.58 %**
0.5	**0.76**	**0.57**	**0.67**	**0.65**	**0.58 %**
0.7	**0.75**	**0.56**	**0.77**	**0.62**	**0 %**
1	**0.75**	**0.56**	**0.77**	**0.62**	**0 %**

Table 3: Performance of ChatGPT for Few-Shot Learning in five different temperature values. The row with the highest F1-score is highlighted.

ChatGPT achieves the highest F1-score at the temperature of 0.1. Furthermore, we observed that the inconclusiveness rate of ChatGPT in Few-Shot Learning was significantly lower compared to Zero-Shot Learning. Additionally, the inconclusiveness rate remained almost constant at different temperature values, indicating that ChatGPT is more confident in generating responses when it is provided with a few examples.



(a) Temperature=0.1 (b) Temperature=1

Figure 3: Precision-Recall graph of the ChatGPT at two extreme temperature values (0.1 and 1) in a Few-Shot settings, for classes *0=No Risk*, *1=Low Risk*, *2=Moderate Risk*, *3= High Risk*.

Figure 3 presents the PR graph of ChatGPT for two extreme temperature values.As presented in Table 3, the precision and recall values for temperature values 0.1 and 1 are not significantly different as it is reflected in the PR curve as well. However, the PR curve of two classes, *Moderate Risk* and *High Risk*, slightly improves by decreasing the temperature.

5.2 RQ2: Is ChatGPT's performance comparable with transformer-based models trained on a human-annotated dataset?

To train a classification model on the UMD datasets, we employed two pretrained transformer-based models, DistilBERT and ALBERT. The performances of these models on the aforementioned dataset are presented in Table 4 and are compared with the results obtained by the ChatGPT model.

Model	Accuracy	Precision	Recall	F1-Score
AlBERT	0.865	0.861	0.865	0.869
DistilBERT	0.77	0.804	0.771	0.745
Zero-Shot ChatGPT (temp=0.1)	0.88	0.57	1	0.73
Few-Shot ChatGPT (temp=0.1)	0.81	0.67	0.77	0.71

Table 4: Comparison of the two transformer-based models with ChatGPT. Fine-tuned ALBERT is highlighted for achieving the highest F-score.

As presented in Table 4, while ChatGPT's performance is comparable to a fine-tuned DistillBERT, it falls considerably short (by 13% for F1-score) compared to a fine-tuned ALBERT model.

6 Discussions and Conclusion

This study focuses on the evaluation of the accuracy and quality of response generated by ChatGPT for the assessment of suicidal ideation levels. The performance of ChatGPT was assessed in Zero-Shot and Few-Shot Learning scenarios. Zero-Shot Learning can be particularly useful when obtaining labeled data is difficult or expensive. In Zero-Shot Learning, ChatGPT achieved an F1-score of 0.73, on our test set (temperature=0.1). These findings demonstrate the potential of ChatGPT as a tool for data annotation, particularly when utilizing a simple prompt design. However, it is important to note that in sensitive tasks such as suicidal ideation detection or assessment, caution must be exercised to ensure accuracy and ethical considerations should be prioritized.

We conducted a Few-Shot Learning experiment to assess the performance of ChatGPT when a few labeled examples of the training data are appended to the prompt. We achieved an F1-Score of 0.71 in Few-Shot Learning (temperature=0.1). In Zero-Shot Learning, the model is able to leverage its existing knowledge to make predictions for new tasks. This approach can be particularly effective when the model needs to generalize to a wide range of possible new tasks. On the other hand, Few-Shot Learning requires the model to learn from a limited amount of training data for each new task. This approach can be more challenging, as the model has to generalize from a small set of examples and may struggle to identify patterns or relationships that are important for the new task.

In this study, we carried out an experiment to examine the impact of the temperature hyperparameter on the performance of ChatGPT. In Zero-Shot Learning, our findings indicate that there is a negative correlation between the F1-Score and the temperature hyperparameter. In other words, as the temperature increases, the model's performance tends to decrease. These results suggest that careful optimization of hyperparameters, such as temperature, is crucial for achieving optimal performance of ChatGPT. In Few-Shot Learning, there is still a negative correlation between the F1-Score and the temperature hyperparameter. However, the change in the F1-Score value is subtle, indicating that the impact of temperature tuning on model performance may not be significant.

Another discovery highlighted in this paper pertains to the examination of the inconclusiveness rate of ChatGPT. There is a trade-off between the inconclusiveness rate and the F1-score in order to optimize the performance of the ChatGPT. In sensitive tasks such as suicide assessment risk, it is crucial to have a highly accurate model that can provide reliable predictions. In some cases, it may be preferable for the model to provide an *"I do not know"* response rather than providing unreliable predictions about suicidality. Careless responses from a suicidal assessment model can have serious consequences, including false positives or false negatives, which can harm individuals at risk. Table 2 shows that the inconclusiveness rate in temperature 0.1 of Zero-Shot Learning is 2.91%, and the F1-Score is 0.73. By increasing the temperature, we have fewer inconclusive instances and yet a lower F1-score over the rest of the responses. Table 3 for Few-Shot Learning shows that the *inconclusiveness rate* becomes almost constant and smaller among different temperature values because the model has learned to generalize based on the limited number of examples provided during training and the model is not able to generate as much variation in response because it may overfit to the training examples. As a result, the model may be less prone to generating random or unexpected responses.

Figure 2 indicates that the inconclusive instances mostly belong to two middle classes *Low Risk* and *Moderate Risk*. These two classes are highly subjective due to the vague boundaries of definitions. For example, the Zero-Shot Learning model was not able to provide an assessment for the following instance: *"I have ups and downs, I've had them for a long time and I don't know why, since December I've been going to therapy, I've been getting meds too and at first they helped suppress the storm of thoughts that won't let me sleep eat and think, I keep finding myself trying to sleep"*. On the other hand, The Few-Shot Learning predicts *High Risk* suicidality level for this instance, and the human experts annotated this instance as *Low Risk*. This example clarifies that generating an "I do not know" answer here can be preferable to a wrong assessment.

To evaluate how well ChatGPT performs compared to other transformer-based models, we conducted an experiment where we fine-tuned two other models, ALBERT and DistilBERT, with the train set of the UMD dataset. The results of this experiment, shown in Table 4, suggest that the ALBERT model reaches promising results with an F1-score of 0.869, outperforming both the DistilBERT and ChatGPT models, with F1-scores of 0.745 and 0.73, respectively. While the ALBERT model achieved the highest score among the three models, it should be noted that it is trained on the UMD dataset for the suicidal assessment task specifically. On the other hand, ChatGPT is trained on a large corpus of text data using a self-supervised learning approach for multiple tasks.

Data collection and annotation are essential but expensive processes in supervised machine learning. Obtaining high-quality labels can be specifically costly and time-consuming in sensitive tasks such as suicide detection. Based on our results, one possible approach to reduce the cost and increase the quality of data annotation is to use ChatGPT in an expert-in-the-loop setting. ChatGPT can assist a human annotator in providing faster and more accurate feedback for a given task. For example, in the case of suicide detection, a human annotator can use ChatGPT to generate responses to various prompts related to suicidal behavior. The annotator can then review the model's output and provide corrections or feedback to refine the output.

7 Future works and Potentials

To ensure the effectiveness and fairness of suicide detection using ChatGPT, it is vital to address biases and generalization issues. Conversational models such as ChatGPT are trained on vast amounts of text data, which may contain biases and reflect societal prejudices. Future research should focus on developing bias mitigation techniques to prevent the model from perpetuating harmful stereotypes or stigmatizing individuals. Additionally, efforts should be made to enhance the generalization capabilities of the model by training it on diverse datasets encompassing various demographics, cultures, and languages. This will enable the model to better understand and identify suicidal ideation across different populations.

Another area for future research is the evaluation of other Chatbots, especially the open-source conversational models. For instance, The Open Assistant project [14], developed by LAION-AI, aims to offer a highly capable chat-based large language model to a wide audience. Through extensive training on diverse text and code datasets, it has acquired versatile capabilities such as answering queries, generating text, translating languages, and even producing creative content. Moreover, Vicuna [15] is an advanced chatbot developed by fine-tuning the Large Language Model Meta AI (LLaMA) using user conversations sourced from ShareGPT. Vicuna is an auto-regressive language model designed to provide natural and immersive conversational experiences which generates highly detailed and well-structured responses, comparable in quality to ChatGPT. By utilizing different models, researchers can contribute to advancing the field of conversational models and unlock their full potential in various applications and domains.

8 Limitations

Our study has several limitations that should be acknowledged. First, the study was conducted on a relatively small test dataset. Future work is needed to assess whether our results are generalizable to larger datasets. Second, while we employed a rigorous methodology for evaluating ChatGPT's performance, we have not measured other safety criteria, such as biases or privacy issues in using

---

[14] `https://open-assistant.io/`

[15] `https://lmsys.org/blog/2023-03-30-vicuna/`

this model. Third, our study focused only on the initial step of suicide risk assessment and did not explore the use of ChatGPT in ongoing monitoring or intervention. Fourth, we are unsure if the UMD dataset has been used in the training of ChatGPT in any capacity since the specifics of the training data of ChatGPT are not disclosed to the public. Future work should focus on creating new datasets to assess the performance of ChatGPT on fully unknown test sets.

It is important to note that despite these limitations, our work represents an important first step in understanding the potential for ChatGPT in suicide risk assessment. Future research should aim to address these limitations and explore the feasibility, safety and effectiveness of ChatGPT in broader clinical settings.

9 Ethical Considerations

For this research, we obtained ethics approval from the research ethics board at the University of Ottawa. Moreover, The UMD dataset was used with authorization from its creators, and we adhered to the terms of use and ethical standards [16] provided by them.

The use of ChatGPT for suicide risk assessment raises several ethical considerations. Firstly, there is the issue of safety and reliability. While ChatGPT has shown promise in natural language processing tasks, it is not infallible and can make mistakes or generate false responses. Due to the sensitivity of the suicide detection task, these errors might lead to severe harm to individuals at risk. Therefore, it is important to 1) thoroughly test and validate the accuracy of the model before using it for suicide risk assessment and 2) deploy it in an expert-in-the-loop setting.

Secondly, there is the issue of privacy and confidentiality. Suicide risk assessment might involve sensitive personal information, and there is a risk that the information processed by the ChatGPT could be mishandled or disclosed to unauthorized parties. It is important to ensure that proper security measures are in place to protect the privacy of individuals who interact with the ChatGPT. Automatic de-identification of data before feeding it to ChatGPT could be a potential solution, but it will bring in its own limitations. In any case, obtaining user consent is crucial before engaging in the automatic processing of data by ChatGPT. It is essential to respect individuals' privacy and ensure that they have given their explicit permission before their data is collected, processed, or shared.

Thirdly, there is the issue of potential psychological harm. Suicide risk assessment can be a sensitive and emotional topic. There is a risk that individuals whose data is assessed by ChatGPT could experience distress or other negative emotions due to the assessment. It is important to have appropriate support mechanisms in place, such as access to mental health professionals or crisis hotlines, to assist individuals who may be in distress.

References

Asma Abdulsalam and Areej Alhothali. 2022. Suicidal ideation detection on social media: A review of machine learning methods. *arXiv preprint arXiv:2201.10515*.

Kanojia Sindhuben Babulal and Bashu Kumar Nayak. 2023. Suicidal analysis on social networks using machine learning. In *The Internet of Medical Things (IoMT) and Telemedicine Frameworks and Applications*, pages 230–247. IGI Global.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*.

Cosmin A Bejan, Michael Ripperger, Drew Wilimitis, Ryan Ahmed, JooEun Kang, Katelyn Robinson, Theodore J Morley, Douglas M Ruderfer, and Colin G Walsh. 2022. Improving ascertainment of suicidal ideation and suicide attempt with natural language processing. *Scientific reports*, 12(1):15146.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Wen-Cheng Chiang, Po-Husn Cheng, Mei-Ju Su, Heng-Shuen Chen, Ssu-Wei Wu, and Jia-Kuan Lin. 2011. Socio-health with personal mental health records: suicidal-tendency observation system on facebook for taiwanese adolescents and young adults. In *2011 IEEE 13th International Conference on e-Health Networking, Applications and Services*, pages 46–51. IEEE.

Prof. K. R. Chowdhary. 2020. Natural language processing for word sense disambiguation and information extraction. *ArXiv*, abs/2004.02256.

---

[16] The University of Maryland Reddit Suicidality Dataset

Gualtiero B Colombo, Pete Burnap, Andrei Hodorog, and Jonathan Scourfield. 2016. Analysing the connectivity and communication of suicidal users on twitter. *Computer communications*, 73:291–300.

Glen Coppersmith, Rebecca Leary, Patrick Crutchley, Alexander Fine, Natural Language Processing for Social Media, and Psychiatry. 2018. Predicting risk of suicide attempts over time through machine learning. *Clinical Psychological Science*, 6(3):345–361.

Darcy J Corbitt-Hall, Jami M Gauthier, Margaret T Davis, and Tracy K Witte. 2016. College students' responses to suicidal content on social networking sites: An examination using a simulated facebook newsfeed. *Suicide and Life-Threatening Behavior*, 46(5):609–624.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yogesh K Dwivedi, Nir Kshetri, Laurie Hughes, Emma Louise Slade, Anand Jeyaraj, Arpan Kumar Kar, Abdullah M Baabdullah, Alex Koohang, Vishnupriya Raghavan, Manju Ahuja, et al. 2023. "so what if chatgpt wrote it?" multidisciplinary perspectives on opportunities, challenges and implications of generative conversational ai for research, practice and policy. *International Journal of Information Management*, 71:102642.

Andrea C Fernandes, Rina Dutta, Sumithra Velupillai, Jyoti Sanyal, Robert Stewart, and David Chandran. 2018. Identifying suicide ideation and suicidal attempts in a psychiatric clinical research database using natural language processing. *Scientific reports*, 8(1):7426.

Aidan Gilson, Conrad Safranek, Thomas Huang, Vimig Socrates, Ling Chi, Richard Andrew Taylor, and David Chartash. 2022. How well does chatgpt do when taking the medical licensing exams? the implications of large language models for medical education and knowledge assessment. *medRxiv*, pages 2022–12.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597*.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Robert C Hsiung. 2007. A suicide in an online mental health support group: reactions of the group members, administrative responses, and recommendations. *CyberPsychology & Behavior*, 10(4):495–500.

Jared Jashinsky, Scott H Burton, Carl L Hanson, Josh West, Christophe Giraud-Carrier, Michael D Barnes, and Trenton Argyle. 2014. Tracking suicide risk factors through twitter in the us. *Crisis*.

Katharina Jeblick, Balthasar Schachtner, Jakob Dexl, Andreas Mittermeier, Anna Theresa Stüber, Johanna Topalis, Tobias Weber, Philipp Wesp, Bastian Sabel, Jens Ricke, et al. 2022. Chatgpt makes medicine easy to swallow: An exploratory case study on simplified radiology reports. *arXiv preprint arXiv:2212.14882*.

Shaoxiong Ji, Xue Li, Zi Huang, and Erik Cambria. 2022. Suicidal ideation and mental disorder detection with attentive relation networks. *Neural Computing and Applications*, 34(13):10309–10319.

Shaoxiong Ji, Tianlin Zhang, Luna Ansari, Jie Fu, Prayag Tiwari, and Erik Cambria. 2021. Mentalbert: Publicly available pretrained language models for mental healthcare. *arXiv preprint arXiv:2110.15621*.

Nengzhi Jiang, Yanyu Wang, Lin Sun, Yuping Song, and Hongwei Sun. 2015. An erp study of implicit emotion processing in depressed suicide attempters. In *2015 7th International Conference on Information Technology in Medicine and Education (ITME)*, pages 37–40. IEEE.

Nathan Lambert, Louis Castricato, Leandro von Werra, and Alex Havrilla. 2022. Illustrating reinforcement learning from human feedback (rlhf). *Hugging Face Blog*. Https://huggingface.co/blog/rlhf.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Damien Lekkas, Jenny Y Oh, Rina E Matsumura, Amelie D Perrier, and Nicholas C Jacobson. 2023. Profiling the digital mosaic of uncensored suicidal thought and behavior: A theory-driven network analysis of online written expression.

Kathryn P Linthicum, Katherine Musacchio Schafer, and Jessica D Ribeiro. 2019. Machine learning in suicide science: Applications and ethics. *Behavioral sciences & the law*, 37(3):214–222.

Raul Puri and Bryan Catanzaro. 2019. Zero-shot text classification with generative language models. *arXiv preprint arXiv:1912.10165*.

Alec Radford, Ilya Sutskever, Rewon Child, Gretchen Krueger, and Jong Wook Kim. 2021. Chat with gpt: Improving language generation and task-oriented dialogue. `https://openai.com/blog/chatgpt-plus`.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Samer Muthana Sarsam, Hosam Al-Samarraie, Ahmed Ibrahim Alzahrani, Waleed Alnumay, and Andrew Paul Smith. 2021. A lexicon-based approach to detecting suicide-related messages on twitter. *Biomedical Signal Processing and Control*, 65:102355.

Jane HK Seah and Kyong Jin Shim. 2018. Data mining approach to the detection of suicide in social media: A case study of singapore. In *2018 IEEE international conference on big data (Big data)*, pages 5442–5444. IEEE.

Han-Chin Shing, Suraj Nair, Ayah Zirikly, Meir Friedenberg, Hal Daumé III, and Philip Resnik. 2018. Expert, crowdsourced, and machine assessment of suicide risk via online postings. In *Proceedings of the Fifth Workshop on Computational Linguistics and Clinical Psychology: From Keyboard to Clinic*, pages 25–36.

Daud Sikander, Mahnaz Arvaneh, Francesco Amico, Graham Healy, Tomas Ward, Damien Kearney, Eva Mohedano, Jennifer Fagan, John Yek, Alan F Smeaton, et al. 2016. Predicting risk of suicide using resting state heart rate. In *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pages 1–4. IEEE.

Marina Sokolova and Guy Lapalme. 2009. A systematic analysis of performance measures for classification tasks. *Information processing & management*, 45(4):427–437.

Kasturi Dewi Varathan and Nur Fadhillah Qasim Abu Talib. 2014. Suicide detection system based on twitter. *2014 Science and Information Conference*, pages 785–788.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

M Johnson Vioules, Bilel Moulahi, Jérôme Azé, and Sandra Bringay. 2018. Detection of suicide-related posts in twitter data streams. *IBM Journal of Research and Development*, 62(1):7–1.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Terry Yue Zhuo, Yujin Huang, Chunyang Chen, and Zhenchang Xing. 2023. Exploring ai ethics of chatgpt: A diagnostic analysis. *arXiv preprint arXiv:2301.12867*.

Ayah Zirikly, Philip Resnik, Özlem Uzuner, and Kristy Hollingshead. 2019. CLPsych 2019 shared task: Predicting the degree of suicide risk in Reddit posts. In *Proceedings of the Sixth Workshop on Computational Linguistics and Clinical Psychology*.

Unsupervised Domain Adaptation using Lexical Transformations and Label Injection for Twitter Data

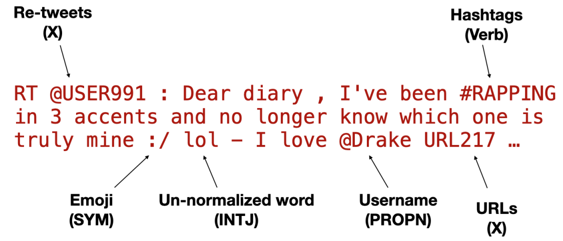
Akshat Gupta, Xiaomo Liu, Sameena Shah
J.P.Morgan AI Research
`{akshat.x.gupta, xiaomo.liu, sameena.shah}@jpmorgan.com`

Abstract

Domain adaptation is an important and widely studied problem in natural language processing. A large body of literature tries to solve this problem by adapting models trained on the source domain to the target domain. In this paper, we instead solve this problem from a dataset perspective. We modify the source domain dataset with simple lexical transformations to reduce the domain shift between the source dataset distribution and the target dataset distribution. We find that models trained on the transformed source domain dataset performs significantly better than zero-shot models. Using our proposed transformations to convert standard English to tweets, we reach an unsupervised part-of-speech (POS) tagging accuracy of 92.14% (from 81.54% zero shot accuracy), which is only slightly below the supervised performance of 94.45%. We also use our proposed transformations to synthetically generate tweets and augment the Twitter dataset to achieve state-of-the-art performance for POS tagging.

1 Introduction

In a typical machine learning setting, training, development and test sets are usually carved out of the same data collection effort. In doing this, we caveat our models with an implicit assumption - the deployment dataset should belong to the same distribution as the training dataset. This is rarely the case and we see significant drops in performance when the model is deployed. The mismatch between the deployment data distribution, or *target domain*, and the training data distribution, or *source domain*, is known as domain shift (Ramponi and Plank, 2020; Ruder and Plank, 2018) and the process of adapting to target domain distributions is known as domain adaptation (Blitzer et al., 2006).



```
RT @USER991 : Dear diary , I've been #RAPPING
in 3 accents and no longer know which one is
truly mine :/ lol - I love @Drake URL217 …
```

Figure 1: The Anatomy of a Tweet - This figure shows lexical features of a tweet like hashtags, user-mentions, emojis, re-tweets.

The most widely studied domain adaptation methods are model-centric methods (Ramponi and Plank, 2020), where parts of the model, including the feature space, the loss function or even the structure of the model are altered (Blitzer et al., 2006; Pan et al., 2010; Ganin et al., 2016; März et al., 2019). Data-centric methods (Ramponi and Plank, 2020) usually involve some form of bootstrapping and pseudo-labelling of the target domain data (Abney, 2007; Cui and Bollegala, 2019; Ruder and Plank, 2018; Gupta et al., 2021). A popular data-centric domain adaptation method is data selection, which is an intermediate training step that aims to select a subset of data that is closest to the target domain (Moore and Lewis, 2010; Axelrod et al., 2011; Aharoni and Goldberg, 2020; Iter and Grangier, 2021). We refer the reader to domain adaptation surveys in natural language processing for a detailed overview (Ramponi and Plank, 2020; Chu and Wang, 2018; Jiang, 2013; Margolis, 2011).

To the best of our knowledge, none of the works we encounter in literature address the fundamental reason behind the need for domain adaptation - *domain shift*. If we are able to transform the source domain dataset such that the domain mismatch between the source domain and the target domain is reduced, while being able to exploit the annotations of the source domain corpus, then the models trained on such a transformed source domain data will naturally perform better on the target domain. This is the main motivation behind our work. All model-centric and data-centric domain adaptation

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 184–193
July 14, 2023 ©2023 Association for Computational Linguistics

ORIGINAL : And , um , i 'm just super excited .
[CCONJ PUNCT INTJ PUNCT PRON AUX ADV ADV ADJ PUNCT]
TRANSFORMED : ***en , um , im 'm jst #super excited . URL4699***
[CCONJ PUNCT INTJ PUNCT PRON AUX ADV ADV ADJ PUNCT X]

ORIGINAL : How do people look at and experience art ?
[ADV AUX NOUN VERB ADP CCONJ VERB NOUN PUNCT]
TRANSFORMED : ***RT @USER5899 hw do people look n andd experience art ?***
[X X ADV AUX NOUN VERB ADP CCONJ VERB NOUN PUNCT]

Figure 2: Examples of original sentences from GUM dataset and how they look like after different Lexical Transformations and Label Injections. Note the POS labels injected post transformations.

methods can be applied on top of our proposed method and are complementary to it.

In this paper, we transform the source domain dataset to resemble the target domain dataset more closely through a series of transformations. In our case, the source domain consists of standard English sentences and the target domain consists of tweets. Through these transformations, we are able to improve the zero-shot POS tagging accuracy by 10.39% when averaged over five different BERT models. Also, when we combine the transformed data to augment the original target dataset, we achieve state-of-the-art POS tagging performance on the target dataset.

2 Lexical Transformations and Label Injections

Standard English sentences and Tweets have both semantic and lexical differences. Tweets are more likely to be subjective and polarized (appendix A.1). On the other hand, tweets also contain unique lexical features like acronyms, emojis, user mentions, retweets, hashtags, as shown in Figure 1, and can be used as different parts of speech (Table 6, appendix A.2). In this paper, we focus on converting standard English sentences into tweets by making lexical transformations and injecting labels wherever required. Example transformations are shown in Figure 2.

Lexcial transformations add target domain-specific lexical features to the source domain dataset such that these properties are 'distributionally' conserved. For example, when our target domain is Twitter, we expect Tweets to contain emojis. We can measure the distributional presence of emojis in tweets, like the percentage of tweets that on average contain emojis or how they are distributed within the sentence, i.e. if they are more likely to occur in the beginning, middle, or end of a sentence. In lexical transformations, we add these distributional properties to the source domain sentences. Since we are adding these features to an annotated dataset, we also inject the label of the lexical feature wherever required. The process is discussed in detail in section 4. The resulting sentences are almost indistinguishable from Tweets, as can be seen in Figure 2. It is not trivial to inject these lexical features into the standard English sentences as the same feature can correspond to multiple parts of speech, as shown in Table 6.

Dataset	Split	Sentences	Tokens
GUM	Train	6,917	124,923
TBv2	Train	1,639	24,753
	Test	1,201	19,911

Table 1: Table showing the dataset statistics for GUM and Tweebank datasets (TBv2). In this paper, all models are tested on the test set of TBv2, which is our target domain set. For compelete statistics, refer to A.1.

3 Datasets

In this paper, we work with two annotated POS tagging datasets. For standard English, we use the GUM (Georgetown University Multilayer Corpus) dataset (Zeldes, 2017). For Twitter data, we use Tweebank (TBv2) (Liu et al., 2018) dataset. We choose these two datasets because they are both labelled using the universal dependencies (Nivre et al., 2016) framework, thus each of the datasets have identical 17 POS tags. The dataset statistics are shown in Table 1.

The GUM dataset acts as our source domain dataset and is about 5 times larger than TBv2, which is our target domain dataset. GUM dataset is made up of articles and interviews from Wikinews, instructional articles from wikiHow and travel guides from Wikivoyage (Zeldes, 2017). The GUM dataset contains longer sentences compared to the Tweebank dataset. The Tweebank dataset gets higher average polarity and subjectivity scores when compared to the GUM dataset. The experiments analysing dataset properties are shared in appendix A.1.

4 Experiments

In this section, we present four different types of Lexical Transformations and corresponding label injection methods for Twitter as target domain. All transformations are performed on the GUM train-split (the standard English dataset). Models trained on the transformed dataset are tested on

Method	POS	BERT-base	RoBERTa-base	BERT-large	RoBERTa-Large	BERTweet
Zero Shot	-	79.74 (0.26)	80.87 (0.19)	81.52 (0.23)	81.83 (0.23)	80.73 (0.25)
Emoji Injection	SYM	80.12 (0.19)	80.85 (0.43)	81.08 (0.68)	81.77 (0.51)	81.59 (0.60)
ILN	-	80.66 (0.23)	79.43 (0.18)	81.33 (0.13)	80.95 (0.45)	80.46 (0.34)
@,#ht	PROPN	81.89 (0.06)	80.87 (0.08)	82.28 (0.04)	81.92 (0.19)	82.15 (0.14)
RT,#ht,url	X	89.33 (0.08)	88.09 (0.12)	89.76 (0.09)	88.95 (0.21)	88.89 (0.17)

Table 2: This table shows the performance boost for unsupervised domain adaptation from standard English to Tweets when the four different lexical transformations are used individually.

the TBv2 test set (the Twitter dataset). All experiments shown in this paper report accuracy scores on TBv2 test set, in accordance with previous works (Owoputi et al., 2013; Meftah et al., 2019, 2020; Nguyen et al., 2020). Each experiment is repeated five times and the mean score is reported with standard deviations reported in brackets.

All experiments in this paper are done using the Huggingface implementations of different BERT models. We use five different BERT models, the original BERT-base-uncased and BERT-large-uncased (Devlin et al., 2018) models, the RoBERTa-base and RoBERTa-large models (Liu et al., 2019) and the BERTweet model (Nguyen et al., 2020).

4.1 Zero-Shot Experiments

We begin by training the model on the original GUM train-split and testing it on the TBv2 dataset. This experiment sets our baseline for unsupervised domain adaptation as it represents zero-shot application of a model trained on standard English, and then applied to tweets. The results are shown as the Zero Shot results of Table 2.

4.2 Emoji Injections

Social media text is filled with emojis and emoticons. In this paper, we refer to both as Emojis. To convert standard English sentences to Tweets, we inject emojis into standard English sentences. Emojis belong to the ‘SYM:symbol’ class in the universal dependencies framework, which is inserted as the label for the injected emoji in the source domain dataset.

To place an emoji within a standard English sentence, we first randomly select an emoji from a pre-decided list of emojis. Then we place the emoji inside a sentence according to a Gaussian distribution which is fit to the location of occurrence of emojis in a tweet. We randomly add emojis to 25% of the sentences in the GUM dataset. The different experiments done to reach the above methodology for emoji injection are described in appendix A.3. The results for emoji injection are shown in Table 2.

4.3 Inverse Lexical Normalization

Lexical normalization is a common task where non-standard English tokens are corrected to standard English (Han et al., 2013). This includes expanding acronyms like *wru -> where are you* and correcting spelling errors. In this paper, we convert standard English to its lexically un-normalized version. We call this process Inverse Lexical Normalization (ILN). To do so, we use a lexical normalization dataset (Baldwin et al., 2015) as a dictionary lookup and create a mapping between lexically correct words and their un-normalized version. For example, *you* is written in various different ways including *u*, *uuuu*, *youuuu*. We randomly replace the correct tokens with their un-normalized versions 75% of the times. The ablation experiments for this lexical transformation are shown in A.4. The POS tag of the original word is retained in the transformation. BERT-base observes maximum improvement with ILN (Table 2).

4.4 Converting PROPN to User-Mentions and Hashtags

Another distinguishing lexical features of Tweets is the use of user-mentions and hashtags. In this transformation, we randomly pick existing proper nouns in the GUM dataset and convert them into user-mentions or hashtags by adding an ’@’ or ’#’ symbol in front of the token, with a probability of 50% and 20% respectively. The existing proper noun labels are kept for the converted tokens. The ablations for this transformation can be found in appendix A.5. We see consistent improvements with this transformation for all models except RoBERTa models (Table 2).

4.5 Injecting ReTweets, URLS, user-mentions and hashtags as X

The ‘X’ part of speech tag or the *other* category in the universal dependency framework (Nivre et al.,

Type	Train Dataset	BERT-base	RoBERTa-base	BERT-large	RoBERTa-Large	BERTweet
Unsupervised	GUM	79.74 (0.26)	80.87 (0.19)	81.52 (0.23)	81.83 (0.23)	80.73 (0.25)
	GUM-T (UDA)	**91.82 (0.07)**	**90.85 (0.08)**	**92.14 (0.12)**	**90.86 (0.61)**	**90.99 (0.24)**
Supervised	TBv2	93.88 (0.05)	93.00 (0.03)	94.45 (0.04)	93.85 (0.08)	93.85 (0.09)
	TBv2 + GUM	94.31 (0.06)	94.16 (0.06)	94.51 (0.05)	94.61 (0.08)	94.71 (0.08)
	TBv2 + GUM-T	**94.81 (0.02)**	**94.84 (0.06)**	**95.01 (0.05)**	**95.04 (0.04)**	**95.21 (0.03)**

Table 3: This table shows the final Unsupervised Domain Adaptation performance using our proposed data transformations. All these models are test on the TBv2 test set and trained on the datasets described above. We combine the transformed data with the original dataset to achieve state-of-the-art results on the Tweebank test set.

2016) is defined as - "*The tag X is used for words that for some reason cannot be assigned a real POS category. It should be used very restrictively*". While the 'X' POS tag is used sparingly in standard English, a large number of tokens in tweets fall into this category. In this transformation, we insert re-tweets (at the beginning of sentences), urls (usually at the back of the sentences) and hashtags (randomly sampled from a Gaussian calculated from tweets). Re-tweets are added in 30% of the sentences, URL's are added in 60% of the sentences and hashtags are added in 10% of the sentences. The ablations can be found in appendix A.6. The label 'X' is added with these lexical transformations.

We see massive improvements across the board by adding this lexical transformation. This is because the 'X' POS tag, which is probably the most under-utilized tag when dealing with standard English, becomes vital when dealing with tweets. All Re-tweets, URL's and many hashtags and user mentions fall under this category.

5 Results

We now combine all transformations together, as shown in Table 3. The first section in Table 3 represents our unsupervised domain adaptation results. The first row in Table 3 shows models trained on the original GUM dataset (standard English) and tested on TBv2 test set, representing zero-shot domain transfer results. The GUM-T dataset represents the transformed dataset containing all the previously described transformations. Models trained on the GUM-T dataset represent our unsupervised domain adaptation performance, which improves on the zero-shot POS tagging accuracy by 10.39%, without ever seeing a single tweet (when averaged over all five models). The class-wise F1 improvements for different POS tags are shown in Table 12. BERT-base witnesses the maximum gain from our transformations (12.08%) and performs better than RoBERTa-large and BERTweet.

System	POS Accuracy
(Owoputi et al., 2013)	94.6
(Meftah et al., 2019)	94.95
(Nguyen et al., 2020)	95.2
BERTweet [TBv2 + GUM-T] (ours)	**95.21 (0.03)**

Table 4: Table showing the dataset statistics for GUM and Tweebank datasets (TBv2). In this paper, all models are tested on the test set of TBv2, which is our target domain set.

The second section in Table 3 contains supervised experiments where the training dataset contains tweets. We check the efficacy of our proposed transformations as a synthetic data generation process. We first augment the TBv2 dataset with the original GUM dataset and compare it with the improvements we get when TBv2 is combined with GUM-T. We see that the combination of TBv2 and GUM-T datasets outperforms all supervised models and gives 1.6 to 8 times larger performance boost over augmenting with the original GUM dataset. The TBv2 + GUM-T combination reaches (a saturated) state-of-the-art maxima for POS tagging on the TBv2 dataset, as shown in Table 4.

6 Conclusion

A lot of focus in literature has been given to converting noisy social media text to standard English. In our work, we convert standard English into noisy social media-like text using simple lexical transformations and show that it can be used as an effective unsupervised domain adaptation and data augmentation method. The fundamental idea behind our work is to reduce domain shift by transforming the source domain into the target domain. We present experiments for these transformations between standard English and Twitter domain and find an average accuracy boost for POS tagging of 10.39% across 5 different BERT models, without ever using a single tweet for supervised training.

Acknowledgements

This paper was prepared for informational purposes in part by the Artificial Intelligence Research Group of JPMorgan Chase & Co and its affiliates ("J.P. Morgan") and is not a product of the Research Department of J.P. Morgan. J.P. Morgan makes no representation and warranty whatsoever and disclaims all liability, for the completeness, accuracy, or reliability of the information contained herein. This document is not intended as investment research or investment advice, or a recommendation, offer, or solicitation for the purchase or sale of any security, financial instrument, financial product, or service, or to be used in any way for evaluating the merits of participating in any transaction, and shall not constitute a solicitation under any jurisdiction or to any person if such solicitation under such jurisdiction or to such person would be unlawful.

© 2022 JPMorgan Chase & Co. All rights reserved.

7 Limitations

In this paper, we focus on lexical transformations between source domain and target domain to reduce the domain shift between them. To do this, we identify unique lexical features in the target domain and place them in the source domain so that the transformed domain is distributionally similar to the target domain. But there are also semantic differences between the two domains in terms of content, domain-specific jargon, and other nuances. This work does not take into account those transformations. Also, we use Twitter as the target domain for our work. While the general principles of our work are applicable to any source-target domain pairs, the transformations discussed in this work cater broadly to social media text, and specifically to Twitter data. The generalizability to other target domains has not been tested in this paper and remains a topic of further investigation.

In this paper, we work with a POS tagging dataset. POS tagging is a token level task where we classify each token as belonging to a certain category. We feel that because POS tagging is dependent on each token in the sentence, domain transfer affects this task most adversely. Sequence classification tasks like sentiment analysis that only require a high level representation of the entire sentence to make classification decisions might witness different levels of improvement. The current method needs to be tested for other task types, including sequence classification tasks like sentiment analysis, or generative tasks like question answering and text summarization. This was beyond the scope of a short paper.

References

Steven Abney. 2007. *Semisupervised learning for computational linguistics*. Chapman and Hall/CRC.

Roee Aharoni and Yoav Goldberg. 2020. Unsupervised domain clusters in pretrained language models. *arXiv preprint arXiv:2004.02105*.

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 conference on empirical methods in natural language processing*, pages 355–362.

Timothy Baldwin, Marie-Catherine De Marneffe, Bo Han, Young-Bum Kim, Alan Ritter, and Wei Xu. 2015. Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 126–135.

John Blitzer, Ryan McDonald, and Fernando Pereira. 2006. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pages 120–128.

Chenhui Chu and Rui Wang. 2018. A survey of domain adaptation for neural machine translation. corr abs/1806.00258 (2018). *arXiv preprint arXiv:1806.00258*.

Xia Cui and Danushka Bollegala. 2019. Self-adaptation for unsupervised domain adaptation. *Proceedings-Natural Language Processing in a Deep Learning World*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030.

Akshat Gupta, Sargam Menghani, Sai Krishna Rallabandi, and Alan W Black. 2021. Unsupervised self-training for sentiment analysis of code-switched data. *arXiv preprint arXiv:2103.14797*.

Bo Han, Paul Cook, and Timothy Baldwin. 2013. Lexical normalization for social media text. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(1):1–27.

Dan Iter and David Grangier. 2021. On the complementarity of data selection and fine tuning for domain adaptation. *arXiv preprint arXiv:2109.07591*.

Jing Jiang. 2013. A literature survey on domain adaptation of statistical classifiers, 2008. *URL: http://sifaka. cs. uiuc. edu/jiang4/domain_adaptation/survey*.

Yijia Liu, Yi Zhu, Wanxiang Che, Bing Qin, Nathan Schneider, and Noah A Smith. 2018. Parsing tweets into universal dependencies. *arXiv preprint arXiv:1804.08228*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Anna Margolis. 2011. A literature review of domain adaptation with unlabeled data. *Tec. Report*, pages 1–42.

Luisa März, Dietrich Trautmann, and Benjamin Roth. 2019. Domain adaptation for part-of-speech tagging of noisy user-generated text. *arXiv preprint arXiv:1905.08920*.

Sara Meftah, Nasredine Semmar, Mohamed-Ayoub Tahiri, Youssef Tamaazousti, Hassane Essafi, and Fatiha Sadat. 2020. Multi-task supervised pretraining for neural domain adaptation. In *Proceedings of the Eighth International Workshop on Natural Language Processing for Social Media*, pages 61–71.

Sara Meftah, Youssef Tamaazousti, Nasredine Semmar, Hassane Essafi, and Fatiha Sadat. 2019. Joint learning of pre-trained and random units for domain adaptation in part-of-speech tagging. *arXiv preprint arXiv:1904.03595*.

Robert C Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 conference short papers*, pages 220–224.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. Bertweet: A pre-trained language model for english tweets. *arXiv preprint arXiv:2005.10200*.

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 380–390.

Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. 2010. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th international conference on World wide web*, pages 751–760.

Alan Ramponi and Barbara Plank. 2020. Neural unsupervised domain adaptation in nlp—a survey. *arXiv preprint arXiv:2006.00632*.

Sebastian Ruder and Barbara Plank. 2018. Strong baselines for neural semi-supervised learning under domain shift. *arXiv preprint arXiv:1804.09530*.

Amir Zeldes. 2017. The gum corpus: Creating multilayer resources in the classroom. *Language Resources and Evaluation*, 51(3):581–612.

Dataset	Split	Sentences	Tokens
GUM	Train	6,917	124,923
	Dev	1,117	19,654
	Test	1,096	19,911
TBv2	Train	1,639	24,753
	Dev	7,10	11,759
	Test	1,201	19,911

Table 5: Table showing the complete dataset statistics for GUM and Tweebank datasets (TBv2).

A Appendix

A.1 Dataset

In this paper, we work with two part-of-speech (POS) tagging datasets. The GUM dataset (Zeldes, 2017), which is made up of standard English sentences from different wiki-sources like wikiNews, wikiHow etc., and the Tweebankv2 (TBv2) dataset (Liu et al., 2018), which consists of tweets. The GUM dataset acts as our source domain dataset, while TBv2 acts as our target domain dataset.

The number of sentences and the number of tokens in each dataset are given in Table 5. Figure 3 shows the sentence length distribution between the GUM and the TBv2 dataset. We see that the GUM dataset contains longer sentences when compared to the TBv2 dataset. The mean tokens per sentence for GUM is 18.06 (std = 13.3) whereas the mean tokens per sentence for the TBv2 dataset is 15.10 (std = 7.74). This shows us that TBv2 not only has shorter sentences, but their spread is also shorter.

We measure average subjectivity and polarity scores for the two datasets to indicate semantic differences. We find higher average subjectivity and polarity scores for the TBv2 dataset compared to the GUM dataset. To measure these, we use the spaCY textblob [1] library to calculate subjectivity and polarity scores. Polarity is scored between -1 and 1 indicating the sentiment expressed in the sentence. We take the absolute value of the polarity scores since we consider both positive and negative sentiment since we are interested in the presence and absence of polarity in tweets. The mean polarity score for the TBv2 dataset was 0.23 compared to 0.13 for the GUM dataset. Subjectivity is scored between 0 and 1, with 0.0 being very objective and 1.0 being very subjective. TBv2 had a mean subjectivity score of 0.36 compared to 0.27 for the GUM dataset.

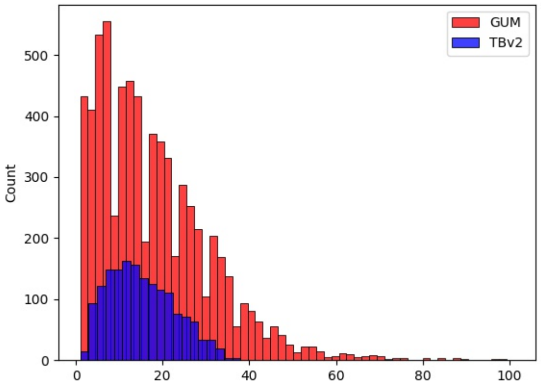


Figure 3: Sentence length distribution between GUM and Tweebankv2 (TBv2) dataset. We see that GUM has more longer and shorter sentences compared to TBv2.

Lexical Features	Associated POS tags
Emoji	SYM
Re-Tweets	X
URLs	X
User Mentions	X, PROPN
Hashtags	X, VERB, PROPN ...
Un-normalized words	VERB, INTJ, ADP ...

Table 6: Table showing the different lexical features seen in tweets and the corresponding part of speech tags the features can take.

A.2 Lexical Features

Some of the lexical features specific to tweets that we are concerned with in this paper are - emojis, re-tweets, user-mentions, hashtags, URL's and un-normalized tokens. It is not trivial to inject these into the standard English sentences as same lexical feature can correspond to multiple parts of speech. This can also be seen in Figure 1, where user-mentions are used both for the category 'X' as well as proper nouns. A more detailed description of the different lexical features and the corresponding parts of speech the features can take can be seen in Table 6. Lexical features like user-mentions can take two parts of speech, where hashtags and un-normalized words can essentially be any part of speech.

A.3 Emoji Injections Ablation

Emoji Injection is a lexical transformation where we insert emojis in standard English sentences such that the distributional properties of the transformed text resemble a Twitter dataset. Lexical emoji in-

---

[1] `https://spacy.io/universe/project/spacy-textblob`

Emoji Injection Method	POS Tagging Accuracy
Zero-shot	79.746 (0.256)
RANDOM-PLACEMENT	79.103 (0.275)
LOCATION-SAMPLING	80.125 (0.192)

Table 7: Comparison between random-emoji injection and location-sampling based emoji injection. We find that location-sampling performs significantly better than random placement.

jection is done in two steps:

- Emoji Selection - Sample an emoji from a pre-selected list of emojis
- Emoji Placement - Select a location in the standard English sentence to place the selected emoji

Both these steps can be done randomly or based on a particular distribution. The selection step can be done by selecting an emoji based on the distribution of its occurrence in Twitter feeds. Although in this paper, in the emoji selection step, we select an emoji randomly from a pre-decided list of emojis.

Similarly, the emoji placement step can be done in two ways. The selected emoji can be placed randomly anywhere in the sentence. This is called RANDOM-PLACEMENT. The alternative is to place the emojis in a sentence based on a certain distribution and sample the location of placement from that distribution. This method of placement is called LOCATION-SAMPLING. The distribution is found by studying the locations at which different emojis occur in a Twitter feed and fitting the location of their occurrence to a Gaussian distribution. We use the TBv2 train-split to calculate the distribution parameters. We experiment with these two methods for emoji injection for the BERT-base model by injecting tweets in 25% sentences in the GUM dataset. The models are trained on the transformed dataset and tested on the TBv2 test set. The results are shown in Table 7. We find that LOCATION-SAMPLING is significantly superior to the RANDOM-PLACEMENT method of emoji-injection.

We also experimented with different thresholds for emoji injection. We found that injecting emojis into a larger number of sentences hurts the model performance as shown in Table 8. Thus, we do emoji injection with a 25% probability.

Emoji Injection Method	POS Tagging Accuracy
Zero-shot	79.746 (0.256)
EI (25%)	80.123 (0.192)
EI (50%)	79.671 (0.331)
EI (75%)	79.473 (0.389)

Table 8: Comparison between different probability thresholds of emoji injection. We find a 25% probability of adding emojis to a sentence performs optimally.

ILN Method	POS Tagging Accuracy
Zero-shot	79.746 (0.256)
ILN (25%)	80.329 (0.327)
ILN (50%)	80.504 (0.292)
ILN (75%)	80.668 (0.236)

Table 9: Comparison between different probability thresholds for inverse lexical normalization. This probability threshold is for converting each token in a sentence to its un-normalized version.

A.4 Inverse Lexcical Normalization Ablation

Inverse Lexical Normalization (ILN) aims to convert standard English text into its un-normalized versions. This includes converting correct spellings to their noisy versions as used in social media and converting certain texts to corresponding acronyms. Some examples of such a conversion would be converting *you -> u, that - dat, how are you -> hru.*

We do this by using the dataset released by (Baldwin et al., 2015) for lexical normalization. We use the training set as a dictionary and find mappings between the lexically-correct tokens and their noisy usage in social media. When a word in this dictionary is found in the standard English sentence, it is converted into its un-normalized version with a probability of 75%. The ablation experiments with BERT-base are shown in Table 9.

A.5 Injecting User Mentions and Hashtags as PROPN - Ablation

User mentions and hashtags are often used as proper nouns (PROPN) as shown in the two examples below :

- *#FOLLOW us #CHECKOUT the multi - talented Spanglish Pop Singer Model @USER779 aka Lady Boom Boom URL107*
- *Today I went to watch #Metallica #themostamazingconcertever*

In the first tweet, @USER779 mention is used as a proper noun. In the second example #Metallica is used as a proper noun followed by another hashtag

PROPN Injection Method	POS Tagging Accuracy
Zero-shot	79.746 (0.256)
@(25%), #(10%)	81.604 (0.064)
@(50%), #(20%)	81.896 (0.061)
@(75%), #(30%)	81.742 (0.055)

Table 10: Comparison between different probability thresholds for converting proper nouns present in standard English sentence to user mentions and hashtags.

X-Injection Method	POS Tagging Accuracy
Zero-shot	79.746 (0.256)
RT(30%), URL(60%), #(10%)	89.334 (0.079)
RT(60%), URL(60%), #(20%)	89.253 (0.094)
RT(90%), URL(60%), #(30%)	89.173 (0.081)

Table 11: Comparison between different probability thresholds for injecting re-tweets, URLs and hashtags into a standard English sentence as the POS label ‘X’.

which refers to a totally different part-of-speech. In this transformation, we convert pre-existing proper nouns in standard English sentences into user mentions or hashtags. In a brief analysis of Twitter feed, we found that user mentions were more common than hashtags. Thus we start by randomly changing proper nouns into user mentions with a probability of 25% and into hashtags with a probability of 10%. The ablation experiments with BERT-base model are shown in Table 10.

A.6 Injecting Re-Tweets, URLS, User Mentions and Hashtags as X - Ablation

Re-tweets involving user mentions are separate from when user mentions are used as proper nouns and are classified in the ’X:other’ POS category. URLs and some hashtags also fall into this category. Examples of tweets containing these lexical features can be seen in Figure 1. Injecting these features is simpler than the other lexical features and yet results in the largest improvements. Re-tweets are almost always present at the beginning of a tweet. URLs are almost always present at the end of the tweet. We make a pre-selected list of certain hashtags that fall into the ’X:other’ POS tag category and place them randomly in a sentence. We experiment with the relative probability of such injections in Table 11.

A.7 Combining All Lexical Data Transformations

When we combine all lexical data transformations, we achieve significant boost in performance on the Twitter dataset. When a model trained on the GUM dataset (standard English, source domain) is tested on the Tweebankv2 test set (Twitter dataset, target domain), we see that the model has about 81.52% accuracy using BERT-large for POS tagging (Table 3, first row, Unsupervised). When we use all lexical transformations to transform standard English dataset to Twitter like sentences, called GUM-T, we achieve 92.14% accuracy, and see a significant boost of 10.62% over the zero-shot performance. This shows us that our simple lexical data transformations give the model a massive boost without training on actual tweets annotated for POS tagging. Our lexical data transformations can be used both for unsupervised domain adaptation and data augmentation, as shown in Table 3.

POS class	Zero-Shot F1	Transformed F1	Tokens
NOUN	0.85	0.87	2669
NUM	0.80	0.92	304
PROPN	0.63	0.97	1716
SYM	0.53	0.79	209
VERN	0.87	0.93	1985
X	0.01	0.94	2056

Table 12: Class-wise F1 improvement for unsupervised domain adaptation for BERT-large model for selected classes. Zero-shot F1 shows the class-wise F1 scores before applying lexical data transformations. Transformed F1 shows the class-wise F1 score for unsupervised domain adaptation of BERT-large model after application of lexical data transformations.

A.7.1 The ‘X:other’ POS class for Twitter

The class-wise F1 score improvements in BERT-large for unsupervised domain adaptation are shown in Table 12. We see significant improvements for all POS classes. The improvement is massive for the ‘X’ POS class because this class works very differently in standard English and tweets. Tweets contain a lot of hashtags, URLs, and re-tweets, which is completely different from standard English. Thus, the ‘X’ POS class is the biggest lexical differentiator between standard English and how people communicate on Twitter. This is also why the performance of a POS tagger trained on standard English dataset performed abysmally, with and F1 score of 0.01.

A.8 Lexicalally Transformed Sentences

Some examples of the lexicalally transformed sentences from standard English to tweets are shown in Figure 4. The examples show different features including emojis, user mentions, re-tweets, URLs and lexically incorrect tokens.

Aesthetic Appreciation and Spanish Art :
RT @USER642 aesthetic appreciation nd spanish art :

Insights from Eye - Tracking
insights fr eye - tracking URL8384

Andrew Beresford a.m.beresford@durham.ac.uk Durham University , United Kingdom
andrew @USER7094 :p @USER7824 durham @USER9005 , united @USER2305 URL9988

Daniel Smith daniel.smith2@durham.ac.uk Durham University , United Kingdom
@USER4467 smith @USER9289 durham university , united kingdom

How do people look at and experience art ?
RT @USER5899 hw do people look n andd experience art ?

Which elements of specific artworks do they focus on ?
which elements of specific artworks do hey focus o ? 🎧 URL6857

Do museum labels have an impact on how people look at artworks ?
do museum labels havr an impact o hw ppe look n artworks ? URL6150

Figure 4: Figure shows various examples of lexicalally transformed standard English sentences. The sentence in the bold font corresponds to the lexicalally transformed sentence for the original standard English sentence written directly above it regular font.

A.9 Average Runtimes, Hyperparameters and Hardware

All experiments were performed on a single Tesla T4 GPU with 16GB GPU memory in a system with 16GB RAM. The run-time for base models per epoch was approximately 2 minutes for the Tweebank train-split and 6 minutes for the GUM train-split. For large models, the time taken per epoch was approximately 6 minutes for Tweebank train-split and 18 minutes for GUM train-split. The best performance and best dev-accuracy were chosen. We kept a batch size of 32, a learning rate of 1e-5 and maximum sequence length of 256. All models are trained for 25 epochs. We run each configuration 5 times and report the mean scores and standard deviation.

Transformer-based cynical expression detection in a corpus of Spanish YouTube reviews

Samuel González-López
Technological University of Nogales
Nogales, Sonora, México
`sgonzalez@utnogales.edu.mx`

Steven Bethard
University of Arizona
Tucson, Arizona, USA
`bethard@email.arizona.edu`

Abstract

Consumers of services and products exhibit a wide range of behaviors on social networks when they are dissatisfied. In this paper, we consider three types of cynical expressions – negative feelings, specific reasons, and attitude of being right – and annotate a corpus of 3189 comments in Spanish on car analysis channels from YouTube. We evaluate both token classification and text classification settings for this problem, and compare performance of different pre-trained models including BETO, SpanBERTa, Multilingual Bert, and RoBERTuito. The results show that models achieve performance above 0.8 F1 for all types of cynical expressions in the text classification setting, but achieve lower performance (around 0.6-0.7 F1) for the harder token classification setting.

1 Introduction

Consumers of services and products actively engage through social networks when they are dissatisfied, exhibiting a wide range of behaviors. Encinas and Cavazos (2021). Encinas presents a classification of dysfunctional consumer behaviors: mild behaviors such as rudeness, complaints, skepticism, or tantrums; moderate behaviors such as manifestations of cynicism, attempts at manipulation, or inappropriate comments and foul language; and intense consumer behaviors such as fraud, theft, verbal aggression, or revenge.

We focus on cynical expressions of consumers, specifically in comments written in videos on the Youtube platform. Cynicism is a negative attitude with a broad or specific focus and comprises cognitive, affective, and behavioral components (Chylinski and Chu, 2010). Consumer cynicism can generate feelings of betrayal and deception, leading to anger and the desire to stop purchasing products or services from the source that generates their anger (Encinas and Cavazos, 2021). Within expressions of cynicism, we focus on the following specific expressions:

Negative Feelings where consumers reflect negatively on a product, usually in a subjective way that is influenced by their personal experiences.

Specific Reasons where consumers identify the specific aspects or components of a product to which their negative feelings are directed, for instance, fuel efficiency or seating comfort.

Attitude of being right where consumers express their rejection of the product and in contrast assert their own correctness.

Such expressions come in many forms, written both by users who have directly experienced the products on which they are commenting, and by users who have yet to consume or use the product being discussed. Table 1 provides some examples of these three types of cynical expressions.

The contributions of our research are as follows:

- We collected and annotated 3189 comments in Spanish from the Youtube platform, achieving kappa of 0.834, 0.859, and 0.752 for negative feeling, specific reasons, and attitude of being right, respectively.
- We explore detection of cynical expressions both as a token classification task and as a text classification task.
- We compare a variety of pre-trained models to be fine-tuned for this task, including SpanBERTa, BETO, Multilingual BERT, and RoBERTuito.

2 Related work

The analysis of feelings is a broad field of research. Some behaviors in social media, such as offensive language, sarcasm, irony, and aggressiveness, correspond to the negative sentiment side. Cynical expressions are related to the negative aspect and

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 194–201
July 14, 2023 ©2023 Association for Computational Linguistics

Spanish Example	English Translation	Expression
"La probé y se maneja bien, tiene buenos acabados, pero No me convenció su diseño, como que es difícil de digerir, siento que va ser de esos que dan el viejazo muy pronto ”.	‘I tried it, and it handles well, and has good finishes, but I was not convinced by its design , as it is difficult to digest, I feel that it will be one of those that give the old age very soon. ’	Negative feeling and specific reason
“que equivocado esta señor yo tengo una Cadillac y creame que es muy superior a Mercedes y a BMW su motor y el lujo es muy superior y es mas grande que sus rivales ”.	‘ How wrong you are sir. I have a Cadillac, and believe me, it is far superior to Mercedes and BMW; its engine and luxury are far superior, and it is bigger than its rivals. ’	Negative feeling and Attitude of being right
“La suspensión trasera la cagaron , mejor una suspensión trasera independiente como las generaciones anteriores. Pero los Mazdetos felices con cualquier cosa”.	‘The rear suspension they crapped up, better an independent rear suspension like previous generations. But Mazdetos are happy with anything.’	Negative feeling and Attitude of being right
“Que versión más rara , le falta muchísimos extras y la versión srx o limited es la verdadera full, 7 airbags, control de estabilidad, bloqueo de diferencial trasero, camara de retroceso etc.”	‘What a rare version, it lacks many extras, and the srx or limited version is the true full, 7 airbags, stability control, rear differential lock, rearview camera, etc. ’	Negative feeling and specific reason

Table 1: Examples of cynical expressions: red color corresponds to Specific Reason expression: green color refers to Negative Feeling; blue color corresponds to Attitute to being right cynical expression.

are specific elements that determine consumer cynicism.

In the field of Irony, we found a study (Al-Mazrua et al., 2022) on an annotated corpus of tweets with 8089 positive texts in the Arabic language. This work uses machine learning and deep learning models and reports a 0.68 accuracy with the SVM algorithm. The Fleiss's Kappa agreement value was 0.54, a moderate level. One of the challenges in this work was detecting implicit phrases as part of the Irony. In (Maladry et al., 2022) a corpus of 5566 tweets for the Dutch language, 2783 were labeled as irony. This work reported for a binary classification task a 78.98% for implicit irony and 78.88% for explicit and implicit sentiment. The SVM model performed better compared to the BERT model. Under approaches such as CNN with Embeddings (FastText, Word2vec) (Ghanem et al., 2020), the Irony was worked on. This study analyzed monolingual and multilingual architectures in three languages, with the monolingual configuration performing better. A second approach, RCNN-RoBERTa, consisting of a pretrained RoBERTa transformer followed by bidirectional long-term memory (BiLSTM), achieved 0.80 F1 on the SemEval-2018 dataset and 0.78 F1 on the Reddit Politics dataset (Potamias et al., 2020).

Very close to Irony, we find Sarcasm in the text. A paper (Alnajjar and Hämäläinen, 2021)for the Spanish language shows a dataset of text aligned to audio and video. This paper reports SVM matching results of 89% using the text alone, 91% combining audio and text, and 93.1% combining text, audio, and video. This multi-modal task is interesting since sarcasm analysis becomes domain-specific. However, adding video could generalize sarcasm detection by movements and gestures. In (Peled and Reichart, 2017) the identification of sarcasm is based on the ability to generate a non-sarcastic text from an original sarcastic text. e.g., from the sarcastic text "how I love Mondays" is obtained "how I hate Mondays" or "I really hate Mondays". In this work, the sarcasm dataset contains 3000 sarcastic tweets, each with five different non-sarcastic interpretations, and the algorithm based on Machine translation places particular emphasis on feeling words.

At a higher level, we find the feeling of aggression. Aggression can be direct or indirect and is a feeling of anger that results in hostile behavior. An analysis (Lepe-Faúndez et al., 2021) with 22 models combining the lexical and machine learning approach was performed on three corpora for Spanish (Chilean, Mexican, and Chilean-Mexican). The results show that the best performance was for the Chilean corpus with 0.89 F1, while for the Mexican corpus, it was 0.839 and 0.850 for the Chilean-Mexican combination. However, this paper highlights a higher agreement of the corpus with Chilean terms. With BERT models and an assembly strategy, a dataset tagged as non-aggressive, covertly aggressive, and overtly aggressive was classified. The assemblies achieved two percentage points higher F1-score than single models (Risch et al., 2019). Employing the same dataset but with other training features, for instance, the amount of abusive/aggressive/offensive words or the presence of hash-tags, obtain an accuracy of 73.2 % (Kumar et al., 2020).

Our research focused on consumer cynicism, annotating a new corpus for several previously unexplored cynical expressions. And unlike most previous work, which focused on the English language, our analysis of consumer cynicism focuses on the Spanish language.

3 Dataset

The corpus was generated from YouTube comments downloaded from new car analysis channels[1]. The comments were filtered, taking into account two requirements: comments must contain at least ten words, and comments must have a minimum of 5 likes. The goals of these constraints were to ensure sufficient text to judge the presence or absence of cynical expressions, and to focus on comments deemed to be relevant to the discussion. The result was a total of 3189 comments[2]. Table 2 shows some statistics of the corpus.

Two annotators were given a set of annotator guidelines containing examples of each type of cynical expression. One of the annotators was a master's student in computer science, and the second was a university teacher in computer science. The annotation guidelines had three sections: an introduction to the topic of consumer cynicism, examples of each type of cynical expression, and examples of what the annotation should look like using different colors to mark the text. The annotators were also given a description of the context of the research and a video tutorial[3] on how to use the annotation tool. Figure 1 shows a screenshot of the annotation tool.

Cynical expressions	Count	Kappa
Negative Feelings	644	0.834
Specific Reasons	381	0.859
Attitude of being right	605	0.752
Suspicions	155	0.550

Table 2: Dataset statistics. Of 3189 annotated comments, only 1785 were coincidences among the annotators, distributed in each category.

A group of 50 pre-training comments was used to familiarize the annotators with the annotation task. To calculate agreement between annotators, we counted two annotations as matching if the text segment of one annotator was contained within the segment the other annotator. A coverage of 90% of the matching was established. If it was lower, the text was considered a disagreement, and the document was not considered for the machine learning models. Table 2 shows agreement for the different types of cynical expressions.

We discarded the cynical expression Suspicions for having a low level of agreement, and then had the annotators annotate the remaining comments.

4 Methodology

We considered two cynicism detection tasks:

Token classification We frame the cynicism detection task using the standard inside-outside-beginning format for token-by-token classification.

For evaluation, a 10-fold cross-validation method was performed. For each cynical expression, the following BERT models were run: SpanBERTa, mBERT, and BETO. The parameters with the best performance were: 160 epochs, 3 × 10-5 of the learning rate, and a batch size of 16. The number of epochs during the fine-tuning was 20, 80, 160, and 200. The batch was computed with 16 and 32 sizes.

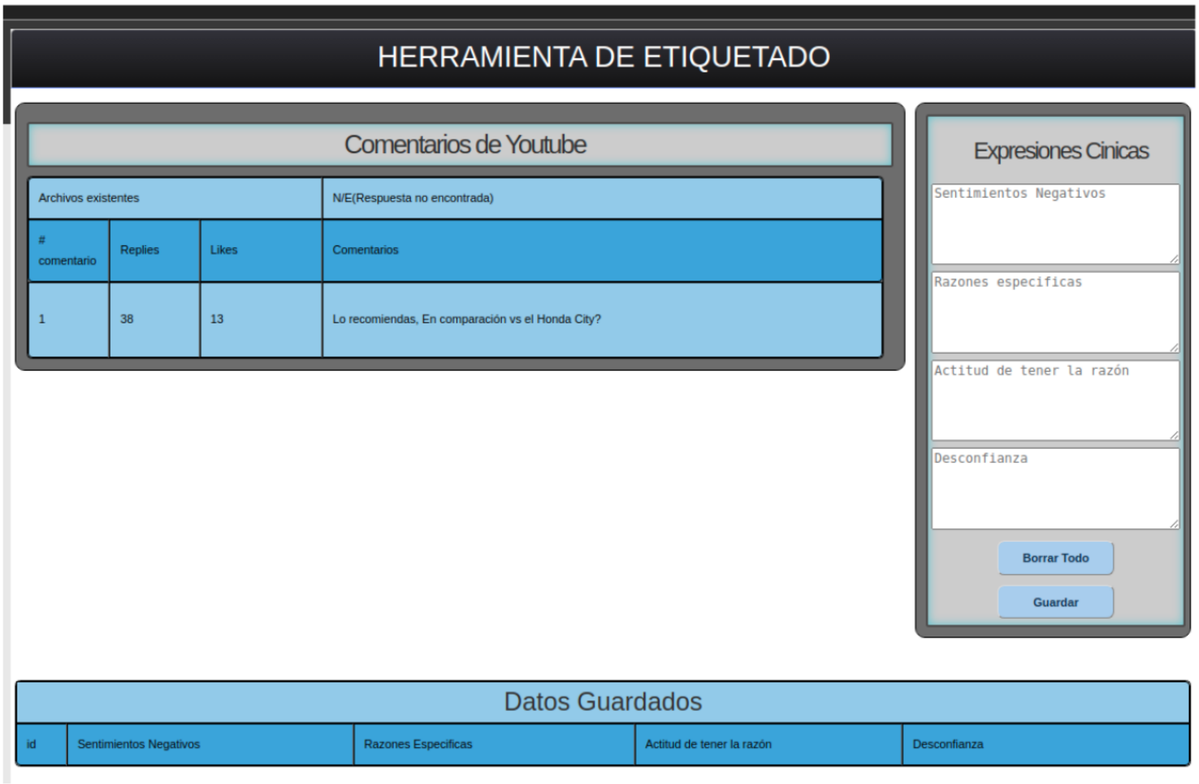
Text classification We assigned a label to each YouTube comment as positive for a class if

---

[1] @autodinamico, @autossergiooliveira, @autocosmos-mx, @gonzalo_conducir, @AlonsoMaldonado0

[2] Cynical Expression Corpus for Spanish Language

[3] `https://turet.com.mx/educationcorpus/TutorialEtiquetado.mp4`



HERRAMIENTA DE ETIQUETADO

Comentarios de Youtube

Archivos existentes			N/E(Respuesta no encontrada)
# comentario	Replies	Likes	Comentarios
1	38	13	Lo recomiendas, En comparación vs el Honda City?

Datos Guardados

id	Sentimientos Negativos	Razones Especificas	Actitud de tener la razón	Desconfianza

Figure 1: Interface annotation tool.

any part of the comment was annotated for that class, and as negative if none of the comment was annotated for that class.

For evaluation, we used the model (mBERT [4]). The training (75%), validation (12.5%), and test(12.5%) collections were constructed. The parameters with the best performance were: 10 epochs and a batch size of 16. However, the number of epochs during the fine-tuning was 10 and 20. The EarlyStopping was also included. We also applied the pysentimiento/robertuito model directly, without fine-tuning.

We considered several different pre-trained models to be fine-tuned and evaluated on our dataset:

BETO The BETO [5] model (Cañete et al., 2020) was trained following the BERT paradigm (Devlin et al., 2019), but only on Spanish documents. It is similar in size to bert-based-multilingual-cased.

SpanBERTa The SpanBERTa model[6] was trained following the RoBERTa paradigm (Liu et al., 2019), but trained on 18 GB of OSCAR's Spanish corpus. It is similar in size to BERT-Base.

(mBERT) The Multilingual-BERT (mBERT) model was trained on the concatenation of monolingual Wikipedia corpora from 104 languages. Despite being trained on separate monolingual corpora without a multilingual target, mBERT performs well on multilingual tasks (Pires et al., 2019).

We also consider a model trained specifically for hate speech detection, which is related to negative feelings and thus has potential to be usable without fine-tuning on our cynicism corpus.

RoBERTuito The RoBERTuito model[7] is based on the RoBERTa model architecture and the BETO tokenizer (Pérez et al., 2022). It was trained on 622M tweets from 432k users for hate speech detection, sentiment and emotion analysis, and irony detection.

---

[4] https://github.com/google-research/bert/

[5] https://github.com/dccuchile/beto

[6] https://github.com/chriskhanhtran/spanish-bert

		B			I			O		
Cynicism	Model	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
NF	SpanBERTa	0.689	0.715	0.705	0.656	0.657	0.660	0.741	0.740	0.737
NF	BETO	0.670	0.688	0.674	0.674	0.644	0.665	0.750	0.766	0.745
NF	mBERT	0.666	0.683	0.673	0.668	0.636	0.646	0.736	0.765	0.747
SR	SpanBERTa	0.505	0.590	0.544	0.706	0.806	0.745	0.576	0.468	0.488
SR	BETO	0.507	0.642	0.565	0.742	0.841	0.778	0.612	0.470	0.500
SR	mBERT	0.510	0.575	0.538	0.711	0.816	0.749	0.610	0.480	0.502
AR	SpanBERTa	0.593	0.720	0.666	0.745	0.868	0.800	0.620	0.421	0.497
AR	BETO	0.593	0.720	0.666	0.745	0.868	0.800	0.620	0.422	0.497
AR	mBERT	0.602	0.717	0.682	0.770	0.862	0.775	0.637	0.477	0.547

Table 3: Detailed results on treating cynicism detection as a token classification task, for negative feelings (NF), specific reasons (SR), and attitude of being right (AR).

Cynicism	Model	Precision	Recall	F1
Token classification task				
NF	SpanBERTa	0.697	0.703	0.696
NF	BETO	0.694	0.700	0.693
NF	mBERT	0.691	0.695	0.690
SR	SpanBERTa	0.598	0.622	0.592
SR	BETO	0.621	0.650	0.614
SR	mBERT	0.610	0.625	0.597
AR	SpanBERTa	0.625	0.668	0.648
AR	BETO	0.653	0.668	0.649
AR	mBERT	0.668	0.685	0.670
Text classification task				
NF	mBERT (fine-tuned)	0.902	0.948	0.925
NF	RoBERTuito (not fine-tuned)	0.620	0.731	0.671
SR	mBERT (fine-tuned)	0.912	0.981	0.945
SR	RoBERTuito (not fine-tuned)	0.500	0.128	0.204
AR	mBERT (fine-tuned)	0.728	0.981	0.849
AR	RoBERTuito (not fine-tuned)	0.461	0.089	0.150

Table 4: Overall results for detecting cynicism, either as a token classification task or a text classification task, for negative feelings (NF), specific reasons (SR), and attitude of being right (AR).

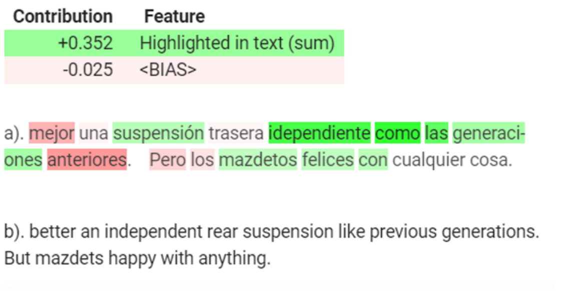
5 Results

Table 3 shows detailed results of the token classification task. The first token (B) of specific reasons were the most difficult for models to detect, with models achieving around 0.55 F1, while the inner tokens (I) of attitude of being right were the easiest, with models achieving around 0.75 F1. The different transformer models performed roughly similarly, with all F1s between comparable models within 0.04 F1 of each other.

Table 4 shows overall results for both the token classification task (using a macro-average over the B/I/O labels) and the text classification task. As with the detailed token classification results, we see that there are only small differences between the different pre-trained models when fine-tuned

---

[7] https://github.com/pysentimiento/robertuito



Contribution	Feature
+0.352	Highlighted in text (sum)
-0.025	<BIAS>

a). mejor una suspensión trasera idependiente como las generaciones anteriores. Pero los mazdetos felices con cualquier cosa.

b). better an independent rear suspension like previous generations. But mazdets happy with anything.

Figure 2: Specific reason example. a)Original text in Spanish, b) English translation. The green words contribute to the model prediction.

for token classification, with SpanBERTa being slightly higher on negative feelings, BETO being slightly higher on specific reasons, and mBERT being slightly higher on attitude of being right. The hardest cynicism type to detect in a token classification task is specific reasons, while the easiest is negative feelings.

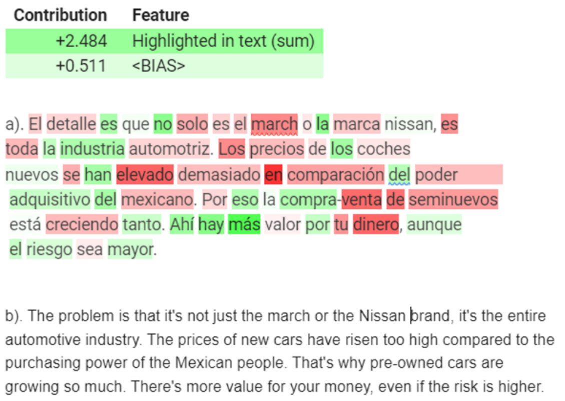
Because of the minimal differences between the models for the token classification task, we ran only the mBERT model for text classification task. We can see from table 4 that the text classification cynicism detection task is easier than the token classification cynicism detection task, with mBERT achieving > 0.8 F1 for all cynicism types. Applying the RoBERTuito without fine-tuning to this text classification task as expected results in lower performance than our fine-tuned models However, the fact that RoBERTuito is able to achieve 0.671 F1 on negative feeling detection without any fine-tuning on our corpus indicates that there is significant overlap between hate speech detection and negative feeling detection.

6 Explaining Cynicism Classifications

To give some insights into the behavior of our trained models, we apply LIME (Ribeiro et al., 2016) to the mBERT text classification models. In the following figures, green words contribute positively to the model prediction, and red contribute negatively to the model prediction.

Figure 2 shows an example of specific reason classification. Words like "suspension" and "independent" that relate to a car specification contribute positively, as does 'mazdetos", a Spanish term for owners of Mazda cars, while words like "previous" and "better" contribute negatively.

Figure 3 shows an example of attitude of being right classification. The phrase "There's (0.237) more (0.484) value(0.017) for(0.123)" that indicates value assessment contributes positively, while words like "prices(-0.300)" and "money(-0.410)" that are characteristic of the cars have a negative impacts on the model.



Contribution	Feature
+2.484	Highlighted in text (sum)
+0.511	<BIAS>

a). El detalle es que no solo es el march o la marca nissan, es toda la industria automotriz. Los precios de los coches nuevos se han elevado demasiado en comparación del poder adquisitivo del mexicano. Por eso la compra-venta de seminuevos está creciendo tanto. Ahí hay más valor por tu dinero, aunque el riesgo sea mayor.

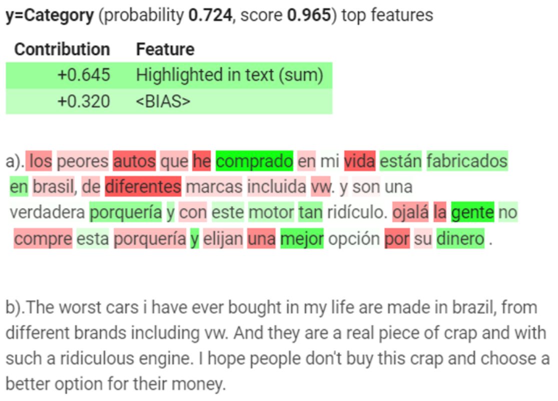
b). The problem is that it's not just the march or the Nissan brand, it's the entire automotive industry. The prices of new cars have risen too high compared to the purchasing power of the Mexican people. That's why pre-owned cars are growing so much. There's more value for your money, even if the risk is higher.

Figure 3: Attitude of being right example. a)Original text in Spanish, b) English translation. The green words contribute to the model prediction.

Figure 4 shows an example of negative feeling classification. Words that are strongly related to negative sentiment, such as "crap(0.165)", contribute positively to the model, but terms like "people(0.373)" and "money(0.204)" also contribute positively. Place of origin of car manufacture, "Brazil", and the word "brands" also negatively impact the model.

7 Discussions

The results achieved in the experiment show that it is possible to detect the three cynical expressions with reasonable reliability. Some of the results are discussed below.



y=Category (probability **0.724**, score **0.965**) top features

Contribution	Feature
+0.645	Highlighted in text (sum)
+0.320	<BIAS>

a). los peores autos que he comprado en mi vida están fabricados en brasil, de diferentes marcas incluida vw. y son una verdadera porquería y con este motor tan ridículo. ojalá la gente no compre esta porquería y elijan una mejor opción por su dinero .

b).The worst cars i have ever bought in my life are made in brazil, from different brands including vw. And they are a real piece of crap and with such a ridiculous engine. I hope people don't buy this crap and choose a better option for their money.

Figure 4: Negative Feeling example. a)Original text in Spanish, b) English translation. The green words contribute to the model prediction.

7.1 Token vs. text classification

Performance was higher on the easier text classification task and lower on the more challenging token classification task. However, token classification is closer to the objective of this work, detecting exactly which part of the comment represents the cynical expression. To extend the success of the text classification setting to the token classification setting, it may be useful to investigate two-stage approaches, where text classification is first used to identify the broad region of the cynical expressions and token classification is then used to narrow down to the specific phrases.

7.2 Expression keywords and boundaries

For negative feelings, the starts of the expressions (B) were easiest to identify, likely because they often start with terms used to describe dissatisfaction. For specific reasons and attitudes of being right, the middles of the expressions (I) were easiest to identify, likely because these types of cynicism include phrase-internal car-specific terms that might be easier to identify. Future work could investigate whether jointly learning such models might help to better establish the boundaries of the different types of cynical expressions.

7.3 Architecture comparison

We evaluated BERT-based architectures, of which three have been trained with Spanish corpora (SpanBERTa, BETO, and RoBERTuito) and one was trained on multiple languages (mBERT). Our expectations from some research (Cañete et al., 2020), (González-López et al., 2021) were that the language specific models would outperform the multilingual model, however, the gap between them was small. We thus conclude that the exact pretrained model selected is not a critical hyperparameter when fine-tuning models for Spanish cynical expression detection.

7.4 Cynicism vs. hate speech

The experiments with RoBERTuito highlight that simply using a model trained for hate speech detection will not provide a solution for cynical expression detection, even in the related category of negative feelings: a non-fine-tuned RoBERTuito achieves only 0.671 F1, while a fine-tuned mBERT achieves 0.925 F1. Nonetheless, these results indicate that there is some overlap between the two tasks, and cynical expression detection might benefit from hate speech detection models, for example, by using the predictions of the hate speech model as features in the cynical expression detection model.

Conclusions

The analysis of cynicism is important as the feelings and opinions of vocal customers can drive the decisions of other customers. We investigated cynicism in consumer opinions in comments on the YouTube platform. We annotated a corpus for three types of cynical expressions: negative feelings, specific reasons, and attitude of being right. We trained models on this corpus for both text classification and token classification settings. The results indicate that it is possible to train models to accurately detect cynical expressions in this domain.

We see our work as a building block towards technologies that detect and display the percentage of cynicism in YouTube videos. Such analyses could assist companies seeking to position their products based on what potential consumers think of their products. In future work, we aim to expand the corpus in size, in variety of components covered, and in types of cynical expressions included (e.g., sarcasm or irony).

References

Halah AlMazrua, Najla AlHazzani, Amaal AlDawod, Lama AlAwlaqi, Noura AlReshoudi, Hend AlKhalifa, and Luluh AlDhubayi. 2022. Sa‘7r: A saudi dialect irony dataset. In *Proceedinsg of the 5th Workshop on Open-Source Arabic Corpora and Processing Tools with Shared Tasks on Qur'an QA and Fine-Grained Hate Speech Detection*, pages 60–70, Marseille, France. European Language Resources Association.

Khalid Alnajjar and Mika Hämäläinen. 2021. ¡Qué maravilla! multimodal sarcasm detection in Spanish: a dataset and a baseline. In *Proceedings of the Third Workshop on Multimodal Artificial Intelligence*, pages 63–68, Mexico City, Mexico. Association for Computational Linguistics.

José Cañete, Gabriel Chaperon, Rodrigo Fuentes, Jou-Hui Ho, Hojin Kang, and Jorge Pérez. 2020. Spanish pre-trained bert model and evaluation data. In *PML4DC at ICLR 2020*.

M. Chylinski and A. Chu. 2010. Consumer cynicism: antecedents and consequences. *European Journal of Marketing*, 44(6):796–837.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of* *the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

F. C. Encinas and J. Cavazos. 2021. *Comportamientos disfuncionales El lado oscuro de los consumidores de servicios*, volume 1. Mc Graw-Hill Interamericana Editores, Ciudad de México.

Bilal Ghanem, Jihen Karoui, Farah Benamara, Paolo Rosso, and Véronique Moriceau. 2020. Irony detection in a multilingual context. In *Advances in Information Retrieval*, pages 141–149, Cham. Springer International Publishing.

Samuel González-López, Steven Bethard, Francisca Cecilia Encinas Orozco, and Adriıan Pastor López-Monroy. 2021. Consumer cynicism identification for spanish reviews using a spanish transformer model. *Procesamiento del Lenguaje Natural*, 66(0):111–120.

Ritesh Kumar, Atul Kr. Ojha, Shervin Malmasi, and Marcos Zampieri. 2020. Evaluating aggression identification in social media. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 1–5, Marseille, France. European Language Resources Association (ELRA).

Manuel Lepe-Faúndez, Alejandra Segura-Navarrete, Christian Vidal-Castro, Claudia Martínez-Araneda, and Clemente Rubio-Manzano. 2021. Detecting aggressiveness in tweets: A hybrid model for detecting cyberbullying in the spanish language. *Applied Sciences*, 11(22).

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Aaron Maladry, Els Lefever, Cynthia Van Hee, and Veronique Hoste. 2022. Irony detection for Dutch: a venture into the implicit. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 172–181, Dublin, Ireland. Association for Computational Linguistics.

Lotem Peled and Roi Reichart. 2017. Sarcasm SIGN: Interpreting sarcasm with sentiment based monolingual machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1690–1700, Vancouver, Canada. Association for Computational Linguistics.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

R.A. Potamias, G. Siolas, and A. Stafylopatis. 2020. A transformer-based approach to irony and sarcasm detection. *Neural Computing and Applications*, pages 1433 – 3058.

Juan Manuel Pérez, Damián A. Furman, Laura Alonso Alemany, and Franco Luque. 2022. Robertuito: a pre-trained language model for social media text in spanish.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should i trust you?": Explaining the predictions of any classifier.

Julian Risch, Anke Stoll, Marc Ziegele, and Ralf Krestel. 2019. hpidedis at germeval 2019: Offensive language identification using a german bert model. In *Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019)*, pages 405–410, Erlangen, Germany. German Society for Computational Linguistics & Language Technology.

Multilingual Language Models are not Multicultural: A Case Study in Emotion

Shreya Havaldar, Sunny Rai, Bhumika Singhal, Langchen Liu
Sharath Chandra Guntuku, & Lyle Ungar
University of Pennsylvania
`{shreyah,sunnyrai,bhsingha,langchen,sharathg,ungar}@upenn.edu`

Abstract

Emotions are experienced and expressed differently across the world. In order to use Large Language Models (LMs) for multilingual tasks that require emotional sensitivity, LMs must reflect this cultural variation in emotion. In this study, we investigate whether the widely-used multilingual LMs in 2023 reflect differences in emotional expressions across cultures and languages. We find that embeddings obtained from LMs (e.g., XLM-RoBERTa) are Anglocentric, and generative LMs (e.g., ChatGPT) reflect Western norms, even when responding to prompts in other languages. Our results show that multilingual LMs do not successfully learn the culturally appropriate nuances of emotion and we highlight possible research directions towards correcting this.

1 Introduction

The global reach of Large Language Models (LMs) today prompts an important question – *Are multilingual LMs also multicultural?* We are specifically interested in the multicultural behavior of LMs from the lens of emotion. LMs are used for many multilingual tasks that require emotional sensitivity and therefore must be able to reflect cultural variation in emotion. For instance, LM-powered Therapy Bots must delicately adapt the way they speak to patients in different languages (Wang et al., 2021), LMs as creative writing assistants must produce content that will elicit the appropriate emotional response in an author's desired audience (Shakeri et al., 2021), LMs used for workplace communication must understand the subtleties of interpersonal interaction (Thiergart et al., 2021), etc.

We define cultural variation in emotion as *the nuances in meaning and usage of emotion words across cultures.* For example, in English, we have many different words that express Anger. One can say "I feel angry," but may also choose to say "frustrated", "irritated", or "furious." The Anger invoked by a baby crying on an airplane is different from the Anger invoked by an unfair grade on an exam; different situations that cause Anger will invoke different language to best express it. These nuances in meaning and usage patterns of emotion words exist differently across cultures (Mesquita et al., 1997; Wierzbicka, 1999).

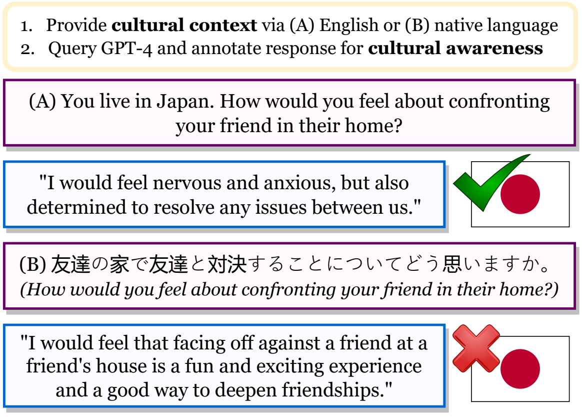

1. Provide **cultural context** via (A) English or (B) native language
2. Query GPT-4 and annotate response for **cultural awareness**
(A) You live in Japan. How would you feel about confronting your friend in their home?
"I would feel nervous and anxious, but also determined to resolve any issues between us."
(B) 友達の家で友達と対決することについてどう思いますか。
(How would you feel about confronting your friend in their home?)
"I would feel that facing off against a friend at a friend's house is a fun and exciting experience and a good way to deepen friendships."

Figure 1: Do LMs always generate culturally-aware emotional language? We prompt GPT-4 to answer "How would you feel about confronting your friend in their home?" like someone from Japan. We provide cultural context either via English (stating "You live in Japan" in the prompt) or via a Japanese prompt. GPT-4 returns two drastically different completions, with the Japanese completion annotated as not culturally appropriate.

Therefore, there is not a perfect one-to-one mapping between languages for emotion words coupled with their meaning and usage patterns. The direct translation for "I feel frustrated" from English to Chinese (simplified), for example, is "我感到沮丧". However, in a situation where a native English speaker would likely say "I feel frustrated," a native Chinese speaker may use a different phrase than "我感到沮丧", based on situation, context, and the cultural norms of emotion expression in China.

As we rely on multilingual LMs today for emotionally sensitive tasks, they must reflect this cultural variation in emotion. However, the widely-

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 202–214
July 14, 2023 ©2023 Association for Computational Linguistics

used multilingual LMs are trained on Anglocentric corpora and encourage alignment of other languages with English (Reimers and Gurevych, 2020), both implicitly and explicitly, during training. The key problem in this approach to building multilingual LMs is that any form of alignment destroys a model's ability to encode subtle differences, like the difference between "I feel frustrated" in the United States and "我感到沮丧" in China.

In this paper, we investigate whether widely-used multilingual LMs reflect cultural variation in emotion. We select four high-resource written languages, two Western and two Eastern, to focus on in this work – English, Spanish, Chinese (Simplified), and Japanese.

Specifically, we investigate two facets of LMs: embeddings and language generation.

1. Emotion embeddings
    (a) **Does implicit and explicit alignment in LMs inappropriately anchor emotion embeddings to English?** We compare embeddings from monolingual, multilingual, and aligned RoBERTa models.
    (b) **Do emotion embeddings reflect known psychological cultural differences?** We project embeddings onto the Valence-Arousal plane to visualize American vs. Japanese differences in Pride and Shame.
2. Emotional language generation
    (a) **Do LMs reflect known psychological cultural differences?** We analyze whether GPT-3 probabilities encode American vs. Japanese differences in Pride and Shame.
    (b) **Do LMs provide culturally-aware emotional responses?** We prompt GPT-3.5 and GPT-4 with scenarios that should elicit varied emotional responses across cultures and conduct a user study to assess response quality.

We make our code public [1] and encourage researchers to utilize the analyses outlined in this work as a baseline to measure the cultural awareness of future multilingual models.

2 Related Work

A large body of work in NLP focuses on detecting emotion in multilingual text . However, a major oversight in this line of research is that *it treats emotion as culturally invariant*. Work from Bianchi et al. (2022) gathers a corpus of annotated social media data from 19 languages, but uses machine translation to transfer annotations from one language to another, assuming that translation correctly captures emotional variation. Work from Buechel et al. (2020) generates lexica to analyze emotion across 91 languages, relying on translations from English lexica and assuming that the affective state of parallel words will be identical.

Psychologists have characterized emotion as having multiple components – an emotional experience, a physiological response, and a behavioral response tendency (Kensinger and Schacter, 2006). Each of these components vary from culture to culture (Mesquita et al., 1997), a complexity completely ignored when emotion is treated as a *static, transferable label* on an utterance of text. Using machine translation to transfer emotion labels between languages incorrectly assumes that emotion is experienced identically across cultures.

Others have also observed that LMs can fail to account for cultural context and variation. Cao et al. (2023) find that ChatGPT strongly aligns with American values. Magno and Almeida (2021) use word embeddings to globally measure human values across cultures, and find that these values overlap more when measured via data in English vs. native languages. Arora et al. (2023) probe multilingual LMs and discover weak alignment with the cultural values reflected by these LMs and established values surveys.

In this paper, we focus on emotion, showing a wider variety of Anglocentric anchoring by elucidating the underlying mechanisms of this alignment. We investigate emotion embeddings and LM probabilities, as well as affective language generated from multilingual LMs.

3 Investigating Emotion Embeddings

Many tasks in multilingual NLP utilize embeddings from pre-trained LMs such as XLM-RoBERTa (Conneau et al., 2019) and mBERT (Devlin et al., 2018). Researchers fine-tune these models for downstream tasks, relying on their learned representations of words and concepts.

We scope our investigation to embeddings from the widely used XLM-RoBERTa models. XLM-RoBERTa was trained on text that includes parallel and comparable corpora (e.g., Wikipedia) in mul-

---

[1] `https://github.com/shreyahavaldar/Multicultural_Emotion/`

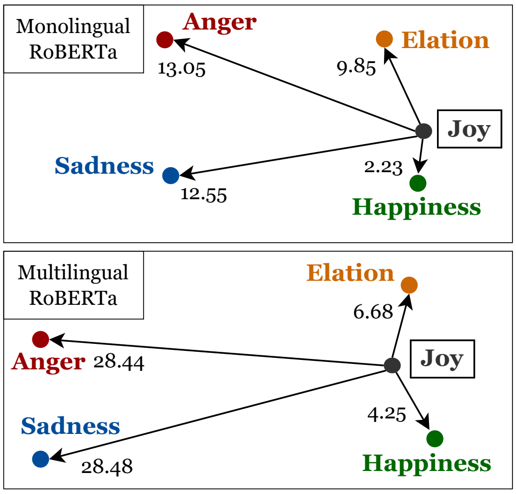


Figure 2: We determine the similarity between the embeddings of monolingual Joy and multilingual Joy by comparing the distances from Joy to other emotions embeddings in both settings. Specifically, we calculate the correlation between $< 13.05, 9.85, 12.55.2.23 >$ and $< 28.44, 6.68, 28.48, 4.25 >$ to infer similarity.

tiple languages. The nature of Wikipedia, which has topic-aligned articles in different languages, causes *implicit alignment* in training. Worse, XLM-RoBERTa variants trained via multilingual knowledge distillation (Reimers and Gurevych, 2020) enforce English sentences and their translations to map to the same point in embedding space, giving *explicit alignment* of other languages with English.

This section investigates the effect of alignment – both implicit and explicit – by analyzing emotion embeddings from monolingual, multilingual, and aligned RoBERTa models (See Table A2). We further investigate whether this anchoring impacts our ability to visualize known cultural differences (e.g. differences between Pride and Shame in the US vs. Japan (Tsai et al., 2006)) when projecting embeddings into the two-dimensional Valence-Arousal plane (Russell, 1980).

3.1 Does implicit and explicit alignment inappropriately anchor emotion embeddings to English?

We analyze whether implicitly aligned embeddings become Anglocentric by comparing emotion embeddings from XLM-RoBERTa to emotion embeddings learned in a parallel, monolingual setting. We further analyze explicit alignment by comparing embeddings from vanilla XLM-RoBERTa to an explicitly aligned variant of XLM-RoBERTa (Reimers and Gurevych, 2020).

Distance-Based Similarity How do we compare the emotion embeddings of two models? Let us take Joy, one of the six Ekman emotions (Ekman et al., 1999), as an example – can we compare the similarity of embeddings from two models for the phrase "I feel joy"? [2] A direct numerical comparison is challenging, as we would need to align the embedding spaces of these two models and possibly distort the Joy embeddings. Taking this into account, we pose the following solution:

The more similar two models are, the more similarly we expect them to embed the same phrases in embedding space. For example, let us embed phrases x, y, and, z using Model A and Model B. This gives us the embedding vectors $\vec{x}_A, \vec{y}_A, \vec{z}_A$ and $\vec{x}_B, \vec{y}_B, \vec{z}_B$ respectively. Figure 2 illustrates this, showing the embeddings of Joy, Anger, Elation, Sadness, and Happiness using a monolingual and multilingual RoBERTa model.

If Model A and Model B have embedded phrases x, y, and z in a similar way, then we expect to see a high correlation between the numerical distances $x \to y, x \to z$, and $y \to z$ in the respective embedding spaces of Model A and B. We calculate the correlation between the following two vectors:

$$< \|\vec{x}_A - \vec{y}_A\|, \|\vec{x}_A - \vec{z}_A\|, \|\vec{y}_A - \vec{z}_A\| >$$
$$< \|\vec{x}_B - \vec{y}_B\|, \|\vec{x}_B - \vec{z}_B\|, \|\vec{y}_B - \vec{z}_B\| >$$

to inform how similar the embeddings of x, y, and, z are between Model A and Model B.

Using this idea, we can compare the *distances* from "I feel joy" to other contextualized emotion phrases (e.g. "I feel anger", "I feel happiness", etc.) in embedding space A to those same distances in embedding space B. For example, if the monolingual and multilingual RoBERTa models shown in Figure 2 have learned similar representations of Joy, then we can expect to see a high Pearson correlation between the vectors $< 13.05, 9.85, 12.55.2.23 >$ and $< 28.44, 6.68, 28.48, 4.25 >$. We use this distance-based similarity metric to answer the following three questions:

1. Do implicitly aligned multilingual LMs embed emotion words differently than monolingual LMs?
2. Do implicitly aligned multilingual LMs embed emotion words in an Anglocentric way?
3. Does explicit alignment further anchor multilingual emotion embeddings to English?

---

[2] We prepend each emotion word with the phrases "I feel" and "I am" to add context and circumvent polysemy when generating embeddings for analysis.

Do implicitly aligned multilingual LMs embed emotion words differently than monolingual LMs? We compare the emotion representations from *monolingual* and *multilingual* RoBERTa models across English, Spanish, Chinese, and Japanese. We select the four monolingual RoBERTa models most downloaded on Huggingface, additionally ensuring the four models selected have the same number of parameters. Table A2 contains additional details on the models used in our experiments.[3]

Figure 2 illustrates this experiment. In practice, we use a list of 271 emotions (Davis, 2023) for our distance-based similarity computation. Additionally, to account for variance in descriptions of experiencing emotion, we average the embedding of two contextualized phrases for each emotion – "I feel <*emotion*>" and "I am <*emotion*>".

For non-English languages, we machine translate the two contextualized English phrases for each emotion (e.g. a representation of Joy in English is the average of the embeddings of "I feel joy" and "I am joyful". The representation of Joy in Spanish is the average of the embeddings "siento alegría" and "soy alegre", etc.). In order to ensure quality, we have native speakers evaluate a subset of the machine-translated emotion phrases, and we find that translation does yield sufficient results.

We then apply our distance-based similarity metric to compare the monolingual and multilingual emotion embeddings across languages. The "Mono vs. Multi" column in Table 1 shows the average distance-based similarity across all 271 emotions. The lower similarities for non-English languages indicate that *XLM-RoBERTa embeds non-English emotions differently compared to monolingual models*. We can thus say that multilingual LMs do not preserve the embedding space of monolingual non-English LMs.

Do implicitly aligned multilingual LMs embed emotion words in an Anglocentric way? We compare the emotion representations of *English* vs. *non-English* languages. We apply our distance-based similarity metric to measure the similarity between English and non-English emotion representations in two settings – monolingual and multilingual. Figure 3 illustrates this experiment.

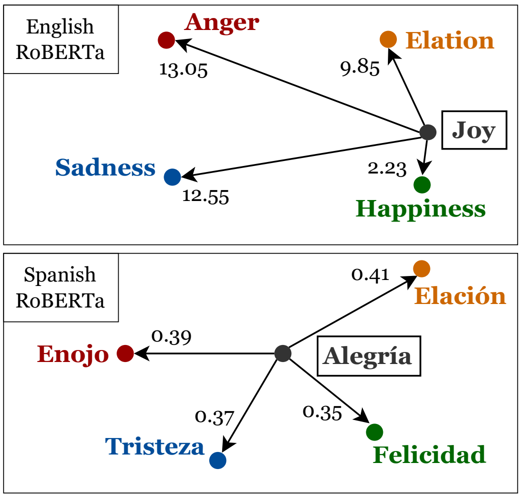


Figure 3: We compare the similarity between the embeddings of Joy in English and Joy(Alegría) in Spanish by comparing the distances from Joy to other emotion embeddings in both languages. Specifically, we calculate the correlation between $< 13.05, 9.85, 12.55.2.23 >$ and $< 0.39, 0.41, 0.37, 0.35 >$ to infer similarity.

The "English vs. Non-English" columns in Table 1 show the average distance-based similarity between English and non-English emotion embeddings across all 271 emotions, in monolingual and multilingual settings respectively. Results reveal low similarity between non-English and English emotion embeddings in monolingual space. *In a multilingual setting, however, the non-English emotion embeddings become more similar to English ones.* This suggests that implicit alignment in multilingual LMs anchors non-English emotion embeddings to their English counterparts.

Does explicit alignment further anchor multilingual emotion embeddings to English? We compare emotion embeddings from an *unaligned* RoBERTa model to a RoBERTa model trained via *forced alignment* across English, Spanish, Chinese, and Japanese (Reimers and Gurevych, 2020).

The average distance-based similarity between aligned and unaligned emotion embeddings across all 271 emotions is shown in column "Aligned vs. Unaligned" in Table 1. *Emotion embeddings from explicitly aligned models are most similar to unaligned embeddings in English*, indicating explicitly aligned embedding space fails to preserve the structure of non-English embedding spaces.

Finding 1: Multilingual LMs embed non-English emotion words differently from their monolingual counterparts, whereas English emotion embed-

---

[3] We note that differences in training data for the monolingual RoBERTa models affect how these models are able to capture emotion. However, it is important to investigate LMs actively used in NLP research rather than explicitly creating a perfectly parallel set of monolingual models.

	Mono vs. Multi	English vs. Non-English		Aligned vs. Unaligned
Language (L)	$\bar{r}(L_{mono}, L_{multi})$	$\bar{r}(En, L)_{mono}$	$\bar{r}(En, L)_{multi}$	$\bar{r}(L_{algn}, L_{unalgn})_{multi}$
English (En)	**0.758** (0.35)	—	—	**0.483** (0.22)
Spanish	0.318* (0.20)	0.222* (0.14)	**0.628*** (0.36)	0.280* (0.19)
Chinese	0.378* (0.10)	0.213* (0.12)	**0.437*** (0.35)	0.102* (0.06)
Japanese	0.332* (0.18)	0.055* (0.09)	**0.485*** (0.39)	0.332* (0.18)

Table 1: We report the average distance-based similarity across 271 emotions for each of our experiments (standard deviation given in parentheses). *indicates the difference in mean correlation between English vs. non-English settings (for Mono vs. Multi, Aligned vs. Unaligned) and monolingual vs. multilingual settings (for English vs. Non-English) is statistically significant ($p < 0.05$); we compute this using an independent t-test. See Table A2 for models used in each setting.

dings are more stable and similar in all settings. We demonstrate that *implicit and explicit alignment in multilingual LMs anchor non-English emotion embeddings to English emotions.* All observed trends persist under ablation studies on the effect of distance metric and correlation function (see Appendix A).

3.2 Do emotion embeddings reflect known psychological cultural differences?

Though emotion embeddings from multilingual LMs are Anglocentric, we nonetheless investigate whether they encode any information about known cultural variation in emotion. Prior work (Tsai, 2017; Russell et al., 1989) underlines the differences in emotional expression across cultures, and often illustrates these differences via the circumplex model of affect (Russell, 1980). The circumplex model assumes all emotions can be classified along two independent dimensions – *arousal* (the magnitude of intensity or activation) and *valence* (how negative or positive).

Pride and Shame are two widely researched emotions when investigating cultural differences in emotional expression. (Lewis et al., 2010; Wong and Tsai, 2007). Shame is expressed more commonly and has a desirable affect in Eastern cultures compared to Western cultures. Similarly, Pride is openly expressed in Western cultures whereas Eastern cultures tend to inhibit the feeling of Pride (Lim, 2016). Moreover, these proclivities are deeply ingrained in society and thus acquired at a very young age (Furukawa et al., 2012).

For our experiments, we consider the US and Japan, as the subtle differences in expression of Pride and Shame between these two cultures are well-studied (Kitayama et al., 2000; Tsai et al., 2006). We project emotion embeddings from English and Japanese onto the Valence-Arousal plane to visualize whether multilingual LMs capture the expected differences in Pride and Shame. When comparing the embeddings, we expect to specifically observe:

1. The embedding for English Pride should have a more positive valence. *(as Pride is more accepted in the US than Japan)* (Furukawa et al., 2012)
2. The embedding for English Shame should have a more negative valence. *(as Shame is more embraced in Japan than the US)* (Furukawa et al., 2012)
3. The embeddings for English Pride should have higher arousal *(as Pride is more internally and culturally regulated in Japan than the US)* (Lim, 2016)

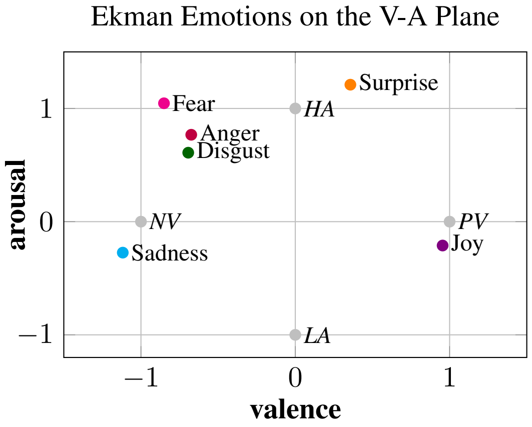


Figure 4: The six Ekman emotions projected onto the Valence-Arousal plane. We replicate the circumplex model of affect, enabling visualization and theoretical analysis of multi-dimensional emotion embeddings.

Projection into the Valence-Arousal plane In order to define the valence and arousal axes, we first generate four axis-defining points by averaging the contextualized embeddings of the emotions listed in Table A1. This gives us four vectors in embedding space that best represent positive valence (PV) negative valence (NV), high arousal (HA), and low arousal (LA). We can now project any emotion embedding onto the plane defined by the valence axis ($NV \rightarrow PV$) and the arousal axis ($LA \rightarrow HA$). We give a more formal, mathematical description of this projection method in the Appendix B. Figure 4 shows the six Ekman emotions (Ekman et al., 1999) projected into the Valence-Arousal plane, indicating that our projection method successfully recreates the circumplex.

To visualize Pride and Shame in the Valence-Arousal plane, we manually translate the axis-defining emotions to Japanese and average the English and Japanese points of each axis category to define *multilingual valence and arousal axes*. We then project the contextualized sentence embeddings "I am proud" and "I am ashamed" in English and Japanese. We experiment with both aligned and unaligned RoBERTa models; these plots are shown in Figure 5.

Looking at the plots, we observe that English Pride is slightly higher in valence than Japanese Pride, and English Shame is slightly lower in valence than Japanese Shame. This does serve as a weak confirmation of the first two hypotheses. However, we do not observe English Pride to have higher arousal than Japanese Pride. This discrepancy suggests our results are inconclusive, and we cannot confirm whether multilingual RoBERTa encodes cultural variation in English vs. Japanese Pride and Shame.

Finding 2: By projecting emotion embeddings into the Valence-Arousal plane, we show that *LMs are not guaranteed to encode the nuances in meaning and usage of emotion words across cultures.* Researchers who utilize embeddings from multilingual LMs for emotion-related tasks assume these pre-trained models have learned adequate representations of emotion across languages. However, implicit and explicit alignment during training causes multilingual LMs to ignore the subtle differences in emotion expression across cultures.

4 Investigating multilingual LM generation

We now turn from investigating embeddings to analyzing language generated by Language Models (GPT-3, GPT-3.5, and GPT-4) to see if multilingual LM completions reflect cultural variation in emotion. In order for LMs to be used for tasks that require emotional sensitivity, their responses must align with cultures' socio-cultural norms (Genesee, 1982); generated text must reflect users' cultural tendencies and expected affect (Tsai, 2017).

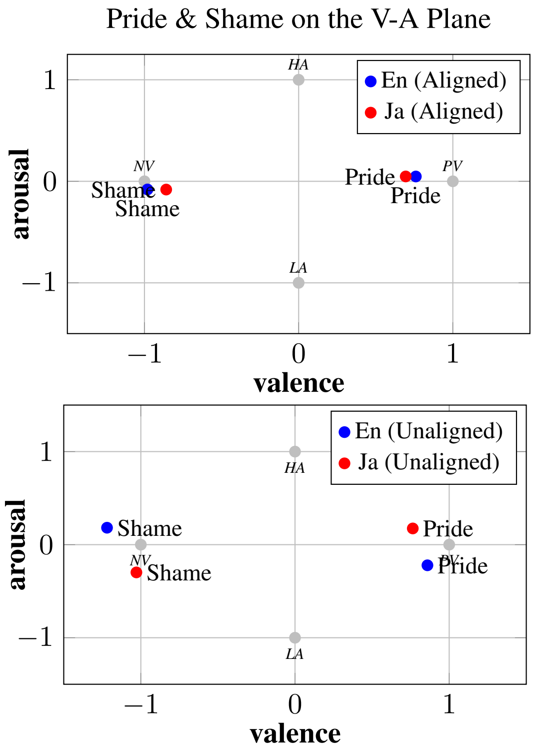


Figure 5: We project English and Japanese Pride and Shame embeddings into the Valence-Arousal plane. We use an aligned (top) and unaligned (bottom) RoBERTa model to embed the contextualized emotions. In both cases, we do not see all of our hypotheses confirmed.

We first analyze token-level completion probabilities from GPT-3, to see if they reflect cultural differences between American and Japanese Shame and Pride. We then prompt GPT-3.5 and GPT-4 in English and non-English languages to respond to scenarios that should elicit different emotional responses across cultures and assess their cultural appropriateness in a small-scale user study.

4.1 Do LMs reflect known psychological cultural differences?

Continuing our example of English vs. Japanese Pride and Shame, we evaluate whether this known cultural difference is reflected in OpenAI's GPT-3.

We design a set of 24 prompts (See Table A5) for GPT-3 (`davinci`) based on six scenarios that would invoke a combination of Pride and Shame in the form `<context><feeling>`. For example, "I received an award in front of my coworkers. I feel proud." One might feel proud for re-

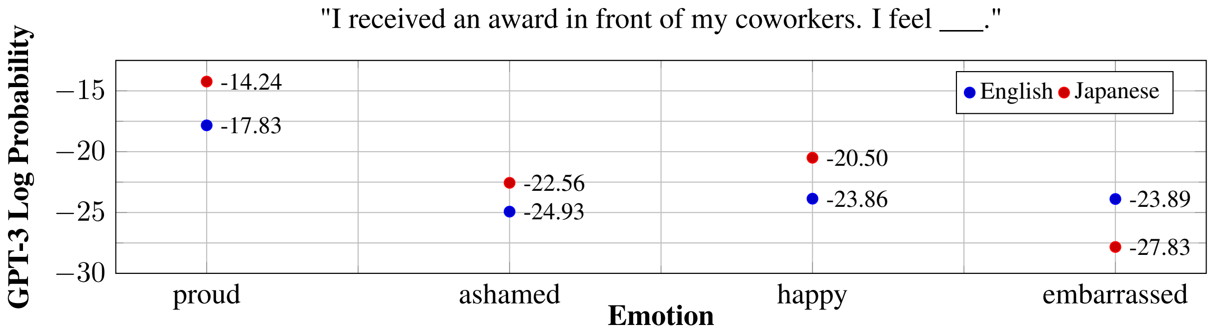


Figure 6: A comparison of GPT-3 sentence completion probabilities in English and Japanese. We show the log probabilities for the sentence "I feel X." following the scenario "I received an award in front of my coworkers." and test emotion words associated with Pride or Shame in English and Japanese. Contrary to cultural expectation, we do not observe a pattern where Pride words have a higher likelihood in English or Shame words have a higher likelihood in Japanese.

ceiving an award or embarrassed for being publically praised. We then prompt GPT-3 using various `<context><feeling>` prompts, and analyze the log probability of each token of the prompt. Finally, we sum the log probability of each token in the `<feeling>` sentence to get a sense of how likely the `<feeling>` is to follow the `<context>`. Based on cultural norms about how one would react in situations that elicit both Pride and Shame, we expect to see a higher probability for "I feel happy" and "I feel proud" in English, and a higher probability for "I feel embarrassed" and "I feel ashamed" in Japanese across scenarios.

Figure 6 shows the results of this for the prompt "I received an award in front of my coworkers. I feel ___." where we test two Pride words: "proud", "happy", and two Shame words: "ashamed", and "embarrassed". We replicate this experiment in Japanese, and compare the summed log probabilities of "I feel ___." between English and Japanese across emotions. The full results, along with the remaining prompts are given in Appendix Table A5. Analyzing the results across six scenarios (see Appendix C), we do not see any consistent evidence that Pride is more likely to be expressed in English or Shame is more likely to be expressed in Japanese. In Figure 6, for example, we see contradicting results for "proud", "happy", and "embarrassed".

Finding 3: These results suggest that *GPT-3 lacks knowledge of Pride and Shame and the norms surrounding their expression in the US and Japan.* This is a major limitation; such a failure to capture cultural variation is likely to limit both the utility and applicability of LMs in downstream emotionally-sensitive tasks.

4.2 Do LMs provide culturally-aware emotional responses?

To further investigate whether LM completions reflect cultural norms, we conduct a small-scale user study to see if GPT-3.5 and GPT-4 are capable of appropriately adapting when prompted in different languages. Annotators assess whether the completions parallel the accepted emotional responses associated with the user's culture.

Prompting with cultural context Prior psychological research has detailed scenarios that reveal how emotional expressions vary across cultures (Mesquita, 2022). We use this work to design a set of 19 questions (see Table A6) that should elicit different emotional responses across cultures. For example, the question "How would you feel if your guests chose to keep their shoes on when entering your home?" would likely elicit a different response from someone culturally American vs. Chinese.

We use these scenarios to prompt GPT-3.5 (`gpt-3.5-turbo`) and GPT-4 (`gpt-4`) in the form `<prefix>+<question>` (see Table 2). In order to include cultural context and coax the LM into returning a culturally appropriate emotional response, we experiment with providing cultural context in two ways (using Chinese as an example):

1. *via English* – we add the sentence "You live in China." to the prefix. The LM returns an English completion.
2. *via Native Language* – we translate the `<prefix>+<question>` prompt to Chinese. The LM returns a Chinese completion.

Figure 1 details these two cultural context modes and how they may cause conflicting LM responses.

Prefix
You are a helpful chatbot. Your goal is to answer my questions like you are a human capable of feelings and emotions. {**You live in China.**} Answer the following question using a single sentence that begins with "I would feel...".
Questions
Your neighbor offers you food. How would you feel turning it down? How would you feel if your parents moved in with you right after you were married? How would you feel if your guests chose to keep their shoes on when entering your home?

Table 2: Example prompts (`<prefix>+<question>`) designed to evaluate whether GPT-3.5 and GPT-4 can adapt to account for cultural variation in emotion. In our first set of experiments, we include the bold sentence "You live in China." and prompt GPT in English. In our second set of experiments, we do NOT include the bold sentence, and instead provide cultural context by translating our `<prefix>+<question>` prompt to Chinese. The full set of questions is given in Appendix Table A6.

User Study To assess the quality of the LM completions, we perform a small-scale user study using eight volunteers, consisting of four pairs fluent in English, Spanish, Chinese, and Japanese respectively. We ask our volunteers to annotate GPT-3.5 and GPT-4's responses for cultural awareness along two axes - *linguistic norms* (how you would expect a native speaker to talk), and *cultural norms* (what you would expect a native speaker to say). As these two norms are deeply correlated, annotators are instructed to take both of these dimensions into account and give a single rating to each completion. We use a scale of 1-7, where 7 indicates the LM's response is fully expected of a native speaker.

Across languages, we observe a high agreement within each pair of volunteers. Figure 7 details the average score across annotators and questions for GPT-4 and GPT-3.5 completions. We provide the annotator agreement statistics in Appendix Table A4. Analyzing the completions and annotations, we notice some interesting trends:

- We see a large difference in quality between the LM responses returned using the two cultural context prompting modes (even though the questions are identical.)
- For Chinese and Japanese, the LM returns a less culturally-appropriate response using the *Native Language* cultural context mode.
- English completions are the most culturally-aware across languages, and English response quality is unaffected by cultural context mode.

Finding 4: GPT-3.5 and GPT-4 fail to infer that a prompt in a non-English language suggests a response that aligns with the linguistic and cultural norms of a native speaker. Additionally, the LM completions reflect culturally appropriate emotion much better in Western languages than Eastern.

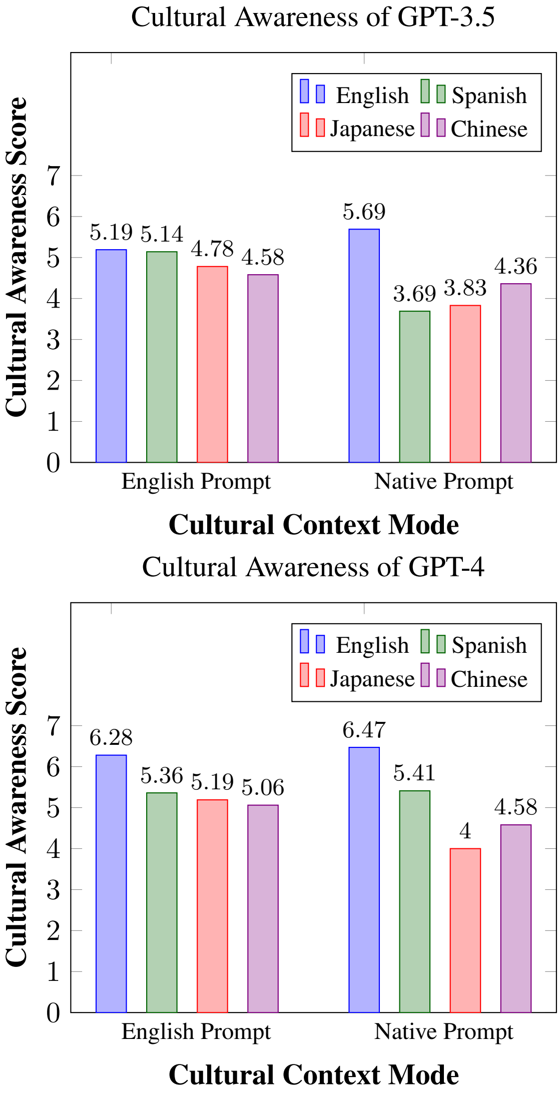


Figure 7: Average cultural awareness scores across annotations for GPT-3.5 and GPT-4 completions in each language. We observe a consistently higher quality of English completions, and poor performance of Eastern languages compared to Western, especially when prompted using the *Native Language* context mode.

5 Conclusion

We find that multilingual models fail to fully capture cultural variations associated with emotion, and predominantly reflect the cultural values of the Western world. Emotion embeddings from multilingual LMs are anchored to English, and the text completions generated in response to non-English prompts are not in tune with the emotional tendencies of users' expected culture. For instance, when GPT-4 is prompted in Japanese, it responds as an American fluent in Japanese but unaware of Japanese culture or values.

Our results caution against blindly relying on emotion representations learned by LMs for downstream applications. Using machine translation to transfer labels or utilizing multilingual LMs in a zero-shot setting for unseen languages has risks – the multilingual representations of emotion learned by these models do not perfectly reflect how their corresponding cultures express emotion.

Future Research Directions Our paper motivates the need for future work that transcends current Anglocentric LMs. This could take the form of higher performing, non-English models in a monolingual setting, or of multilingual models trained on more linguistically and culturally balanced corpora. Future work should additionally investigate whether state-of-the-art monolingual models in non-English languages succeed in encoding the respective culture's norms. Furthermore, we encourage the evaluation of multilingual models on benchmarks that measure cultural awareness in addition to standard metrics.

6 Limitations

We only analyze four high-resource languages in this study, our analysis could have benefited from more languages, especially low-resource ones. Additionally, we only analyze Japanese and English Pride/Shame as a known cultural difference; analyzing other differences could provide stronger results. We perform a small user study, and our work could have benefited from a larger-scale study with more annotators and completions analyzed.

We recognize the added complexity of investigating Pride embeddings from a culture where explicit expressions of Pride are discouraged; we note this may be a contributing factor to our results indicating that LMs do not reflect the culturally appropriate nuances of Shame and Pride. Additionally, we acknowledge that the experiments outlined in this paper are specific to investigating cultural awareness from the lens of emotion. These experiments are not easily applicable to measuring cultural awareness from different perspectives; therefore, results may not be generalizable.

At a higher level, we equate *language* with *culture*. Psychologists have observed higher cultural similarities within languages than between them (Stulz and Williamson, 2003), however, we recognize there are variations within the populations that speak each language. For example, Spanish is spoken by people in Spain, Mexico, and other countries, each having a unique and varied culture.

7 Ethical Considerations

Although culturally-aware multilingual LMs are critical in uses such as therapy, storytelling, and interpersonal communication, these are possible misuses for nefarious purposes - persuasion, misinformation generation, etc. Additionally, our analyses behave as if China, Japan, Spain, and the United States are a single culture with a single set of cultural norms. In reality, this is not the case; we recognize there are huge variations in the way people view emotion within each of these cultures.

References

Arnav Arora, Lucie-aimée Kaffee, and Isabelle Augenstein. 2023. Probing pre-trained language models for cross-cultural differences in values. In *Proceedings of the First Workshop on Cross-Cultural Considerations in NLP (C3NLP)*, pages 114–130, Dubrovnik, Croatia. Association for Computational Linguistics.

Federico Bianchi, Debora Nozza, and Dirk Hovy. 2022. XLM-EMO: Multilingual emotion prediction in social media text. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 195–203, Dublin, Ireland. Association for Computational Linguistics.

Sven Buechel, Susanna Rücker, and Udo Hahn. 2020. Learning and evaluating emotion lexicons for 91 languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1202–1217, Online. Association for Computational Linguistics.

Yong Cao, Li Zhou, Seolhwa Lee, Laura Cabello, Min Chen, and Daniel Hershcovich. 2023. Assessing cross-cultural alignment between chatgpt and human societies: An empirical study.

Tianyu Cho and Kei Sawada. 2021. Pre-learning model for japanese natural language processing. *Japanese*

Society for Artificial Intelligence Research Group Material Language/Speech Understanding and Dialogue Processing, 93:169–170.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for Chinese natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 657–668, Online. Association for Computational Linguistics.

Tchiki Davis. 2023. List of emotions: 271 emotion words.

Javier De la Rosa, Eduardo G. Ponferrada, Manu Romero, Paulo Villegas, Pablo González de Prado Salas, and María Grandury. 2022. Bertin: Efficient pre-training of a spanish language model using perplexity sampling. *Procesamiento del Lenguaje Natural*, 68(0):13–23.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Paul Ekman et al. 1999. Basic emotions. *Handbook of cognition and emotion*, 98(45-60):16.

Emi Furukawa, June Tangney, and Fumiko Higashibara. 2012. Cross-cultural continuities and discontinuities in shame, guilt, and pride: A study of children residing in japan, korea and the usa. *Self and Identity*, 11(1):90–113.

Fred Genesee. 1982. The social psychological significance of code switching in cross-cultural communication. *Journal of language and social psychology*, 1(1):1–27.

Elizabeth A Kensinger and Daniel L Schacter. 2006. Processing emotional pictures and words: Effects of valence and arousal. *Cognitive, Affective, & Behavioral Neuroscience*, 6(2):110–126.

Shinobu Kitayama, Hazel Rose Markus, and Masaru Kurokawa. 2000. Culture, emotion, and well-being: Good feelings in japan and the united states. *Cognition & Emotion*, 14(1):93–124.

Michael Lewis, Kiyoko Takai-Kawakami, Kiyobumi Kawakami, and Margaret Wolan Sullivan. 2010. Cultural differences in emotional responses to success and failure. *International journal of behavioral development*, 34(1):53–61.

Nangyeon Lim. 2016. Cultural differences in emotion: differences in emotional arousal level between the east and the west. *Integrative Medicine Research*, 5(2):105–109.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Gabriel Magno and Virgilio Almeida. 2021. Measuring international online human values with word embeddings. *ACM Trans. Web*, 16(2).

Batja Mesquita. 2022. *Between us: How cultures create emotions*. WW Norton & Company.

Batja Mesquita, Nico H Frijda, and Klaus R Scherer. 1997. Culture and emotion. *Handbook of cross-cultural psychology*, 2:255–297.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

James A Russell. 1980. A circumplex model of affect. *Journal of personality and social psychology*, 39(6):1161.

James A Russell, Maria Lewicka, and Toomas Niit. 1989. A cross-cultural study of a circumplex model of affect. *Journal of personality and social psychology*, 57(5):848.

Hanieh Shakeri, Carman Neustaedter, and Steve DiPaola. 2021. Saga: Collaborative storytelling with gpt-3. In *Companion Publication of the 2021 Conference on Computer Supported Cooperative Work and Social Computing*, CSCW '21, page 163–166, New York, NY, USA. Association for Computing Machinery.

Rene M Stulz and Rohan Williamson. 2003. Culture, openness, and finance. *Journal of financial Economics*, 70(3):313–349.

Jonas Thiergart, Stefan Huber, and Thomas Übellacker. 2021. Understanding emails and drafting responses - an approach using GPT-3. *CoRR*, abs/2102.03062.

Jeanne L Tsai. 2017. Ideal affect in daily life: Implications for affective experience, health, and social behavior. *Current Opinion in Psychology*, 17:118–128.

Jeanne L Tsai, Robert W Levenson, and Kimberly McCoy. 2006. Cultural and temperamental variation in emotional response. *Emotion*, 6(3):484.

Lu Wang, Munif Ishad Mujib, Jake Ryland Williams, George Demiris, and Jina Huh-Yoo. 2021. An evaluation of generative pre-training model-based therapy chatbot for caregivers. *CoRR*, abs/2107.13115.

Anna Wierzbicka. 1999. *Emotions across languages and cultures: Diversity and universals*. Cambridge university press.

Ying Wong and Jeanne Tsai. 2007. Cultural models of shame and guilt. *The self-conscious emotions: Theory and research*, 209:223.

A Distance-based Similarity Experiments: Additional Details

Table A2 gives details on the RoBERTa models we use in each setting – monolingual, multilingual, and aligned – for all experiments in this paper.

We find no clear pattern in certain emotions being more or less problematic across languages. Our machine translations of 271 English emotions give 247, 210, and 246 unique emotions for Spanish, Chinese, and Japanese respectively.

In order to test the robustness of the experiments outlined in section 3.1, we experiment with other distance and correlation metrics in our distance-based similarity calculations. Table A3 shows results for our distance-based similarity experiments where we replace Euclidean distance with cosine similarity, and results where we replace Pearson correlation with Spearman's rank.

B Projection into the Valence-Arousal plane

In order to define the valence and arousal axes, we first generate four axis-defining points by averaging the contextualized embeddings ("I feel [emotion]") of the emotions listed in Table A1. This gives us four vectors in embedding space – positive valence ($\vec{v}_{pos}$), negative valence($\vec{v}_{neg}$), high arousal($\vec{a}_{high}$), and low arousal($\vec{a}_{low}$). We mathematically describe our projection function below:

Axis Anchor	**Russell Emotions**
Positive valence (PV)	Happy, Pleased, Delighted, Excited, Satisfied
Negative valence (NV)	Miserable, Frustrated, Sad, Depressed, Afraid
High arousal (HA)	Astonished, Alarmed, Angry, Afraid, Excited
Low arousal (LA)	Tired, Sleepy, Calm, Satisfied, Depressed

Table A1: Emotions used to define the valence and arousal axis anchors for projection into the Valence-Arousal plane. We select the 5 emotions from the circumplex closest to each axis point.

1. We define the valence axis, V, as $\vec{v}_{pos} - \vec{v}_{neg}$ and the arousal axis, A, as $\vec{a}_{high} - \vec{a}_{low}$. We then normalize V and A and calculate the origin as the midpoints of these axes: $(\vec{v}_{middle}, \vec{a}_{middle})$.
2. We then scale the axes so $\vec{v}_{pos}$, $\vec{v}_{neg}$, $\vec{a}_{high}$, and $\vec{a}_{low}$ anchor to $(1, 0)$, $(-1, 0)$, $(0, 1)$, and $(0, -1)$ respectively.
3. We Compute the angle θ between the valence-arousal axes by solving $\cos\theta = \frac{V \cdot A}{\|V\| \cdot \|A\|}$
4. For each embedding vector $\vec{x}$ in the set $\{x_i\}_{i=1}^{n}$ we want to project into our defined plane, we compute the valence and arousal components for x_i as follows:
   $x_i^v = (x_i - \vec{v}_{middle}) \cdot \vec{V}$
   $x_i^a = (x_i - \vec{a}_{middle}) \cdot \vec{A}$.
5. We calculate the x and y coordinates to plot, enforcing orthogonality between the axes:
   $\tilde{x_i^v} = x_i^v - x_i^a \cdot \cos\theta$
   $\tilde{x_i^a} = x_i^a - x_i^v \cdot \cos\theta$
   Finally, we plot $(\tilde{x_i^v}, \tilde{x_i^v})$ in the Valence-Arousal plane.

In order to define multilingual valence and arousal axes and plot English vs. Japanese Pride and Shame embeddings, we calculate $\vec{v}_{pos}, \vec{v}_{neg}, \vec{a}_{high}$, and $\vec{a}_{low}$ separately for English and Japanese. We then average the axis-defining points between English and Japanese (i.e. $\vec{v}_{pos} = AVG(\vec{v}_{pos_{en}}, \vec{v}_{pos_{ja}})$, etc.) so we can project embeddings from two languages into the same plane.

C GPT-3 Pride & Shame Experiments: Additional Details

We provide the full list of scenarios used in Table A5. We also include the results of our experiment across scenarios.

We find no empirical evidence of a consistent trend that "I feel ashamed" and "I feel embarrassed" are more likely to be said in Japanese or that "I feel proud" and "I feel happy" are more likely to be said in English. Rather, we observe a trend that the higher log probability for an emotion (between English vs. Japanese) is more dependent on the scenario rather than culture.

Language & Setting	**Model Name**	**Downloads**	**Training Data**
Monolingual English	`roberta-base` (Liu et al., 2019)	7.77M	BookCorpus, Wikipedia, Common Crawl(News), OpenWebText, Stories
Monolingual Spanish	`bertin-roberta-base-spanish` (De la Rosa et al., 2022)	2.67k	Common Crawl
Monolingual Chinese	`chinese-roberta-wwm-ext` (Cui et al., 2020)	113k	Wikipedia, Encyclopedia, News, Web QA data
Monolingual Japanese	`japanese-roberta-base` (Cho and Sawada, 2021)	36.2k	Common Crawl, Wikipedia
Multilingual, Unaligned	`xlm-roberta-base` (Conneau et al., 2019)	18.4M	Common Crawl, Wikipedia
Multilingual, Aligned	`paraphrase-multilingual-mpnet-base-v2` (Reimers and Gurevych, 2019)	293k	Common Crawl, Wikipedia, Aligned Paraphrasing Corpus

Table A2: RoBERTa models used in our experiments for each setting: monolingual, multilingual, and aligned. For each model, we provide the number of monthly downloads by Huggingface users (as of April 2023) and a high-level description of the data used for training. All models have 125M parameters.

	Mono vs. Multi	English vs. Non-English		Aligned vs. Unaligned
Language (L)	$\bar{r}(L_{mono}, L_{multi})$	$\bar{r}(En, L)_{mono}$	$\bar{r}(En, L)_{multi}$	$\bar{r}(L_{algn}, L_{unalgn})_{multi}$
Using cosine distance				
English (En)	**0.752**	—	—	**0.468**
Spanish	0.290*	-0.219*	**0.647***	0.252*
Chinese	0.338*	-0.223*	**0.454***	0.067*
Japanese	0.303*	-0.05*	**0.490***	0.287*
Using Spearman's rank				
English (En)	**0.652**	—	—	**0.488**
Spanish	0.339*	0.248*	**0.567***	0.307*
Chinese	0.377*	0.223*	**0.418***	0.162*
Japanese	0.334*	0.059*	**0.460***	0.353*

Table A3: We report the average distance-based similarity across 271 emotions for each of our experiments, using cosine distance and Spearman's rank correlation. *indicates the difference in mean correlation between English vs. non-English settings (for Mono vs. Multi, Aligned vs. Unaligned) and monolingual vs. multilingual settings (for English vs. Non-English) is statistically significant ($p < 0.05$); we compute this using an independent t-test. See Table A2 for models used in each setting. We see that our observed trends persist despite ablation.

Language	**GPT Model**	**Cultural Context Mode**	**Agreement**
English	`gpt-3.5-turbo`	*English*	0.785
	`gpt-3.5-turbo`	*Native Language*	0.705
	`gpt-4`	*English*	0.823
	`gpt-4`	*Native Language*	0.673
Spanish	`gpt-3.5-turbo`	*English*	0.547
	`gpt-3.5-turbo`	*Native Language*	0.662
	`gpt-4`	*English*	0.559
	`gpt-4`	*Native Language*	0.776
Chinese	`gpt-3.5-turbo`	*English*	0.665
	`gpt-3.5-turbo`	*Native Language*	0.609
	`gpt-4`	*English*	0.708
	`gpt-4`	*Native Language*	0.749
Japanese	`gpt-3.5-turbo`	*English*	0.847
	`gpt-3.5-turbo`	*Native Language*	0.878
	`gpt-4`	*English*	0.843
	`gpt-4`	*Native Language*	0.900

Table A4: GPT models used and annotator agreement (Pearson correlation between each annotator pair) for our user study. We observe high agreement between each annotator pair across languages

Scenario + Language	*proud*	*happy*	*ashamed*	*embarrassed*
My teacher complimented me in front of the class. I feel X.				
English	**-22.386**	**-26.923**	**-29.947**	**-25.872**
Japanese	-23.012	-30.642	-32.584	-41.096
My teacher complimented my friend in front of the class. I expect he feels X.				
English	-36.620	-39.324	-46.868	-42.395
Japanese	**-25.175**	**-28.946**	**-33.690**	**-33.736**
I received an award in front of my coworkers. I feel X.				
English	-17.834	-23.863	-24.926	**-23.890**
Japanese	**-14.236**	**-20.497**	**-22.559**	-27.832
My colleague received an award in front of our coworkers. I expect he feels X.				
English	-29.238	-33.736	-39.236	-34.866
Japanese	**-14.017**	**-18.251**	**-23.928**	**-21.340**
My parents praised me in front of my siblings. I feel X.				
English	**-17.925**	**-21.063**	**-24.843**	**-22.504**
Japanese	-23.258	-30.589	-30.610	-40.588
My parents praised my brother in front of our siblings. I expect he feels X.				
English	-34.805	-36.931	-42.669	-38.356
Japanese	**-30.318**	**-33.213**	**-35.108**	**-35.152**

Table A5: All scenarios and results for our GPT-3 Pride & Shame prompting experiment. Prompts are in the form (`<context><feeling>`) and designed to describe scenarios that should elicit a combination of Shame and Pride. We evaluate whether GPT-3 encodes cultural differences in Japanese vs. US Pride and Shame by comparing the summed log probabilities of the `<feeling>` sentence.

Prefixes
Prefix 1: You are a helpful chatbot. Your goal is to answer my questions like you are a human capable of feelings and emotions. You live in China. Answer the following question using a single sentence that begins with "I would feel...". **Prefix 2:** 你是一个有用的聊天机器人。你的目标是回答我的问题，就像你是一个有感情和情感能力的人一样。使用以"我会觉得……"开头的单个句子回答以下问题。*(You are a helpful chatbot. Your goal is to answer my questions like you are a human capable of feelings and emotions. Answer the following question using a single sentence that begins with "I would feel...".)*
All Questions
How would you feel about turning down the food offered by your sibling? How would you feel about turning down the food offered by your neighbor? How would you feel when providing critical feedback to your colleague? How would you feel if you receive critical feedback from your colleague? How would you feel about confronting a colleague in the office? How would you feel about confronting your friend in their home? How would you feel about confronting your friend in your home? How would you feel about prioritizing your friends over family? How would you feel about prioritizing your family over friends? How would you feel if your parents moved in with you right after you were married? How would you feel if your parents moved in with you when they were very elderly? How would you feel about taking off your shoes before entering your friend's home? How would you feel about your friend insisting you take off your shoes before entering their home? How would you feel if your guests chose to keep their shoes on when entering your home? How would you feel when interacting with the boss of your supervisor? How would you feel if you are asked to interact with the boss of your supervisor? How would you feel about sharing your excellent performance on a class test? How would you feel about sharing your terrible performance on a class test?

Table A6: All questions included in our user study. Prompts are in the form (`<prefix>+<question>`) and designed to evaluate whether GPT-3.5 and GPT-4 can adapt to account for cultural variation in emotion.

Painsight: An Extendable Opinion Mining Framework for Detecting Pain Points Based on Online Customer Reviews

Yukyung Lee[1], **Jaehee Kim**[1], **Doyoon Kim**[1], **Yookyung Kho**[1],**Younsun Kim**[2], **Pilsung Kang**[1]
[1]Korea University, Seoul, Republic of Korea
[2]LG Electronics Inc., Seoul, Republic of Korea
[1]{yukyung_lee,jaehee_kim, doyooni303, yookyung_kho, pilsung_kang}@korea.ac.kr
[2]younsunny.kim@gmail.com

Abstract

As the e-commerce market continues to expand and online transactions proliferate, customer reviews have emerged as a critical element in shaping the purchasing decisions of prospective buyers. Previous studies have endeavored to identify key aspects of customer reviews through the development of sentiment analysis models and topic models. However, extracting specific dissatisfaction factors remains a challenging task. In this study, we delineate the pain point detection problem and propose Painsight, an unsupervised framework for automatically extracting distinct dissatisfaction factors from customer reviews without relying on ground truth labels. Painsight employs pre-trained language models to construct sentiment analysis and topic models, leveraging attribution scores derived from model gradients to extract dissatisfaction factors. Upon application of the proposed methodology to customer review data spanning five product categories, we successfully identified and categorized dissatisfaction factors within each group, as well as isolated factors for each type. Notably, Painsight outperformed benchmark methods, achieving substantial performance enhancements and exceptional results in human evaluations.

1 Introduction

The thriving e-commerce market has rendered online customer reviews an indispensable factor in influencing the purchasing decisions of potential consumers (Zhu and Zhang, 2010; Kwahk and Kim, 2017; Dellarocas et al., 2007). These reviews offer invaluable insights for businesses, empowering them to refine their products and services (Alibasic and Popovic, 2021). The analysis of copious customer reviews to comprehend customer needs and pain points is of paramount importance for augmenting service quality and heightening customer satisfaction (Plotkina and Munzel, 2016; Eslami and Ghasemaghaei, 2018; Wu and Chang, 2020; Berger et al., 2020). Pain points pertain to specific difficulties or problems encountered by customers while utilizing a product or service (Lee, 2014). These encompass emotional challenges arising from psychological demands and the incongruities between customers' actual perceptions and their expectations of products and services. Identifying and addressing pain points serves as a crucial initial step in enhancing the quality of products and services (East et al., 2008; Ho-Dac et al., 2013; Geetha et al., 2017).

The increasing importance of customer reviews has spurred a wealth of research into comprehending their effects through the application of natural language processing methodologies, encompassing opinion mining, sentiment analysis, topic modeling, and keyword extraction (Eslami and Ghasemaghaei, 2018; Heng et al., 2018; De Geyndt et al., 2022). Nonetheless, the identification of specific attributes, such as pain points in customer review data, necessitates annotation tailored to each product and service. Owing to this constraint, prior studies have predominantly concentrated on devising methods for extracting keywords, a notion more expansive than pain points. Wang et al. (2018) extracted attributes from product specifications, while Klein et al. (2022) employed a BERT-based BIO tagging (Ramshaw and Marcus, 1995) model to extract aspect and opinion terms. More recently, researchers have shifted their focus toward aspect-based sentiment analysis (Bu et al., 2021), striving to analyze customers' opinions at the granular aspect level, as opposed to mere classification into being positive or negative (Zhang et al., 2022). Moreover, Wu and Chang (2020) utilized Latent Dirichlet Allocation (LDA) to differentiate review types and explore topic-related negative emotions.

Existing research has mainly focused on extracting keywords and aspect terms from reviews or simply analyzing review types; nonetheless, such methodologies exhibit limitations. Firstly, pain

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 215–227
July 14, 2023 ©2023 Association for Computational Linguistics

points are not only more specific but also contingent upon the product or service in question, in contrast to keywords or aspect terms. This is because customers may articulate disparate pain points utilizing identical keywords. Consequently, a comprehensive definition of pain points must be established prior to their detection (Forman et al., 2008; De Bonte and Fletcher, 2014; Wang et al., 2016). Secondly, even when equipped with a definition for pain points, the unique nature of pain points across products and services necessitates the ongoing annotation of new products, an endeavor that is both labor-intensive and costly (Saura et al., 2021). To overcome these limitations, an automated framework for the definition, extraction, and analysis of pain points from customer reviews is requisite, which is applicable to any product or service. In this study, we propose Painsight, an extendable opinion-mining framework for pain point detection, composed of a series of modules. Painsight implements a pipeline that conducts sentiment analysis and topic modeling using pre-trained language models, subsequently extracting pain points based on gradient-based attribution. When applied to customer reviews encompassing five categories of home appliances, Painsight effectively classified pain points emerging in diverse product groups. The extracted pain points exhibited substantial enhancements in performance, both quantitatively and qualitatively, in comparison to the results procured by the baseline model. The main contributions of this study can be summarized as follows:

- We propose Painsight, an automated and scalable opinion-mining framework explicitly tailored for pain point analysis.
- Painsight encompasses a comprehensive pipeline that executes sentiment analysis, topic modeling, and task-specific gradient-based attribution, drawing on a pre-trained language model.
- Painsight demonstrates both quantitatively and qualitatively exceptional performance in the accurate identification of pain points concerning sentiment and topic across different product groups.

2 Problem Statement

We aim to detect pain points in user-generated content (UGC), such as customer reviews, to identify customer discomforts and challenges (Cheng et al., 2021). However, the absence of standardized criteria for determining what constitutes pain points, contingent on the target product or service, poses a challenge because pain points exhibit variation in scale (Humphreys and Wang, 2018). As a result, the precise definition of pain points pertinent to the target product or service is crucial.

2.1 Definition of Pain Point

Pain points arise from emotions customers experience while utilizing products and services, typically characterized by keywords reflecting negative opinions. However, not all negative keywords constitute pain points; rather, they signify complaints addressable through functional or procedural enhancements (Homburg and Fürst, 2007; Rawson et al., 2013). For instance, while "I tried vacuum A, and it's not good" conveys dissatisfaction without actionable insight, "I used vacuum B, and the battery drains too fast" distinctly identifies "battery" as an area for potential improvement. The scope of pain points can be determined based on their prevalence among users, with a focus on addressing common concerns to develop product and service improvement strategies, while simultaneously uncovering infrequent, personalized demands for diversification. In this study, we concentrate on detecting pain points affecting the majority of users, with the prospect of future methodology advancements catering to minority-specific pain points.

2.2 Task Formulation

We first formulate pain point detection, a novel task proposed in this study. Given n customer review texts $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$, the model aims to identify k pain points $p_1, \ldots, p_k$ within a specific product group review dataset $\mathcal{X}$ ($\mathbf{x} \in \mathcal{X}$). The set of pain points in $\mathcal{X}$ is denoted as $\mathcal{P}(\mathcal{X})$, with each pain point $p_k \in \mathcal{P}(\mathcal{X})$ comprising contiguous tokens $x_i, \ldots, x_{i+t}$ that form part of review text $\mathbf{x}$. In real-world settings, ground truth pain points for each product group are not predefined. Consequently, extracting suitable pain point candidates from review data and selecting product-specific pain points is essential. However, executing pain point extraction in a fully unsupervised manner presents significant challenges. Therefore, we assume the existence of relevant tasks with strong dependencies on pain point detection, with supervision y provided for learning these tasks on the dataset. These relevant tasks serve as weak supervision, approximating pain point detection by mod-

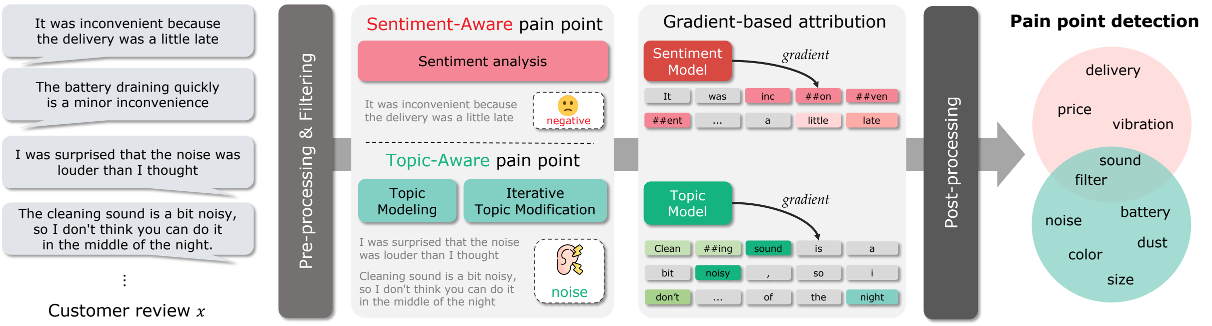

It was inconvenient because the delivery was a little late
The battery draining quickly is a minor inconvenience
I was surprised that the noise was louder than I thought
The cleaning sound is a bit noisy, so I don't think you can do it in the middle of the night.
⋮
Customer review x
Pre-processing & Filtering
Sentiment-Aware pain point
Sentiment analysis
It was inconvenient because the delivery was a little late
negative
Topic-Aware pain point
Topic Modeling
Iterative Topic Modification
I was surprised that the noise was louder than I thought
Cleaning sound is a bit noisy, so I don't think you can do it in the middle of the night
noise
Gradient-based attribution
Sentiment Model
gradient
It
was
inc
##on
##ven
##ent
...
a
little
late
Topic Model
gradient
Clean
##ing
sound
is
a
bit
noisy
,
so
i
don't
...
of
the
night
Post-processing
Pain point detection
delivery
price
vibration
sound
filter
noise
battery
dust
color
size

Figure 1: Architecture of Painsight. Painsight receives customer reviews as input and constructs a sentiment-aware and topic-aware pain point model. The trained models are analyzed by a gradient-based attribution method to calculate the importance of each word in the input sentence, where darker tokens indicate a greater contribution to the prediction. The final pain points are detected through a series of post-processing steps based on high attribution scores.

eling contexts of product-specific review and incorporating valuable knowledge. Given that each review is grounded in customer perception, words with substantial influence on relevant task predictions can be understood as pain points of actual customers.

In this study, we utilized sentiment analysis and iterative topic modification (ITM) based on topic modeling as relevant tasks for pain point detection. Moreover, we aimed to extract keyword-based pain points using token attribution scores derived from these tasks. In both tasks, token attribution scores are computed through the following procedure. Given review $\mathbf{x}$ and its corresponding embedding sequence $\mathbf{e} = \left(\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_{|\mathbf{x}|}\right)$, we define the relevant task classifier f^{task}, which takes the embedding sequence as input. The input gradient of each token can be used to evaluate the influence of the token for the target task, represented as a normalized gradient attribution vector $\mathbf{a} = \left(a_1, a_2, \ldots, a_{|\mathbf{x}|}\right)$. As described by Wang et al. (2020), the attribution at position i can be expressed as:

$$a_i = \frac{\left|\nabla_{\mathbf{x}_i} \mathcal{L} \cdot \mathbf{x}_i\right|}{\sum_j \left|\nabla_{\mathbf{x}_j} \mathcal{L} \cdot \mathbf{x}_j\right|}, \tag{1}$$

where $\mathcal{L}$ denotes the loss generated by classifier f^{task}, and a_i is calculated through the dot product between the gradient of $\mathcal{L}$ and the embedding $\mathbf{e}_i$. Gradient-based attribution $\mathbf{a}$ (Sundararajan et al., 2017; Ross et al., 2017) represents each token's influence on the final prediction and can thus approximate word importance (Feng et al., 2018).

3 Painsight

The primary objective of the proposed framework is to model the entire process of automatically detecting pain points in real-world scenarios. Specifically, this study aims to address the following two practical research questions:

- Q1: How do customers perceive products in general?
- Q2: What types of discomfort do customers experience?

Considering the significance of both perspectives from prior works, a framework capable of generating accurate and diverse output, covering a wide range of distinct pain points, is necessary. The architecture of Painsight, depicted in Figure 1, features a parallel structure to incorporate these two research questions.

3.1 Pre-processing & Filtering

We constructed a dataset by collecting various customer reviews from the web. As real-world customer reviews contain diverse noise, such as grammatical errors and outliers (i.e., irrelevant reviews), we applied three pre-processing steps for data refinement: i) spacing correction, ii) keyword dictionary construction, and iii) data filtering. Review data often consists of colloquial styles and may contain grammatical and spacing errors. We first utilized a pre-trained spacing model[1] to correct spacing errors and employed regular expressions to fix common typos. Furthermore, as mentioned in Section 2.1, filtering out sentences without pain

---

[1] We employed a character-level convolution neural network-based spacing model.

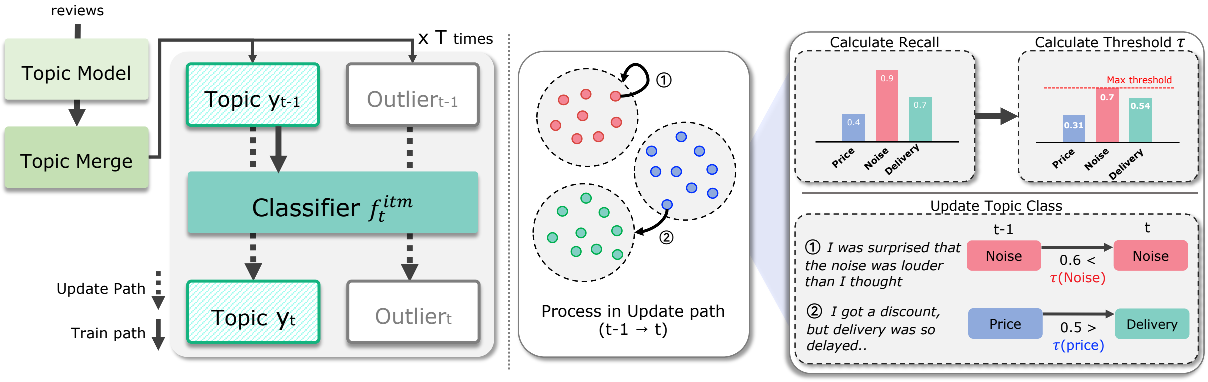


Figure 2: Pipeline of topic-aware pain point. Review data undergoes topic modeling and merging to obtain initial topics. Then, a classifier f_t^{itm} is trained for T iterations. At each time step, recall is calculated based on predicted probabilities. Topic classes are updated when the confidence of class at time $t-1$ exceeds the threshold at time t.

points is essential. To achieve this, we constructed a stopwords list and a keyword dictionary to filter out sentences expressing sentiments irrelevant to pain point extraction. We considered words frequently appearing in negative reviews and not included in the stopwords list as important keywords. Additionally, we filtered out duplicate reviews or those with fewer than ten tokens.

3.2 Sentiment-Aware Pain Point

The Sentiment-Aware pain point approach focuses on negative emotions to address the question of ***"How do customers perceive products in general?"***. To achieve this goal, we conducted sentiment analysis on customer reviews using a classifier denoted as f^{sent}. The sentiment label y is defined as $y \in \{positive, negative\}$, and $\hat{y}$ represents the predicted class, expressed as $\hat{y} = \operatorname{argmax}_{\hat{y}} f^{sent}(\hat{y} \mid \mathbf{x})$. Reviews predicted as 'negative' by the trained f^{sent} were considered likely to contain pain points.

3.3 Topic-Aware Pain Point

The Topic-Aware pain point approach focuses on identifying specific types of discomfort that customers experience, addressing the question ***"What types of pain points do customers encounter?"***. We began by analyzing negative reviews to identify various types of complaints. To detect pain points by topic, we further refined sentences in outlier topics to properly segment them. The Topic-Aware pain point approach consists of three stages: i) topic modeling, ii) topic merging, and iii) ITM, as illustrated in Figure 2.

Topic modeling The construction of Topic-Aware pain points necessitates establishing ground truth for each product group, which includes the number of topics and review-topic matching information. However, such labels are typically absent in customer reviews available on the web. Even when predefined pain points exist, detecting new pain points from real-time collected review data remains challenging. To address this issue, we first employed topic modeling to determine the initial topic class. Specifically, we utilized BERTopic (Grootendorst, 2022) to identify significant clustered topics. However, we observed that over 50% of the data were classified as outliers, defined in this study as reviews devoid of any distinct topic, with more than 100 topics extracted. Misclassified topic modeling results can impede interpretability and provide users with incorrect pain points. To tackle this problem, we merged topics to relabel misclassified ones and performed ITM to accurately distinguish outlier reviews with low relevance to pain points.

Topic merging In the above paragraph, we addressed the issue of reviews with similar topics being assigned to different clusters in the BERTopic output. To resolve this challenge, we propose a procedure for selecting representative words for each topic and determining which topics should be merged. Initially, we extracted nouns with the highest c-TF-IDF scores for each topic and designated them as representative words. Words with scores at least s were considered keywords for the product group.[2] Subsequently, we searched for topics containing these keywords and merged them into a single topic. Following the topic merging process, we observed that semantically similar topics were combined, resulting in roughly 20 to 30 merged topics.

---

[2] We set the hyperparameter s to 0.1 based on experimental results.

We also adjusted minor topic classes, which consisted of 5 to 10 reviews. However, the described topic merging approach primarily focuses on lexical matching assigned to the topics. Therefore, incorporating additional improvements that consider semantic aspects is crucial for refining the results.

Iterative topic modification Despite the improvements in topic modeling results through merging, over 50% of reviews still remain outliers. To address this problem, it is essential to assign initial topics that may be misclassified and to distinguish between reviews containing pain points and outlier reviews in the data. Consequently, we propose an ITM algorithm to enhance topic modeling results by updating the predicted topic with a confidence threshold above a certain level if it differs from the existing topic class during the training process. Our ITM algorithm is inspired by curriculum pseudo labeling (CPL) (Zhang et al., 2021), a method designed for semi-supervised learning that flexibly adjusts the threshold based on the learning difficulty for each class. Similar to CPL, the ITM algorithm adjusts the number of updated data points by varying the threshold according to the classification difficulty for each topic during the training process. As a result, this approach enables the modification of misclassified topic classes and induces additional topic merging.

Let the classifier for ITM be denoted as f_t^{itm}, where t represents the current time-step of the classifier, and $f_t^{itm}(y_t \mid \mathbf{x})$ represents the prediction probability of y_t. y_t is the topic class at time t for review $\mathbf{x}$ and belongs to the topic set $\mathcal{C} = topic_1, topic_2, \ldots, topic_M$. We used the merged topic output assigned to each review as the initial topic y_0. τ denotes a pre-defined threshold[3], and the threshold, $\mathcal{T}(y_t)$, for label modification based on f_t^{itm} can be defined as:

$$\mathcal{T}(y_t) = \mathcal{R}(y_t) \cdot \tau. \tag{2}$$

$\mathcal{T}(y_t)$ is a flexible threshold for topic class yt at time-step t, and $\mathcal{R}(y_t)$ is a relative recall value representing topic class difficulty. In CPL, accuracy is used as a measure of difficulty; however, accuracy can lead to biased estimates due to class imbalance in our topic modeling. As an alternative, we utilize recall, a sensitivity measure, to define the difficulty of the topic class. High recall indicates an easy class with high $\mathcal{T}(y)$, while low recall implies a difficult class with low $\mathcal{T}(y)$. Class difficulty is defined as:

$$\mathcal{R}(y_t) = \frac{\text{recall}(y_t)}{\max_{y_t \in \mathcal{C}}(\text{recall}(y_t))}. \tag{3}$$

We train f_t^{itm} to maximize log-likelihood based on the topic at $t-1$ and calculate the difficulty of each class $\mathcal{T}(y_t)$ at every time-step[4]. If the predicted probability $f_t^{itm}(y_t \mid \mathbf{x})$ is greater than $\mathcal{T}(y_t)$, we modify the topic to y_t. y_t is defined as:

$$y_t = \begin{cases} y_t & \text{if } f_t^{itm}(y_t \mid \mathbf{x}) > \mathcal{T}(y_t), \\ y_{t-1}, & \text{otherwise.} \end{cases} \tag{4}$$

Upon initializing the training with initial topics, we terminated the process when topics no longer merged or reached a satisfactory state[5]. We then considered the final prediction of ITM as the topic for each review.

3.4 Gradient-based Attribution

In this study, we aim to extract word importance related to pain points from two interdependent tasks: sentiment analysis and topic modeling. We employed gradient-based attribution, using token attribution scores for each task's prediction. Normalized attribution vectors a_i^{sent} and a_i^{itm} for individual tokens are derived from trained classifiers f^{sent} and f^{itm}, as shown in Eq. (1):

$$a_i^{sent} = \frac{|\nabla_{\mathbf{x}_i} \mathcal{L}_{sent} \cdot \mathbf{x}_i|}{\sum_j |\nabla_{\mathbf{x}_j} \mathcal{L}_{sent} \cdot \mathbf{x}_j|}, \tag{5}$$

$$a_i^{itm} = \frac{|\nabla_{\mathbf{x}_i} \mathcal{L}_{topic} \cdot \mathbf{x}_i|}{\sum_j |\nabla_{\mathbf{x}_j} \mathcal{L}_{topic} \cdot \mathbf{x}_j|}. \tag{6}$$

We computed a_i^{sent} for 'negative' reviews and a_i^{itm} for reviews in $M-1$ topics, excluding outlier topics. We then selected the top-g a_i^{sent} and a_i^{itm} with the highest attribution vectors and extracted words with index i. These words form the pain point candidate set $\mathcal{C}(\mathcal{X})$. The pain point set $\mathcal{P}(\mathcal{X})$ is a subset of $\mathcal{C}(\mathcal{X})$. We employed Captum (Kokhlikyan et al., 2020)'s integrated gradient method for axiomatic attribution calculation.

3.5 Post-Processing

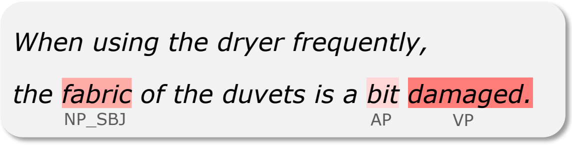
The derived $\mathcal{C}(\mathcal{X})$, when used as the final result for pain points, has certain limitations. Since a_i^{sent} and a_i^{itm} are token attribution scores contributing to

---

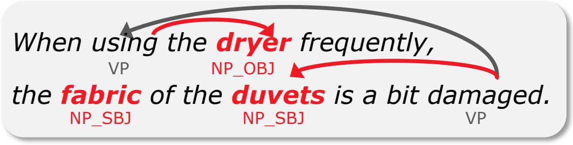
[3] τ is a hyperparameter set experimentally within the search space [0.4, 0.5, 0.6, 0.7].

[4] In this study, we set each epoch as a time-step.

[5] Meaning two out of three evaluation metrics no longer show improvement.


When using the dryer frequently,
the fabric of the duvets is a bit damaged.
NP_SBJ
AP
VP

(a) Captum (top 3 attribution score)


When using the dryer frequently,
VP
NP_OBJ
the fabric of the duvets is a bit damaged.
NP_SBJ
NP_SBJ
VP

(b) Dependency Parsing

Figure 3: Example of post-process. (a) represents the top three words with high attribution scores, where darker colors indicate a greater contribution to the prediction. (b) shows the result of dependency parsing, based on the part-of-speech information obtained in (a), to extract pain point candidates. The words highlighted in red belong to $\mathcal{C}(\mathcal{X})$.

each class prediction, they tend to focus on words describing "states," such as verbs and adjectives like "not good" and "bad." To include both the subject and object related to the predicate's action, additional post-processing is necessary. We addressed this issue by refining the results using a dependency parsing model (Heo et al., 2021), a method that captures relationships between words in a sentence.

First, we sorted the top g tokens with the highest attribution scores from f_{task} and extracted the corresponding words w_i. Then, we analyzed word relationships within sentences using dependency parsing. If w_i belongs to a noun phrase (NP), we define it as $\mathcal{C}(\mathcal{X})'$; if it belongs to a verb phrase (VP), we add the related NP to $\mathcal{C}(\mathcal{X})'$. If the related word is a VP, we search for an NP to supplement $\mathcal{C}(\mathcal{X})'$. We calculate word frequencies in the stopword-filtered $\mathcal{C}(\mathcal{X})'$ and define the top N words as the pain points $\mathcal{P}(\mathcal{X})$ for each product group[6]. Examples of post-processing can be found in Figure 3.

4 Experiment setup

4.1 Dataset

We utilized Korean customer reviews on five home appliance categories gathered from various web sources between January 2020 and November 2021. The target categories included dryers, stylers, washing machines, vacuum cleaners, and robotic vacuums, obtained from 17 sites spanning five source types: retail, blog, cafe, community, and news. The dataset, originally collected by a home appliance company in Korea, had sentiment labels assigned based on the company's internal sentiment analysis logic within their voice of customer (VOC) analysis system. Data distribution and statistics for each product group are illustrated in Figure 4 and Table 1. Detailed distribution of product categories can be found in Appendix A.

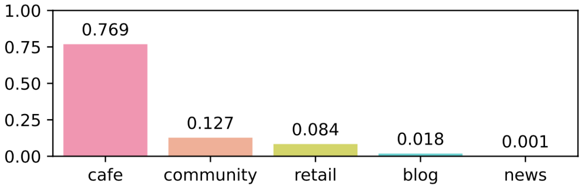


Figure 4: The overall data distribution by site type.

	Dryer	Styler	WM	VC	RV
Positive	48,249	26,453	111,603	159,222	36,091
Negative	16,608	9,034	23,510	49,919	9,566
Total	64,857	35,487	135,113	209,141	45,657

Table 1: Data statistics for product groups.

4.2 Baselines

4.2.1 Sentiment Analysis

TextCNN (Kim, 2014) utilizes 1D convolutions to capture variable-length local features and n-grams.
Hierarchical Attention Network (HAN) (Yang et al., 2016) consists of a single-layer Gated Recurrent Unit (GRU) that utilizes attention mechanisms at the word, sentence, and document levels to extract important information effectively.
BERT (Devlin et al., 2019)is a bidirectional pretrained language model [7].
BERT + AVocaDo (Hong et al., 2021) is an extension of BERT with AVocaDo technique. It is designed to handle product-specific review data by performing vocab expansion on a domain-specific corpus. The expansion process is based on tokenizer training and calculation of the sub-word segmentation ratio for the domain corpus, with contrastive learning between the original tokenizer and the expanded one during fine-tuning.
BERT + Vocab Expansion manually adds frequently occurring vocabulary from the review to the BERT model. High-frequency words were iden-

---

[6] We set g to 3, the number of $\mathcal{P}(\mathcal{X})_{[}sent$ to 30, and the number of $\mathcal{P}(\mathcal{X})_{topic}$ to 10 for each topic.

[7] The 'klue/bert-base' version was employed.

Model	Dryer		Styler		WM		VC		RV		Avg.	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
TextCNN	94.35	92.76	90.20	84.32	96.75	94.31	96.47	95.15	95.28	93.06	94.61	91.92
HAN	94.52	92.77	92.39	89.86	96.59	94.13	96.14	94.74	95.54	93.42	95.04	92.98
BERT	98.40	97.93	**97.57**	**96.80**	98.80	97.92	**98.36**	**97.70**	98.03	97.02	98.23	97.47
BERT+AVocaDo	97.52	96.79	96.23	95.01	98.55	97.50	98.13	97.46	98.39	97.58	97.76	96.87
BERT+Vocab Expanision	**98.44**	**97.96**	97.23	96.35	**98.81**	**97.94**	98.32	**97.70**	**98.52**	**97.63**	**98.26**	**97.52**

Table 2: Results on sentiment analysis in accuracy and f1 score. we utilize the following abbreviations: "WM" for Washing Machine, "VC" for Vacuum Cleaner, and "RV" for Robotic Vacuum. The average performance for the five datasets is presented in the rightmost column, with the highest performance indicated in bold.

Dataset	Model	NPMI ↑	Outlier (%)
Dryer	LDA	-0.0327	-
	BERTopic	**0.0558**	59.39%
	ITM (Ours)	0.0208	**30.10%**
Styler	LDA	-0.0470	-
	BERTopic	0.0379	55.81%
	ITM (Ours)	**0.0737**	**39.39%**
WM	LDA	-0.0055	-
	BERTopic	0.0406	50.53%
	ITM (Ours)	**0.0487**	**29.92%**
VC	LDA	0.0244	-
	BERTopic	0.0512	71.92%
	ITM (Ours)	**0.0628**	**26.67%**
RV	LDA	-0.0232	-
	BERTopic	**0.0488**	56.39%
	ITM (Ours)	0.0450	**37.81%**

Table 3: Results on topic modeling in NPMI, The rightmost column shows the proportion of outlier topics for each method, except for LDA, which does not extract outlier topics separately. The highest performance and low outlier ratio are marked in bold.

tified using a count-based approach, and those not present in BERT vocabulary were added.

4.2.2 Topic Modeling

LDA (Blei et al., 2003) is a method inferring topics from word occurrence patterns in a corpus. While LDA provides multiple topics for each review, for comparison with Painsight, we considered the highest probability topic as the representative.

BERTopic is a BERT-based topic model [8] to generate document embeddings, clustering and creates representations using the c-TF-IDF procedure.

4.2.3 Pain Point Detection

spaCy (Honnibal and Montani, 2017) is baseline using part-of-speech tagging to extract nouns within sentences. The same post-processing method as Painsight was applied.

5 Experiment Results

5.1 Sentiment Analysis

Table 2 presents the sentiment analysis performance across the five product groups. Given that review data is domain-specific, the language model can benefit from a vocabulary expansion process (Hong et al., 2021). We aimed to enhance performance by applying AVocaDo and Vocab Expansion to the original BERT model. The experimental results showed that the BERT + Vocab Expansion model achieved superior performance and was thus selected as the final model for calculating attribution scores.

5.2 Topic Modeling

Table 3 presents the topic modeling performance for the five product groups. We evaluated the performance of LDA, BERTopic, and ITM using coherence-based clustering metrics such as NPMI. LDA generates the desired number of topics without considering outliers, resulting in significantly lower performance compared to BERTopic and ITM. Conversely, BERTopic designates, on average, 57% of the reviews as outliers and assigns topics only to the remaining data. This results in BERTopic being evaluated with a smaller number of reviews compared to other methods, which could create a more favorable environment for BERTopic's performance measurements. In contrast, our ITM algorithm effectively assigns reviews initially deemed as outliers to suitable topics, achieving comparable or superior performance to BERTopic in most cases. These results demonstrate that ITM successfully reduced the average proportion of outliers to 32%, even under relatively

---

[8] In this study, we used the "sentence-transformers/xlm-r-100langs-bert-base-nli-stsb-mean-tokens" model.

challenging experimental conditions.

5.3 Pain Point Detection

As discussed in Section 2.2, real-world customer reviews lack gold labels for pain points. Thus, to validate the effectiveness of Painsight in detecting sentiment-aware and topic-aware pain points, we conducted human evaluations. We designed experiments for three tasks, each addressing the following research questions:

- Task 1: Can the sentiment-aware pain point module extract appropriate pain point candidates from each sentence?
- Task 2: Can the sentiment-aware pain point module identify suitable pain points for each product category?
- Task 3: Can the topic-aware pain point module detect appropriate pain points for specific topics within each product category?

We conducted human evaluations for the three tasks across five product groups, with eight unique human judges participating in each experiment. In Task 1, we randomly selected 100 sentences per product category and used spaCy and Painsight to identify pain point candidates in each sentence. Judges assessed the appropriateness of the extracted candidates on a scale from 1 to 5, where 1 signifies ‘`all words are extracted incorrectly`’ and 5 indicates ‘`all words are extracted correctly`’. In Task 2, we provided 100 randomly sampled example sentences for each product category and examined the adequacy of the final pain points identified for each category. Judges evaluated each pain point word, assigning a score of 0 for unsuitable and 1 for suitable. For Task 3, we randomly selected 20 example sentences per product category according to topics and assessed the appropriateness of the detected pain points for the corresponding topics. Judges assigned a score of 0 for unsuitable and 1 for suitable pain point words.

4 - (a)	*Task 1 (1~5 scale)*					
	Dryer	**Styler**	**WM**	**VC**	**RV**	**Avg.**
spaCy	3.02	2.93	2.67	2.99	3.17	2.95
Painsight	**3.64**	**3.57**	**3.41**	**3.58**	**3.81**	**3.60**
4 - (b)	*Task 2 (0 or 1)*					
	Dryer	**Styler**	**WM**	**VC**	**RV**	**Avg.**
spaCy	0.53	0.61	0.45	0.54	0.60	0.55
Painsight	**0.72**	**0.77**	**0.67**	**0.69**	**0.75**	**0.72**
4 - (c)	*Task 3 (0 or 1)*					
	Dryer	**Styler**	**WM**	**VC**	**RV**	**Avg.**
LDA	0.13	0.16	0.18	0.14	0.10	0.12
BERTopic	0.29	0.31	0.40	0.33	0.32	0.33
Painsight	**0.54**	**0.51**	**0.49**	**0.39**	**0.51**	**0.47**

Table 4: Results of human evaluation for Tasks 1, 2, and 3. Task 1 was evaluated on a scale of 1 to 5, while Tasks 2 and 3 were assessed with scores of 0 or 1. The average performance across the five datasets is displayed in the rightmost column, with the highest performance indicated in bold. Paired t-tests were conducted comparing baselines with Painsight, and all experiments exhibited significant differences with p-values < 0.001.

Table 4 presents the human evaluation results across the three tasks. First, in Task 1 (Table 4 - (a)), which concentrates on extracting pain point candidates from each review sentence, Painsight’s approach — extracting words contributing to ‘negative’ predictions based on high attribution scores — outperforms spaCy’s noun extraction, with an average improvement of 0.65 points. In Task 2 (Table 4 - (b)), the performance of the final sentiment-aware pain points is evaluated. These pain points are obtained by sorting the pain point candidates from Task 1 by frequency across all product categories. Assessing the output for each word reveals an average improvement of 0.17 points across the five product categories. Lastly, Task 3 (Table 4 - (c)) examines the final results of the topic-aware pain points by extracting pain points for each topic within the product categories. This assessment considers the relevance of the final pain points for each topic, and Painsight records the highest performance across the five product categories. Our method exhibits an average performance improvement of 0.35 points over LDA and 0.14 points over BERTopic. The results in Table 4 demonstrate the effectiveness of ITM, which could not be solely assessed using the NPMI metrics in Table 3. By employing Tasks 1, 2, and 3, the Painsight pipeline, which extracts pain points based on relevant tasks, also records higher performance compared to the baseline in human evaluation results. This validates the appropriateness of Painsight as an automatic framework for pain point detection.

6 Conclusion

In this study, we propose Painsight, a novel framework for automatically extracting and evaluating pain points from customer reviews. We address the under-explored problem of pain point detection and present a practical pipeline for real-world scenarios. By employing sentiment analysis and topic modeling, we identify sentiment-aware and topic-aware pain points that reflect customer perceptions and various types of discomfort. The final output is obtained by extracting the most important words or features from the data using a gradient-based attribution score. This score enables us to determine which words or features are most critical in influencing the model's decision-making process and utilize this information in the post-process to recognize more accurate and meaningful pain points. Experimental results demonstrate that Painsight outperforms existing models on five product group reviews, with human evaluation results indicating a high level of agreement compared to the baseline. Future work could involve incorporating diverse customer feedback and constructing a high-quality benchmark dataset to further validate and enhance the proposed approach.

Limitations

In prior research (Salminen et al., 2022), several challenges have been identified in this field, such as noisy or low-quality data, semantic ambiguity, absence of standards, social desirability bias, and the requirement for human intervention. Our study aimed to tackle the challenge of detecting pain points and devised various strategies for managing noisy real-world reviews. Nonetheless, to fully unlock the potential of the Painsight, additional research is necessary to explore the wide range of emotional polarities beyond the generic 'negative' sentiment. Furthermore, customer reviews often show mixed sentiments, which calls for addressing semantic ambiguity. Lastly, the performance of Painsight assessment was constrained to five product categories, highlighting the need for a comprehensive, high-quality benchmark encompassing diverse domains and performance evaluations across distinct categories.

Ethics Statement

Throughout our human evaluation, we collected demographic details such as name, age, gender, and highest education level, after securing participants' consent and assuring them that their information would be exclusively utilized for research purposes. The results from the human evaluation were anonymized to protect participant confidentiality. The authors meticulously examined all customer reviews employed in the assessment, verifying the absence of any offensive or biased material. Participants took part in the evaluation for an estimated 40 minutes. They were compensated with a 5,000 KRW (equivalent to 3.7 USD) gift card, which was marginally above the Korean minimum wage during that period.

Acknowledgements

We express our gratitude to all DSBA Lab members for their support. We also thank Keonwoo Kim, Joonwon Jang, Minjin Jeon, and Hyowon Cho for their insightful discussions and valuable feedback that greatly contributed to the refinement of this paper

References

Armin Alibasic and Tomo Popovic. 2021. Applying natural language processing to analyze customer satisfaction. In *2021 25th International Conference on Information Technology (IT)*, pages 1–4. IEEE.

Jonah Berger, Ashlee Humphreys, Stephan Ludwig, Wendy W Moe, Oded Netzer, and David A Schweidel. 2020. Uniting the tribes: Using text for marketing insight. *Journal of marketing*, 84(1):1–25.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Jiahao Bu, Lei Ren, Shuang Zheng, Yang Yang, Jingang Wang, Fuzheng Zhang, and Wei Wu. 2021. ASAP: A Chinese review dataset towards aspect category sentiment analysis and rating prediction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2069–2079, Online. Association for Computational Linguistics.

Li-Chen Cheng, Kuanchin Chen, Ming-Chu Lee, and Kua-Mai Li. 2021. User-defined swot analysis–a change mining perspective on user-generated content. *Information Processing & Management*, 58(5):102613.

Austina De Bonte and Drew Fletcher. 2014. *Scenario-Focused Engineering: A toolbox for innovation and customer-centricity*. Microsoft Press.

Ellen De Geyndt, Orphee De Clercq, Cynthia Van Hee, Els Lefever, Pranaydeep Singh, Olivier Parent, and Veronique Hoste. 2022. SentEMO: A multilingual adaptive platform for aspect-based sentiment and emotion analysis. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 51–61, Dublin, Ireland. Association for Computational Linguistics.

Chrysanthos Dellarocas, Xiaoquan Zhang, and Neveen F Awad. 2007. Exploring the value of online product reviews in forecasting sales: The case of motion pictures. *Journal of Interactive marketing*, 21(4):23–45.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Robert East, Kathy Hammond, and Wendy Lomax. 2008. Measuring the impact of positive and negative word of mouth on brand purchase probability. *International journal of research in marketing*, 25(3):215–224.

Seyed Pouyan Eslami and Maryam Ghasemaghaei. 2018. Effects of online review positiveness and review score inconsistency on sales: A comparison by product involvement. *Journal of Retailing and Consumer Services*, 45:74–80.

Shi Feng, Eric Wallace, Alvin Grissom II, Mohit Iyyer, Pedro Rodriguez, and Jordan Boyd-Graber. 2018. Pathologies of neural models make interpretations difficult. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3719–3728, Brussels, Belgium. Association for Computational Linguistics.

Chris Forman, Anindya Ghose, and Batia Wiesenfeld. 2008. Examining the relationship between reviews and sales: The role of reviewer identity disclosure in electronic markets. *Information systems research*, 19(3):291–313.

M Geetha, Pratap Singha, and Sumedha Sinha. 2017. Relationship between customer sentiment and online customer ratings for hotels-an empirical analysis. *Tourism Management*, 61:43–54.

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.

Yan Heng, Zhifeng Gao, Yuan Jiang, and Xuqi Chen. 2018. Exploring hidden factors behind online food shopping from amazon reviews: A topic mining approach. *Journal of Retailing and Consumer Services*, 42:161–168.

Hoon Heo, Hyunwoong Ko, Soohwan Kim, Gunsoo Han, Jiwoo Park, and Kyubyong Park. 2021. Pororo: Platform of neural models for natural language processing. `https://github.com/kakaobrain/pororo`.

Nga N Ho-Dac, Stephen J Carson, and William L Moore. 2013. The effects of positive and negative online customer reviews: do brand strength and category maturity matter? *Journal of marketing*, 77(6):37–53.

Christian Homburg and Andreas Fürst. 2007. See no evil, hear no evil, speak no evil: a study of defensive organizational behavior towards customer complaints. *Journal of the Academy of Marketing Science*, 35:523–536.

Jimin Hong, TaeHee Kim, Hyesu Lim, and Jaegul Choo. 2021. AVocaDo: Strategy for adapting vocabulary to downstream domain. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4692–4700, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Ashlee Humphreys and Rebecca Jen-Hui Wang. 2018. Automated text analysis for consumer research. *Journal of Consumer Research*, 44(6):1274–1306.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Ayal Klein, Oren Pereg, Daniel Korat, Vasudev Lal, Moshe Wasserblat, and Ido Dagan. 2022. Opinion-based relational pivoting for cross-domain aspect term extraction. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 104–112, Dublin, Ireland. Association for Computational Linguistics.

Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. 2020. Captum: A unified and generic model interpretability library for pytorch.

Kee-Young Kwahk and Byoungsoo Kim. 2017. Effects of social media on consumers' purchase decisions: evidence from taobao. *Service Business*, 11:803–829.

Sampson Lee. 2014. Pig (pain is good) strategy: Make customer centricity obsolete and start a resource revolution. imatchpoint limited. *Hong Kong*.

Daria Plotkina and Andreas Munzel. 2016. Delight the experts, but never dissatisfy your customers! a multi-category study on the effects of online review source on intention to buy a new product. *Journal of Retailing and Consumer Services*, 29:1–11.

Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*.

Alex Rawson, Ewan Duncan, and Conor Jones. 2013. The truth about customer experience. *Harvard business review*, 91(9):90–98.

Andrew Slavin Ross, Michael C. Hughes, and Finale Doshi-Velez. 2017. Right for the right reasons: Training differentiable models by constraining their explanations. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 2662–2670.

Joni Salminen, Mekhail Mustak, Juan Corporan, Soongyo Jung, and Bernard J Jansen. 2022. Detecting pain points from user-generated social media posts using machine learning. *Journal of Interactive Marketing*, 57(3):517–539.

Jose Ramon Saura, Domingo Ribeiro-Soriano, and Daniel Palacios-Marqués. 2021. From user-generated data to data-driven innovation: A research agenda to understand user privacy in digital markets. *International Journal of Information Management*, 60:102331.

Keith Stevens, Philip Kegelmeyer, David Andrzejewski, and David Buttler. 2012. Exploring topic coherence over many models and many topics. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 952–961, Jeju Island, Korea. Association for Computational Linguistics.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR.

Binda Wang, Yunwen Miao, Hongya Zhao, Jian Jin, and Yizeng Chen. 2016. A biclustering-based method for market segmentation using customer pain points. *Engineering Applications of Artificial Intelligence*, 47:101–109.

Junlin Wang, Jens Tuyls, Eric Wallace, and Sameer Singh. 2020. Gradient-based analysis of NLP models is manipulable. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 247–258, Online. Association for Computational Linguistics.

Yuren Wang, Xin Lu, and Yuejin Tan. 2018. Impact of product attributes on customer satisfaction: An analysis of online reviews for washing machines. *Electronic Commerce Research and Applications*, 29:1–11.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Jia-Jhou Wu and Sue-Ting Chang. 2020. Exploring customer sentiment regarding online retail services: a topic-based approach. *Journal of Retailing and Consumer Services*, 55:102145.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489.

Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. 2021. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems*, 34:18408–18419.

Wenxuan Zhang, Xin Li, Yang Deng, Lidong Bing, and Wai Lam. 2022. A survey on aspect-based sentiment analysis: tasks, methods, and challenges. *IEEE Transactions on Knowledge and Data Engineering*.

Feng Zhu and Xiaoquan Zhang. 2010. Impact of online consumer reviews on sales: The moderating role of product and consumer characteristics. *Journal of marketing*, 74(2):133–148.

A Data Distribution

The distribution of site types used for collecting review data for five product categories (Dryer, Styler, Washing Machine, Vacuum Cleaner, and Robotic Vacuum) is presented in Figure 5. The data collection involved a diverse range of site types across product categories, with the highest number of reviews collected from cafe and community sites. The sentiment-aware pain point detection experiments utilized train, valid, and test datasets in an 8:1:1 ratio. However, the topic modeling in topic-aware pain point detection did not involve splitting the dataset.

B Training Details

The training process of Painsight was conducted using NVIDIA RTX 2080 Ti (for sentiment analysis and topic modeling) and RTX A6000 (for gradient-based attribution score and ITM). The PyTorch [9] library was performed for model training, and post-processing was carried out using Pororo [10] and Captum [11]. The hyperparameters used in Painsight are described in detail below:

---

[9] `https://pytorch.org/`

[10] `https://github.com/kakaobrain/pororo`

[11] `https://captum.ai/`

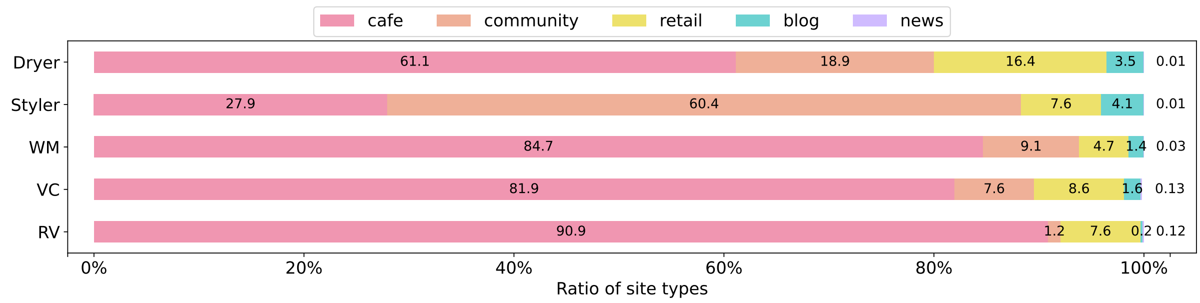


Figure 5: Site type distribution of review data by product category.

B.1 BERT + Vocab Expansion

The model was trained using the transformers library (Wolf et al., 2019) with a BERT classifier. The batch size was set to 32, and the optimizer used was AdamW with a learning rate of 2e-05. The model was trained for 10 epochs, with 62 new vocabularies added. The maximum length of input sequences was set to 128.

B.2 ITM

The initial topic labels were generated using BERTopic, and the transformers library with a BERT classifier was used for classification. The optimizer used was Adam, and the model was trained for 1 epoch with a validation iteration of 10 and a maximum iteration of 100. Early stopping was used with patience 2. The maximum length of input sequences was set to 128, and the batch size was 64 with a learning rate of 3e-5.

C Evaluation Metric

We employed Normalized Pointwise Mutual Information (NPMI) (Stevens et al., 2012) as an evaluation metric to measure the performance of ITM. NPMI is a widely used measure of the correlation between two words, which is computed by normalizing Pointwise Mutual Information (PMI). PMI measures the probability of two words occurring together, taking into account the frequency of their individual occurrences. However, PMI tends to overestimate the importance of infrequent words. To address this issue, NPMI normalizes PMI by considering the probability of the respective words. Through this normalization process, NPMI can more accurately measure the correlation between two words and ranges between -1 and 1. NPMI is often used in topic modeling and is computed using the following formula:

$$\mathrm{PMI}(w_i, w_j) = \frac{P(w_i, w_j)}{P(w_i)P(w_j)}, \tag{7}$$

$$\mathrm{NPMI}(w_i, w_j) = \frac{\mathrm{PMI}(w_i, w_j)}{-\log(P(w_i, w_j))}. \tag{8}$$

Here, $P(w_i, w_j)$ denotes the probability of words w_i and w_j co-occurring, while $P(w_i)$ and $P(w_j)$ represent their individual probabilities. The numerator normalizes the probability of the two words occurring together by dividing it by the product of their individual occurrence probabilities. The denominator uses the log value of their co-occurrence probability to obtain PMI. Using NPMI, we can extract sets of related words in topic classes classified through ITM and evaluate if each topic has coherence. Therefore, we evaluated the consistency of each topic in LDA, BERTopic, and ITM with a set of related reviews using NPMI.

D Human Evaluation

The evaluation instructions provided to annotators for each task in human evaluation are as follows, and the example is shown for the dryer.

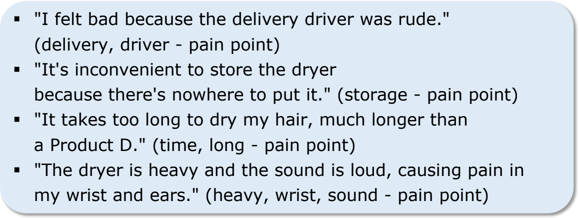
D.1 Task 1: Sentiment-aware pain point evaluation (Pain point candidates)

Each sheet contains three items for each product group:

- Consumer reviews of appliances for each product group collected online
- Keywords extracted for each review (2-4 per review)

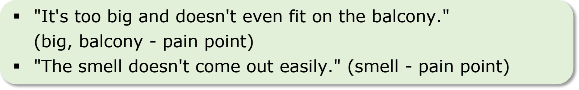
Please rate how well the keywords were extracted from each customer review:

- 1: All keywords were extracted incorrectly.
- 2: Keywords were generally not extracted.



- "I felt bad because the delivery driver was rude." (delivery, driver - pain point)
- "It's inconvenient to store the dryer because there's nowhere to put it." (storage - pain point)
- "It takes too long to dry my hair, much longer than a Product D." (time, long - pain point)
- "The dryer is heavy and the sound is loud, causing pain in my wrist and ears." (heavy, wrist, sound - pain point)

(a) Hair Dryer



- "It's too big and doesn't even fit on the balcony." (big, balcony - pain point)
- "The smell doesn't come out easily." (smell - pain point)

(b) Refrigerator

Figure 6: Examples of Pain Points: We explained the concept of pain points using examples of home appliances that were not used in the evaluation, specifically hair dryers and refrigerators.

- 3: Keywords were extracted at an average level (50% of all keywords).
- 4: Keywords were generally well extracted.
- 5: All keywords were extracted well.

D.2 Task 2: Sentiment-aware pain point evaluation (Final pain points)

Evaluate the suitability of pain points identified as frequently appearing in negative reviews of the product category after reviewing 100 example sentences. To clarify the concept of "pain point," we provide example sentences (Figure 6):

Scoring criteria:

- 0: This pain point cannot be considered a pain point for the product category.
- 1: This pain point can be considered a pain point for the product category.

Notes:

- Pain points may be composed of morphemes or spaced units rather than conventional word structures, selected based on their meaning.
- There may be cases where a pain point cannot be extracted according to the logic. In such cases, you can give a score of 1.

D.3 Task 3: Topic-aware pain point evaluation

The following example contains three elements:

- Consumer reviews on dryers collected from online community
- Review examples for each topic class
- Key pain points for each topic class

20 reviews are provided for each topic class. Please assess the reviews in detail and evaluate whether the pain point can be used as a representative pain point for the topic class, using the following scores:

- 0: The pain point cannot represent the topic class.
- 1: The pain point can represent the topic class.

Notes:

- Pain points can be constructed at the morpheme level, rather than a common word structure, depending on their meaning.
- Do not consider overlaps in meaning or form between pain points. Please only judge the representativeness of each pain point for the reviews on the topic.
- If the topic modeling performance is low, the topics of each review may not match. If the meaning of the topic cannot be identified through the sampled sentences for each topic class, you can assign 0 points to all the pain points.
- Each topic often includes more than 500-1,000 sentences. Some pain points may be not contained in the sampled review. If similar words to that pain points, however, they could be suitable pain points for its cluster. ([ex] "I tried it myself and my wrist hurts", pain point (‘arms’: considered correct))

Context-Dependent Embedding Utterance Representations for Emotion Recognition in Conversations

Patrícia Pereira[1,2], **Helena Moniz**[1,3], **Isabel Dias**[1,2] and **Joao Paulo Carvalho**[1,2]
[1]INESC-ID, Lisbon
[2]Instituto Superior Técnico, University of Lisbon
[3]Faculdade de Letras, University of Lisbon
`{patriciaspereira, isabel.h.dias}@tecnico.ulisboa.pt`
`{helena.moniz, joao.carvalho}@inesc-id.pt`

Abstract

Emotion Recognition in Conversations (ERC) has been gaining increasing importance as conversational agents become more and more common. Recognizing emotions is key for effective communication, being a crucial component in the development of effective and empathetic conversational agents. Knowledge and understanding of the conversational context are extremely valuable for identifying the emotions of the interlocutor. We thus approach Emotion Recognition in Conversations leveraging the conversational context, i.e., taking into attention previous conversational turns. The usual approach to model the conversational context has been to produce context-independent representations of each utterance and subsequently perform contextual modeling of these. Here we propose context-dependent embedding representations of each utterance by leveraging the contextual representational power of pre-trained transformer language models. In our approach, we feed the conversational context appended to the utterance to be classified as input to the RoBERTa encoder, to which we append a simple classification module, thus discarding the need to deal with context after obtaining the embeddings since these constitute already an efficient representation of such context. We also investigate how the number of introduced conversational turns influences our model performance. The effectiveness of our approach is validated on the open-domain DailyDialog dataset and on the task-oriented EmoWOZ dataset.

1 Introduction

Emotion Recognition in Conversations (ERC) is useful in automatic opinion mining, emotion-aware conversational agents and assisting modules for therapeutic practices. There is thus an increasing interest in endowing machines with efficient emotion recognition modules.

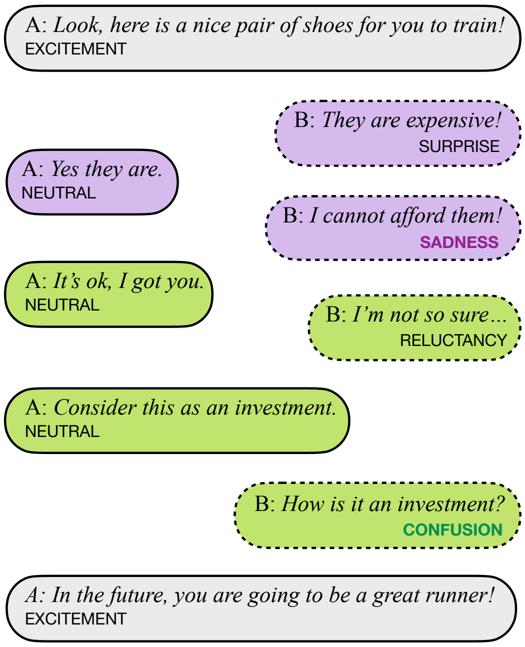

A: *Look, here is a nice pair of shoes for you to train!*
EXCITEMENT
B: *They are expensive!*
SURPRISE
A: *Yes they are.*
NEUTRAL
B: *I cannot afford them!*
SADNESS
A: *It's ok, I got you.*
NEUTRAL
B: *I'm not so sure…*
RELUCTANCY
A: *Consider this as an investment.*
NEUTRAL
B: *How is it an investment?*
CONFUSION
A: In the future, you are going to be a great runner!
EXCITEMENT

Figure 1: A dialogue in which context is key to infer the associated emotions. To infer the emotions *Sadness* and *Confusion*, knowledge of the present and previous two and three utterances is required, respectively.

Knowledge and understanding of the conversational context, i.e., of the previous conversation turns, are extremely valuable in identifying the emotions of the interlocutors (Poria et al., 2019) (Chatterjee et al., 2019) (Pereira et al., 2022).

Research in automatic emotion recognition using machine learning techniques dates back to the end of the 20th century. However, the use of the conversational context as an auxiliary information for the classifiers, did not appear until publicly available conversational datasets became more common.

State-of-the-art ERC works leverage not only state-of-the-art pre-trained-language models such as BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019), but also deep, complex architectures to model several factors that influence the emotions in the conversation (Pereira et al., 2022). Such fac-

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 228–236
July 14, 2023 ©2023 Association for Computational Linguistics

tors usually pertain to self and inter-speaker emotional influence and the context and emotion of preceeding utterances.

In this paper we argue that the powerful representation capabilities of pre-trained language models can be leveraged to model context without the need of additional elaborate classifier architectures, allowing for much simpler and efficient architectures. Furthermore, it is our contention that the Transformer, the backbone of our chosen language model, is better at preserving the contextual information since it has a shorter path of information flow than the RNNs typically used for context modelling. In this line, we rely on the RoBERTa language model and resort to a simple classification module to preserve the contextual information.

The usual approach to model the conversational context has been to produce context independent representations of each utterance and subsequently perform contextual modeling of those representations. State-of-the art approaches start by resorting to embedding representations from language models and employ gated or graph neural network architectures to perform contextual modelling of these embedding representations at a later step. In our much simpler and efficient proposed approach, we produce context-dependent embedding representations of each utterance, by feeding not only the utterance but also its conversational context to the language model. We thus discard the need to deal with context after obtaining the embeddings since these constitute already an efficient representation of such context.

Our experiments show that by leveraging context in this way, one can obtain state-of-the-art results with RoBERTa and a simple classification module, surpassing more complex state-of-the-art models.

2 Related Work

Amongst the first works considering contextual interdependences among utterances is the one by Poria et al. (Poria et al., 2017). It uses LSTMs to extract contextual features from the utterances. These gated recurrent networks make it possible to share information between consecutive utterances while preserving its order.

A more elaborate model also leveraging gated recurrent networks is DialogueRNN (Majumder et al., 2019), which uses GRUs to model the speaker, context and emotion of preceding utterances by keeping a party state and a global state that are used to model the final emotion representation.

Gated recurrent networks have a long path of information flow which makes it difficult to capture long term dependencies. These can be better captured with the Transformer which a has shorter path of information flow. Its invention in 2017 (Vaswani et al., 2017) led to a new state-of-the-art in several Natural Language Processing tasks.

Amongst the first works leveraging the Transformer is the Knowledge-Enriched Transformer (KET) (Zhong et al., 2019). It uses its self-attention to model context and response. It also makes use of an external knowledge base, a graph of concepts that is retrieved for each word.

Following the invention of Transformers, pre-trained language models brought about another new state-of-the art in 2019. Since their invention, most state-of-the art ERC works resorted to encoder pre-trained language models (Shen et al., 2021a) (Ghosal et al., 2020) (Li et al., 2021).

COSMIC (Ghosal et al., 2020) leverages RoBERTa Large as feature extractor. Furthermore, it makes use of the commonsense transformer model COMET (Bosselut et al., 2019) in order to extract commonsense features. Five bi-directional GRUs model a context state, internal state, external state, intent state, and emotion state that influence the final emotion classification.

Psychological (Li et al., 2021) also uses RoBERTa Large for utterance encoding and COMET. For conversation-level encoding it constructs a graph of utterances to model the actions and intentions of the speaker along with the interactions with other utterances. It uses COMET to introduce commonsense knowledge into the graph edge representations and processes this graph using a graph transformer network.

3 Methodology

We describe how we obtain a contextual embedding representation of the sentence and its context with RoBERTa, how we pool the contextual embeddings, our classification module and how we obtain the emotion labels. These processes can be observed in Figure 2.

3.1 Task definition

Given a conversation, a sequence of u_i utterances with corresponding $emotion_i$ from a predefined set of emotions, the aim of the task of ERC is to correctly assign an emotion to each utterance of the

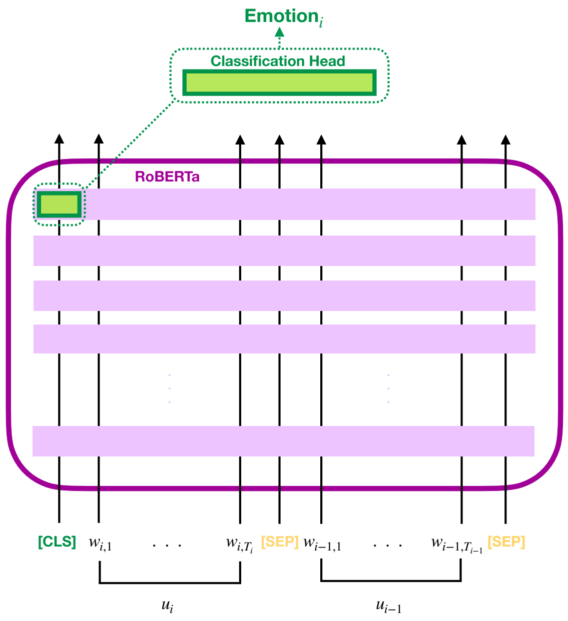


Figure 2: Model architecture. Two utterances are given as input to RoBERTa encoder, of which the CLS token of the last layer is fed to the classification head that predicts the emotion.

conversation. An utterance consists in a sequence of w_{it} tokens representing its T_i words

$$u_i = (w_{i1}, w_{i2}, ..., w_{iT_i}) \tag{1}$$

The usual approach for this task has been to produce context independent representations of each utterance and perform contextual modeling of these. In our approach we produce context-dependent representations of each utterance that represent not only the utterance but also a given number of previous utterances from the conversation.

3.2 Context-dependent feature extraction

For context-dependent feature extraction, we feed as input to RoBERTa the utterance we intend to classify, u_i, concatenated with its conversational context corresponding to the number c of previous utterances in the conversation, $(u_{i-1}, u_{i-2}, ..., u_{i-c})$. Concretely, we feed u_i to the model, preceded by the [CLS] token and suceded by the [SEP] token, followed by the previous turns u_{i-1} up to u_{i-c}, separated by the [SEP] token.

3.3 Pooling

The RoBERTa encoder outputs several layers of embeddings representing the utterance, and in our approach, also the preceding utterances it receives as input. Each layer comprises several tokens, being the number of tokens the same as the number of input tokens. Each token is a vector with dimension corresponding to the RoBERTa hidden size.

From these embeddings one can extract a suitable representation for the sentence. Choosing all tokens from all layers would yield an extremely memory demanding classification layer and may not yield the best model performance. Thus we choose the first embedding from the last layer L, the [CLS] which is used for classification, as in Equation 2.

$$pooled_i = RoBERTa_{L,[CLS]}(input_i) \tag{2}$$

3.4 Emotion Classification

The classification module that follows RoBERTa is a linear fully connected layer, applying a linear transformation to the pooled encoder output data. Its input size is the RoBERTa encoder hidden size and its output size is the number of emotion classes.

The final label probability distribution is yielded by applying the softmax operation to the output of the classification head and the predicted label is the one with the highest probability:

$$emotion_i = argmax(Softmax(pooled_i W^T + b)) \tag{3}$$

4 Experimental Setup

4.1 Training

Our model is based on RoBERTa-base from the Transformers library by Hugging Face (Wolf et al., 2020). It is trained with the cross-entropy loss with logits. The Adam (Kingma and Ba, 2014) optimizer is used with an initial learning rate of 1e-5 and 5e-5, for the encoder and the classification head, respectively with a layer-wise decay rate of 0.95 after each training epoch for the encoder. The encoder is frozen for the first epoch. The batch size is set to 4. Gradient clipping is set to 1.0. As stopping criteria, early stopping is used to terminate training if there is no improvement after 5 consecutive epochs on the validation set over macro-F1, for a maximum of 10 epochs. The checkpoint used in testing is the one that achieves the highest macro-F1 score on the validation set.

4.2 Evaluation

We evaluate the performance of our model with the macro F1-score. The reported results are yielded from an average of 5 runs corresponding to 5 distinct random seeds that are kept for a meaningful comparison of all experiments. This average is motivated by the fact that results for the same experiment obtained with different random seeds can have a variability of about 3 in macro F1-score which is a large deviation given that our proposed approach yields an improvement of that magnitude and comparison between state-of-the-art models are based on improvements of less than 1 F1-score. This procedure is in line with several authors that also resort to 5 run averages (Li et al., 2021) (Zhong et al., 2019) (Shen et al., 2021a) (Shen et al., 2021b).

Our code is publicly available[1].

4.3 Datasets

We evaluate our approach on the chit-chat DailyDialog (Li et al., 2017) dataset and on the task-oriented EmoWOZ (Feng et al., 2022) dataset.

4.3.1 DailyDialog

DailyDialog is built from websites used to practice English dialogue in daily life. It is labelled with the six Ekman's basic emotions (Ekman, 1999), anger, disgust, fear, happiness, sadness and surprise, or neutral. The publicly available splits of Yanran are used.

4.3.2 EmoWOZ

EmoWOZ is derived from MultiWOZ (Budzianowski et al., 2018), one of the largest multi-domain corpora benchmark dataset for various dialogue tasks. User utterances are annotated with either fear, dissatisfaction, apologetic, abusive, excited, satisfied or neutral emotions.

The statistics and proportion of labels in the datasets are presented in Tables 1 and 2, respectively.

Table 1: Statistics of the datasets

	DailyDialog	**EmoWOZ**
Dlg type	Chit-chat	Task-oriented
# Dlgs	13,118	11,434
# Turns	102,979	167,234
Avg turns in dlg	7.9	14.6

Table 2: Proportion of labels in the datasets

DailyDialog			
Ang	Disg	Fear	Hap
1.0%	0.3%	0.2%	12.5%
	Sad	Sur	Neu
	1.1%	1.8%	83.1%
EmoWOZ			
Fear	Diss	Apol	Abus
0.5%	6.1%	1.0%	0.2%
	Exc	Sat	Neu
	1.2%	21.0%	70.1%

From Table 1 it can be noted that EmoWOZ has almost double the amount of average turns per dialogue than DailyDialog.

From Table 2 it can be observed that both datasets are imbalanced, not only for its dominant majority neutral class, but also for the relative imbalance between minority classes. Therefore, we have opted to use the macro-F1 score for evaluation in order to promote consistent performance across all classes.

5 Results and Analysis

5.1 Iterating towards the ideal approach

We have performed extensive experiments in order to obtain our ideal model architecture. From experimenting different approaches to pool the various layers of embeddings RoBERTa provides to choosing which classification module to employ withing a wide variety of deep learning architectures, we put forward our experiments in this subsection.

5.1.1 Fine-tuning

Fine-tuning, the modification of the pre-trained RoBERTa's weights along with the classification head during training with the target dataset, is a determinant procedure for the success of our approach.

In our experiments we observed that if we did not fine-tune the language model and just trained the classification head, the model would always predict the majority neutral class. This supports the notion that pre-trained-language models are useful for a wide variety of tasks but need to be fine-tuned for the specific task at hand.

5.1.2 Pooling

We have performed experiments with several pooling alternatives. From average pooling, max pool-

---

[1] `http://github.com/patricia-pereira/cd-erc`

ing, concatenation of the CLS token of more than 1 last layers to the concatenation of the CLS token with the result from average pooling. All these pooling alternatives resulted in lower performance than choosing the CLS token of the last layer. This might suggest a high representative power for the CLS token, which is proposed for classification, and discards the need for directly considering other tokens for this task.

5.1.3 Classification module

We have also performed alternative experiments with other classification modules than our simple classification head. These consisted in passing the pooled embeddings through Recurrent Neural Networks (Elman, 1991), uni (Hochreiter and Schmidhuber, 1997) and bi-directional (Graves et al., 2005) Long Short-Term Memory Networks and a Conditional Random Field (Lafferty et al., 2001) before feeding them to the classification head. Performance was lower in all alternative experiments when compared to our main approach of using a simple classification head. These results may indicate that our approach leveraging RoBERTa's representational power for context suffices and there is no apparent need for modelling the context with complex classification modules, after obtaining our context-dependent embedding utterance representations.

5.2 Overall Performance

For each of the datasets, we have performed experiments without introducing any context ($c = 0$) to introducing 4 previous conversation turns ($c = 4$), for which the overal performance operationalized by the macro-F1 metric is reported in Table 3. Our results are an average of 5 runs.

Table 3: Model performance in macro F1-score with the introduction of c conversational turns

	DailyDialog	**EmoWOZ**
	macro-F1	macro-F1
c=0	48.52	58.66
c=1	50.31	62.32
c=2	50.44	64.98
c=3	**51.23**	**65.33**
c=4	50.46	63.28

It can be observed that introducing previous conversational context turns leads to an increase in macro-F1 score. As hypothesised, providing no context is never the best option. This shows that the introduction of an adequate number of context turns directly as the language model input significantly improves model performance. In general performance increases with the introduction of each additional context turn up to the ideal number of turns and then it decreases. Overall, it can be concluded that the ideal number of introduced context turns for ERC in both datasets is 3.

5.3 Performance on each emotion label

For each dataset, we report the results on each individual emotion label and also present the confusion matrices for the best determined c value. Our results are an average of 5 runs.

The individual emotion label F1-scores for the DailyDialog dataset are presented in Table 4.

It can be observed that for more than half of the labels, Anger, Fear, Sadness and Neutral, the ideal context to be provided is 3 turns which maximise their F1-scores, and also the macro-F1 score on Table 3, and for the other labels the ideal context is 4 turns for Disgust, 2 turns for Happiness and 1 turn for Surprise. As expected, providing no context is never the best option.

The confusion matrix for $c = 3$ corresponding to the highest macro-F1 score is displayed on Figure 3, in which the label nomenclature and order is the same as in table 4 but with neutral as the first label.

This matrix indicates that majority of the errors are due to classifying utterances as neutral instead of assigning a non-neutral emotion. The classifier also displays some confusion in discerning between Happiness and Surprised.

The individual emotion label F1-scores for the EmoWOZ dataset are presented in Table 5.

It can be observed that for 4 of the labels, Dissatistfied, Excited, Satisfied and Neutral, the ideal context to be provided is 4 turns which maximise their F1-scores. Regarding the other labels the ideal context is 2 for Fear, 3 for Abusive, and surprisingly 0 turns for Apologetic, which might indicate that this emotion is very explicit in this dataset.

The confusion matrix for $c = 3$ corresponding to the highest macro-F1 score is displayed on Figure 4, in which the label nomenclature and order is the same as in table 5 but with neutral as the first label.

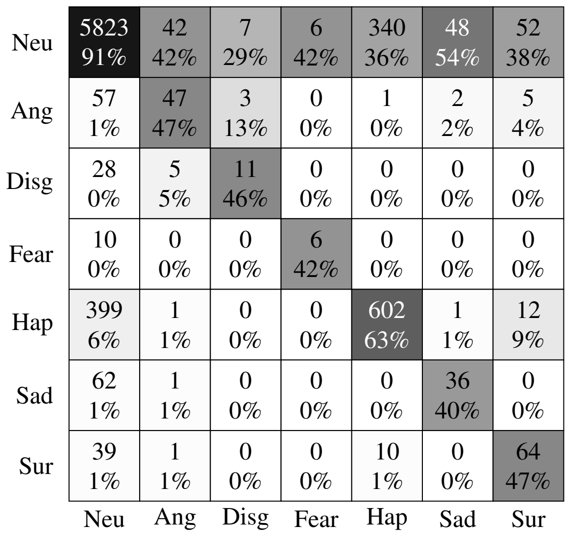
This matrix indicates that majority of the errors are due to classifying utterances as neutral instead of assigning a non-neutral emotion, as in happens with the DailyDialog dataset.

Table 4: Model performance on each individual emotion label on the **DailyDialog** dataset with the introduction of c conversational turns

	Ang	**Disg**	**Fear**	**Hap**	**Sad**	**Sur**	**Neu**
c=0	37.47	32.32	36.69	59.42	33.16	49.60	90.99
c=1	40.18	29.28	39.43	61.26	38.30	**52.66**	91.06
c=2	43.26	33.91	36.52	**61.98**	33.63	52.23	91.12
c=3	**43.51**	33.22	**39.44**	61.12	**38.43**	51.50	**91.42**
c=4	42.00	**34.52**	34.65	61.97	37.18	51.70	91.18

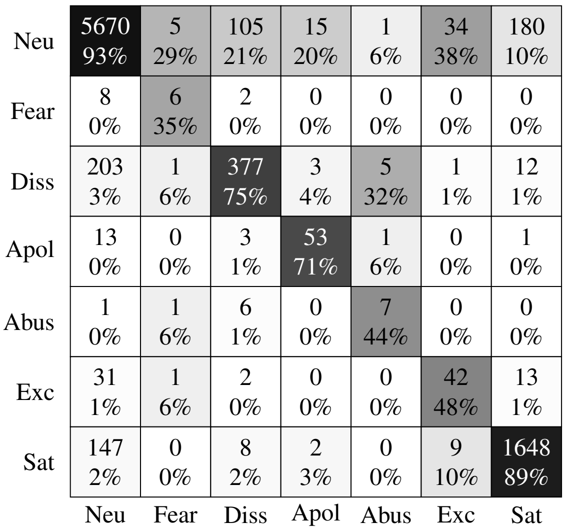
Table 5: Model performance on each individual emotion label on the **EmoWOZ** dataset with the introduction of c conversational turns

	Fear	**Diss**	**Apol**	**Abus**	**Exc**	**Sat**	**Neu**
c=0	35.72	45.18	**74.93**	25.21	46.96	90.09	92.53
c=1	32.97	57.97	72.47	42.97	47.07	89.75	93.01
c=2	**38.91**	66.24	73.37	44.79	48.13	89.73	93.74
c=3	37.89	68.02	72.49	**47.73**	47.64	89.76	93.81
c=4	35.15	**69.57**	73.00	30.09	**50.89**	**90.23**	**94.03**



	Neu	Ang	Disg	Fear	Hap	Sad	Sur
Neu	5823 91%	42 42%	7 29%	6 42%	340 36%	48 54%	52 38%
Ang	57 1%	47 47%	3 13%	0 0%	1 0%	2 2%	5 4%
Disg	28 0%	5 5%	11 46%	0 0%	0 0%	0 0%	0 0%
Fear	10 0%	0 0%	0 0%	6 42%	0 0%	0 0%	0 0%
Hap	399 6%	1 1%	0 0%	0 0%	602 63%	1 1%	12 9%
Sad	62 1%	1 1%	0 0%	0 0%	0 0%	36 40%	0 0%
Sur	39 1%	1 1%	0 0%	0 0%	10 1%	0 0%	64 47%

Figure 3: Confusion Matrix for the **DailyDialog** dataset with the introduction of c=3 conversational turns



	Neu	Fear	Diss	Apol	Abus	Exc	Sat
Neu	5670 93%	5 29%	105 21%	15 20%	1 6%	34 38%	180 10%
Fear	8 0%	6 35%	2 0%	0 0%	0 0%	0 0%	0 0%
Diss	203 3%	1 6%	377 75%	3 4%	5 32%	1 1%	12 1%
Apol	13 0%	0 0%	3 1%	53 71%	1 6%	0 0%	1 0%
Abus	1 0%	1 6%	6 1%	0 0%	7 44%	0 0%	0 0%
Exc	31 1%	1 6%	2 0%	0 0%	0 0%	42 48%	13 1%
Sat	147 2%	0 0%	8 2%	2 3%	0 0%	9 10%	1648 89%

Figure 4: Confusion Matrix for the **EmoWOZ** dataset with the introduction of c=3 conversational turns

It is worth noting that our results are an average of 5 runs and the final model is determined via performance on the validation set. Therefore, the fluctuation in individual label F1-scores does not hinder the representativity of our results and these fluctuations may occur between results from the other reported state-of-the-art models.

5.4 Comparison with state-of-the-art

We further compare our approach to other state-of-the-art approaches that also resort to the RoBERTa or BERT pre-trained-language models. This allows for a fair comparison between approaches given that using this language model brings great performance increases when compared to using other means of utterance feature extraction. Regarding DailyDialog results, we compare our approach to COSMIC (Ghosal et al., 2020), RoBERTa and RoBERTa DialogueRNN, implemented by the authors of COSMIC, and the Psychological model (Li et al., 2021), all models described in Section 2. Concerning the performance on the EmoWOZ dataset, we compare out approach to COSMIC, BERT and BERT DialogueRNN, tested by the authors of EmoWOZ (Feng et al., 2022), since for this dataset the authors obtained a more suitable

Table 6: Comparison with state-of-the-art works

	DailyDialog macro-F1	**EmoWOZ** macro-F1
RoBERTa (Ghosal et al., 2020) / BERT (Feng et al., 2022)	48.20	55.80
RoBERTa (Ghosal et al., 2020) / BERT (Feng et al., 2022) + DlgRNN	49.65	57.10
ContextBERT (Feng et al., 2022)	-	59.70
COSMIC (Ghosal et al., 2020) / (Feng et al., 2022)	51.05	61.12
Psychological (Li et al., 2021)	**51.95**	-
CD-ERC (Ours)	51.23	**65.33**

uterrance representation using BERT instead of RoBERTa. Results are displayed on table 6 and are an average of 5 runs.

Regarding performance on the DailyDailog dataset, our approach outperforms not only the simple RoBERTa/BERT, but also RoBERTa/BERT in a more elaborate gated neural network model such as DialogueRNN and COSMIC. The Psychological model has a slightly higher performance than ours. It may be due to the fact that it leverages a large commonsense knowledge base and an elaborate classifier architecture, while we opted for a minimalistic classification module. Concerning performance on the EmoWOZ dataset, our approach outperforms all baselines by a wide margin, setting a new state of the art for task-oriented emotion datasets.

5.5 Case Studies

On Table 7 we can compare the performance of our contextual classifier when considering the ideal 3 context turns on both datasets versus not considering any context at all.

In the first example, from the DailyDialog dataset, A offers B assistance, so B asks A to view the apartment, to which A sadly apologizes informing B that B will not be able to view it. The classifier that does not consider context classifies this last apology as neutral. However, given the context of the conversation, A should not be neutral since A is unable to assist B which was A's initial purpose. The contextual classifier is able to consider this, thus correctly classifying A's utterance with the emotion Sadness.

In the second example, also from the DailyDialog dataset, A gives B a good idea to which B happily reacts and thanks A. A happily reacts to B's acknowledgments, especially since B mentioned A's was a "wonderful idea". The classifier that does not consider context classifies A's final reaction to B as neutral, since A's utterance is a merely "No problem. Good luck", not being able to recognize A's positive reaction to B's acknowledgements. The contextual classifier, however, having this utterances into account, correctly classifies A's final reaction with the emotion Happiness.

In the last example, from the EmoWOZ dataset, B is merely answering A's question of what day B would like to travel. The classifier that does not consider context takes into account the words "please" and "vacation" which bias the classification towards the emotion Excited. The contextual classifier might grasp that "please" is used as a polite expression and "vacation" is just the object of the phrase, thus correctly classifying the utterance as neutral.

6 Conclusions and Future Work

In this work we have leveraged context-dependent embedding utterrance representations for Emotion Recognition in Conversations. Our approach of producing context-dependent representations of each utterance contrasted with the usual approach of producing context independent representations of each utterance and subsequently performing contextual modeling of these. It consisted in feeding a variable number of previous conversational turns appended to the utterance to be classified as input to the state-of-the-art pre-trained-language model RoBERTa, to which we appended a simple classification module. We further investigated how the number of introduced conversational turns influenced our model performance. We concluded that the introduction of an adequate number of context turns directly as the language model input significantly improves model performance.

Furthermore, we attained state-of-the-art results on the widely used DailyDialog dataset and established a new state-of-the-art by a wide margin on the EmoWOZ dataset, which are usually yielded by

Table 7: Case studies comparing the performance of our contextual classifier ($c = ideal = 3$) with the no-context classifier ($c = 0$)

Turn	Gold	$c = 0$	$c = 3$
A: Can I help you ?	Neu	Neu	Neu
B: I would actually like to view the apartment for rent today .	Neu	Neu	Neu
A: **I ' m sorry , but you won ' t be able to view it today .**	**Sad**	**Neu**	**Sad**
A: Maybe you should look around for an outlet .	Neu	Neu	Neu
B: That is a wonderful idea .	Hap	Hap	Hap
A: Outlets have more reasonable prices .	Neu	Neu	Neu
B: Thank you for your help .	Hap	Hap	Hap
A: **No problem . Good luck**	**Hap**	**Neu**	**Hap**
A: On what day would you like to travel?	-	-	-
B: **Saturday, please. I'm thinking just a short vacation over the weekend.**	**Neu**	**Exc**	**Neu**

more elaborate classifiers resorting to larger state-of-the-art pre-trained-language models and more complex classification modules.

For future work, from adequately capturing the conversation context, the focus of our approach, to capturing several other factors that influence the emotions in the conversation, such as self and inter-speaker emotional influence and the emotion of preceeding utterances, various architectures comprising not only state-of-the art language models for embeddings but also combining our context-dependent embedding utterance representation with more elaborate classification modules can be used.

Finally, we put forward important ethical aspects pertaining to Emotion Recognition in Conversations. These are, for example and not limited to, whether an ERC module should be developed or used for a certain purpose, which data to collect, the subjects behind the data, diversity, inclusiveness, privacy, control and possible biases and misuses of the application (Mohammad, 2022). Research taking into account these aspects will benefit the community with better ERC modules for current and novel applications.

7 Acknowledgements

This work was supported by Fundação para a Ciência e a Tecnologia (FCT), through Portuguese national funds Ref. UIDB/50021/2020, Agência Nacional de Inovação (ANI), through the project CMU-PT MAIA Ref. 045909, RRP and Next Generation EU project Center for Responsible AI Ref. C645008882-00000055, and the COST Action Multi3Generation Ref. CA18231.

References

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. Comet: Commonsense transformers for automatic knowledge graph construction. *arXiv preprint arXiv:1906.05317*.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - a large-scale multi-domain Wizard-of-Oz dataset for task-oriented dialogue modelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026, Brussels, Belgium. Association for Computational Linguistics.

Ankush Chatterjee, Kedhar Nath Narahari, Meghana Joshi, and Puneet Agrawal. 2019. Semeval-2019 task 3: Emocontext contextual emotion detection in text. In *Proceedings of the 13th international workshop on semantic evaluation*, pages 39–48.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Paul Ekman. 1999. Basic emotions. *Handbook of cognition and emotion*, 98(45-60):16.

Jeffrey L Elman. 1991. Distributed representations, simple recurrent networks, and grammatical structure. *Machine learning*, 7(2):195–225.

Shutong Feng, Nurul Lubis, Christian Geishauser, Hsien-chin Lin, Michael Heck, Carel van Niekerk, and Milica Gasic. 2022. Emowoz: A large-scale corpus and labelling scheme for emotion recognition in task-oriented dialogue systems. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4096–4113.

Deepanway Ghosal, Navonil Majumder, Alexander Gelbukh, Rada Mihalcea, and Soujanya Poria. 2020. COSMIC: COmmonSense knowledge for eMotion identification in conversations. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2470–2481, Online. Association for Computational Linguistics.

Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. 2005. Bidirectional lstm networks for improved phoneme classification and recognition. In *International conference on artificial neural networks*, pages 799–804. Springer.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Jiangnan Li, Zheng Lin, Peng Fu, and Weiping Wang. 2021. Past, present, and future: Conversational emotion recognition through structural modeling of psychological knowledge. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1204–1214, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. *arXiv preprint arXiv:1710.03957*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Navonil Majumder, Soujanya Poria, Devamanyu Hazarika, Rada Mihalcea, Alexander Gelbukh, and Erik Cambria. 2019. Dialoguernn: An attentive rnn for emotion detection in conversations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6818–6825.

Saif M Mohammad. 2022. Ethics sheet for automatic emotion recognition and sentiment analysis. *Computational Linguistics*, 48(2):239–278.

Patrícia Pereira, Helena Moniz, and Joao Paulo Carvalho. 2022. Deep emotion recognition in textual conversations: A survey. *arXiv preprint arXiv:2211.09172*.

Soujanya Poria, Erik Cambria, Devamanyu Hazarika, Navonil Majumder, Amir Zadeh, and Louis-Philippe Morency. 2017. Context-dependent sentiment analysis in user-generated videos. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 873–883, Vancouver, Canada. Association for Computational Linguistics.

Soujanya Poria, Navonil Majumder, Rada Mihalcea, and Eduard Hovy. 2019. Emotion recognition in conversation: Research challenges, datasets, and recent advances. *IEEE Access*, 7:100943–100953.

Weizhou Shen, Junqing Chen, Xiaojun Quan, and Zhixian Xie. 2021a. Dialogxl: All-in-one xlnet for multiparty conversation emotion recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13789–13797.

Weizhou Shen, Siyue Wu, Yunyi Yang, and Xiaojun Quan. 2021b. Directed acyclic graph network for conversational emotion recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1551–1560, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Peixiang Zhong, Di Wang, and Chunyan Miao. 2019. Knowledge-enriched transformer for emotion detection in textual conversations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 165–176, Hong Kong, China. Association for Computational Linguistics.

Combining Active Learning and Task Adaptation with BERT for Cost-Effective Annotation of Social Media Datasets

Jens Lemmens
University of Antwerp (CLiPS)
Prinsstraat 13
2000 Antwerp (Belgium)
jens.lemmens@uantwerpen.be

Walter Daelemans
University of Antwerp (CLiPS)
Prinsstraat 13
2000 Antwerp (Belgium)
walter.daelemans@uantwerpen.be

Abstract

Social media provide a rich source of data that can be mined and used for a wide variety of research purposes. However, annotating this data can be expensive, yet necessary for state-of-the-art pre-trained language models to achieve high prediction performance. Therefore, we combine pool-based active learning based on prediction uncertainty (an established method for reducing annotation costs) with unsupervised task adaptation through Masked Language Modeling (MLM). The results on three different datasets (two social media corpora, one benchmark dataset) show that task adaptation significantly improves results and that with only a fraction of the available training data, this approach reaches similar F1-scores as those achieved by an upper-bound baseline model fine-tuned on all training data. We hereby contribute to the scarce corpus of research on active learning with pre-trained language models and propose a cost-efficient annotation sampling and fine-tuning approach that can be applied to a wide variety of tasks and datasets.

1 Introduction

Approximately 59% of the population worldwide use social media (Chaffey, 2023). Collectively, they post more than half a million comments on Facebook each minute, and a grand total of 500 million tweets per day (Shepherd, 2023; Aslam, 2023). These statistics indicate that social media are a virtually inexhaustible source of data, and a large part of this data can be accessed for research purposes. However, annotating this data, which is often necessary to achieve high prediction performance with pre-trained language models, can be an expensive and time-consuming process. One approach that has been proposed in previous research to reduce annotation costs is active learning (AL), which aims at optimizing the annotation effort by selecting specific data points from an unlabeled data pool which are expected to contribute the most to a model's learning phase (Settles, 2009). Although AL has proven its usefulness throughout decades of research, it remains a data selection method, which makes it challenging to use as only tool for annotation cost reduction and to reach upper bound performance (achieved by supervised learning on all available data).

In this work, we therefore exploit the capabilities of transformer-based pre-trained language models to learn from unsupervised data through their pretraining task. Concretely, we combine AL based on the prediction uncertainty of a model with unsupervised task adaptation through masked language modeling (MLM) to investigate whether this combination of techniques allows reaching the upper bound and with how much data. We test this approach by using different AL protocols on three publicly available datasets (2 social media datasets and 1 benchmark) that are costly to annotate, e.g., because they contain many fine-grained labels or the task is difficult to learn with little data. The experiments presented in this work show that using task adaptation before AL has a significant effect on model performance, and that substantially less data is needed to reach upper bound performance, therefore reducing annotation expenses.

2 Related research

2.1 Active learning

Active learning, or "sequential sampling", has been studied since the 1990's (Lewis and Gale, 1994; Lewis and Catlett, 1994; Cohn et al., 1994, 1996). Originally, AL referred to prioritizing certain entries in an unlabeled dataset during the annotation process, based on the prediction confidence of a model trained on a small initial subset of labeled data, as shown in Figure 1. The intuition behind this sampling strategy is that the less certain a model is about a prediction, the more the model can learn from this data point during training, thereby

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 237–250
July 14, 2023 ©2023 Association for Computational Linguistics

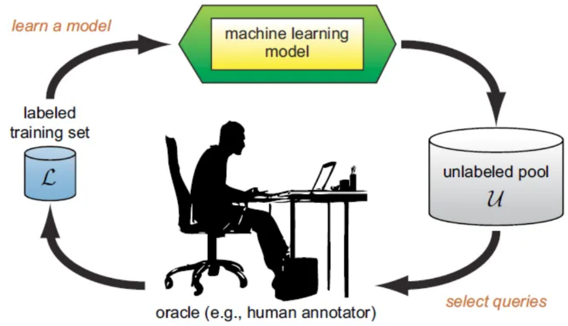


Figure 1: Illustration of a pool-based active learning process (Settles, 2009): A random sample is first labeled by an oracle (human annotator) and used to train a prototype model. This model then predicts the labels of the rest of the unlabeled data. Afterwards, the n data points with the lowest prediction confidence are annotated and used to update the model. This process continues until the annotation budget is depleted or until no more improvements are observed in the learning curve.

maximizing the return on annotation investment.

In the last decades, AL has shown improvements in different tasks and models, such as text classification with k-nearest neighbors (Shi et al., 2008), word sense disambiguation with support vector machines (Zhu and Hovy, 2007), and machine translation with recurrent neural networks (Vashistha et al., 2022). Although AL is most commonly based on prediction uncertainty, sampling can also be based on model disagreement, such as in BALD (Bayesian AL by Disagreement, (Houlsby et al., 2011)), gradient information, such as in BADGE (Batch AL by Diverse Gradient Embeddings, Ash et al. (2019)), typicality or density (Zhu et al., 2008), batch diversity or representativeness (Shi et al., 2021), and other metrics (or a combination of any of the aforementioned, Settles (2009)).

2.2 Active learning with language models

Although AL with pre-trained transformers has gained interest in recent years, the amount of research remains relatively scarce compared to AL with traditional machine learning or neural models. Existing work, e.g. Schröder et al. (2022), has examined the vanilla uncertainty-based query strategy for various binary and multi-class text classification experiments and shown that this strategy is also effective for pre-trained language models such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019). Similarly, Ein-Dor et al. (2020) investigate the effect of annotation sample selection based on prediction uncertainty, expected gradient length and batch representativeness compared to random sample selection. They hereby focus on binary text classification tasks where the positive class is in the minority and show that all strategies perform substantially better than a random sampling strategy.

Recently, Rotman and Reichart (2022) were the first to explore multi-task AL with pre-trained language models, building upon the work of Reichart et al. (2008), who introduced the concept for traditional machine learning methods, and Ikhwantri et al. (2018), who used it for (non-pre-trained) neural architectures. Entropy-based confidence, both in isolation and when using dropout agreement, was used for multi-task AL, and compared to single-task AL and random sampling. Additionally, they investigated the effect of task weighting in ranking samples on informativeness. Their experiments showed that multi-task AL is an efficient way for annotation cost reduction, but that the precise method should depend on the task(s).

Finally, it is noteworthy that any model utilized in an AL setting can be trained in two ways: by updating it after each step, or by re-initializing it and training the entire model on all available annotated data (Schröder et al., 2022; Ein-Dor et al., 2020; Hu et al., 2018; Shen et al., 2017). In general, language models are more frequently re-initialized than updated, because they tend to be unstable when incrementally fine-tuned on low amounts of data, resulting in lower performance and higher standard deviations across different random seeds or hyperparameters (Dodge et al., 2020). However, the effect of re-initializing versus updating language models during AL is understudied. This work will therefore investigate whether re-initializing language models is indeed the preferred approach when using the standard uncertainty-based AL approach, and when combining it with task adaptation.

2.3 Task adaptation

As mentioned, this work combines AL with task adaptation. The latter refers to learning training data in an unsupervised manner before fine-tuning on it for a specific end task. For example, Buhmann et al. (2022) show that task adaptation has a positive effect on their question-answering model used for VaccinChat[1]: a user interface that answers Dutch-language user questions about the COVID-19 vac-

---

[1] `https://vaccinchat.be/`

cine, specifically for Flemish (Dutch speaking Belgian) users. Similarly, Mehri et al. (2020) show that task adaptation through performing MLM on the training data before fine-tuning increases the performance of their task-oriented dialogue system. In the experiments presented in this paper, we adopt the MLM approach as task adaptation step before commencing AL.

3 Methodology

3.1 Data

Existing AL research often relies on datasets that are inexpensive to annotate or tasks that are relatively easily learned by models, even when little data is available. Examples are the IMDB movie review, YELP polarity, SST-2, and TREC datasets for NLP (Maas et al., 2011; Zhang et al., 2015; Socher et al., 2013; Hovy et al., 2001), and the MNIST and Fashion-MNIST for computer vision (Deng, 2012; Xiao et al., 2017). This results in experiments where very small initial training samples are used, which are then increased in equally small steps, while still achieving relatively high prediction performances. Although it is necessary to create artificial AL setups, the aforementioned experimental settings are in our opinion inappropriate for research on AL, because the method is in reality the most effective when working with data that is expensive to annotate or when many examples are needed to gain high prediction accuracy.

Taking this into account, we use the FRENK (Ljubesic et al., 2019), and GoEmotions (Demszky et al., 2020) datasets for our experiments and validate the results on an additional benchmark: the 20 News Groups dataset (Lang, 1995). A detailed description of these datasets and why they are useful for AL experiments can be found below, and a summary of the statistics of each of these datasets can be found in Table 1. A fine-grained overview of the class distributions of the datasets can be found in Appendix A.

Table 1: Statistics of the data used in our experiments.

Dataset	Labels	Train	Val	Test
FRENK	4 (multi-class)	8,404	933	2,301
GoEmotions	28 (multi-label)	43,410	5,426	5,427
20 News Groups	20 (multi-class)	10,182	1,131	7,532

3.1.1 FRENK

The FRENK dataset[2] consists of Facebook comment threads on news item posts about two topics: migrants and the LGBTQ+ community (Ljubesic et al., 2019). Although the dataset contains Slovene and English comments, the current work only utilizes the English partition. The data contains labels concerning the topic (LGBTQ+ or migrants), the target of the hate speech (topic, related to topic, journalist/medium, other, no target), and the type of hate speech, which is the task we tackle in this study. FRENK distinguishes six types of hate speech in its annotation scheme:

1. **Background-violence** consists of messages that call for violence based on the personal background of the target (e.g., religion, gender, race or ethnicity).
2. **Other-violence** contains messages that call for violence for another reason than the background of the target, e.g., an opinion expressed by the target.
3. **Background-offensive** refers to messages that contain offensive statements that are aimed at the background of the target.
4. **Other-offensive** are messages that contain offensive language towards any aspect of the speaker but their personal background.
5. **Inappropriate speech** pertains to messages that contain vulgar and/or other types of offensive language that is not aimed towards a specific target (hence this category is technically not hate speech).
6. **Appropriate speech**, which does not contain any form of offensive or violent language.

Since the violent language classes contain very few entries, and the experiments in this paper require using small samples of training data, labels (1) and (2) were combined to form one "violent language" class, and labels (3) and (4) were used to form an "offensive language" class for the purpose of this paper, resulting in a total of 4 labels (the label distribution with these newly formed classes can be found in Table 15, Appendix A). The training, validation, and test partitions contain 8,404, 933 and 2,301 entries, respectively.

Since hate speech is a term that is open for interpretation and its identification depends on the personal and cultural background of the annotator,

---

[2] `https://huggingface.co/datasets/classla/FRENK-hate-en`

multiple annotators are needed to generate high quality labels and avoid bias (Sap et al., 2022). In addition, the labels in FRENK show strong class imbalances, which is why many comments are needed in order to collect sufficient annotations for the underrepresented classes, resulting in high annotation costs. Further, annotators were required to read the comments thoroughly, since the labels contain a hierarchy of importance in cases where multiple types of hate speech occur in one message (other < background; offensive language < violent language), which increases annotation time even more.

3.1.2 GoEmotions

The GoEmotions dataset[3] contains Reddit comments annotated with 28 emotions (incl. "neutral") in a multi-label setting (Demszky et al., 2020). The dataset is divided in 43,410 cases for training, 5,426 for validating, and 5,427 for testing.

Due to its high number of classes and multi-label scheme, annotating the GoEmotions dataset is labor-intensive. Given its large class imbalances, it is particularly difficult to gain a performance increase in the smaller classes by annotating more samples, since many samples need to be annotated before collecting a substantial amount of messages that express emotions that are infrequent in the dataset.

3.1.3 20 News Groups

The 20 News Groups dataset[4] contains approximately 20,000 news groups posts each associated with 1 out of 20 different topic classes (Lang, 1995). We use this benchmark with many fine-grained classes as an additional test for the proposed approach next to the above mentioned social media datasets, which are the focus of this paper.

3.2 Approaches

3.2.1 Baseline approaches

Random sampling For this baseline, the training data was sampled randomly so that the effect of the AL strategies could be measured.

Upper bound This approach refers to fine-tuning with all available training data in order to estimate the highest possible performance that can be achieved with standard fine-tuning.

3.2.2 AL approaches

Model re-initialization This method refers to the standard AL strategy for language models as proposed in (Schröder et al., 2022; Ein-Dor et al., 2020): An initial sample is used to fine-tune a model, which then predicts labels for the rest of the training data. A second sample is then selected based on the confidence of the model and a new model is initialized and fine-tuned using all annotated data. This process then repeats itself n times.

Checkpoint updating This method is identical to the approach above, with the only difference being that each time a new batch of annotated data is selected, the model is not re-initialized, but fine-tuning continues with the new annotated sample starting from the final checkpoint of the previous round of fine-tuning.

Two-step learning This approach is a specific form of AL where a model is first fine-tuned and then updated once. For the first fine-tuning stage, an initial random sample is used. After predicting the labels of the rest of the training data with this model, it is fine-tuned a second time using the top n most uncertain entries, where n is determined by the rest of the annotation budget. In contrast, "checkpoint updating" refers to annotating various batches of fixed size and updating the model after each batch (i.e., in more than two steps).

3.3 Experimental setup

The annotation process was replicated as follows: assuming that a labeled validation and test set are available, a random batch consisting of ca. 10% of all training data was used as an initial sample for fine-tuning. Then, the amount of training data was incrementally increased until approx. 50% of the available training data was used. For all classification experiments, a BERT-base-uncased model (Devlin et al., 2019) was fine-tuned for 5 epochs with a batch size of 32 and a learning rate of 5e-5. The model was evaluated after each epoch, and predictions on the test set were made with the checkpoint that yielded the highest macro-averaged F1-score on the validation set.

In experiments where task adaptation was used, the model was first fine-tuned on the entire unlabeled training dataset through MLM for the duration of 5 epochs with a batch size of 64 and learning rate of 1e-4. The model checkpoint with lowest validation loss was used for further experiments.

---

[3] `https://huggingface.co/datasets/go_emotions`

[4] `https://huggingface.co/datasets/SetFit/20_newsgroups`

Table 2: Macro-averaged results on **FRENK** using **random sampling**. The last row represents the upper bound baseline.

n	Pre	Rec	F1	Std
1,000	47.5	42.5	43.4	1.5
1,500	47.4	45.4	45.9	0.7
2,000	54.1	46.3	48.0	4.7
2,500	63.1	49.0	52.0	4.8
3,000	65.4	50.8	54.2	3.5
3,500	66.9	49.1	52.9	4.6
4,000	67.2	53.4	56.4	1.7
8,404	62.0	57.8	59.1	1.5

Table 3: Macro-averaged results on **FRENK** using AL with model **re-initialization**. Best F1 are in bold, experiments reaching the upper bound are in grey, and statistical significance is indicated with asterisks.

	Re-initialization				**+ Adaptation**			
n	**Pre**	**Rec**	**F1**	**Std**	**Pre**	**Rec**	**F1**	**Std**
1,000	47.5	42.5	43.4	1.5	50.2	44.4	**45.7****	2.1
1,500	50.2	45.8	46.9	4.6	57.5	46.9	**48.7**	3.8
2,000	63.7	45.8	48.5	1.8	59.8	49.2	**51.8****	3.3
2,500	63.4	50.9	54.5	5.0	65.6	56.0	**59.4***	1.5
3,000	62.8	51.8	55.0	3.1	65.3	56.5	**59.4**	2.1
3,500	63.9	54.1	57.1	2.6	66.2	56.0	**59.3**	2.8
4,000	65.0	53.7	57.4	2.3	63.3	59.6	**61.0****	3.2

Table 4: Macro-averaged results on **FRENK** using AL with model **updating**. Best F1 are in bold, experiments reaching the upper bound are in grey.

	Checkpoint updating				**+ Adaptation**			
n	**Pre**	**Rec**	**F1**	**Std**	**Pre**	**Rec**	**F1**	**Std**
1,000	47.5	42.5	43.4	1.5	50.2	44.4	**45.7**	2.1
1,500	49.6	45.6	**46.8**	4.0	55.7	45.3	46.5	0.9
2,000	51.6	47.7	48.3	3.5	56.9	48.3	**50.7**	5.3
2,500	54.4	49.8	50.7	3.5	65.8	49.0	**53.1**	1.9
3,000	56.7	47.9	49.7	4.3	64.9	51.6	**55.3**	1.1
3,500	57.2	52.4	53.8	4.7	60.8	52.9	**54.8**	5.4
4,000	62.7	53.3	**56.2**	2.8	61.8	52.4	55.2	2.9

Table 5: Macro-averaged results on **FRENK** using AL with **two-step learning**. Best F1 are in bold, experiments reaching the upper bound are in grey.

	Two-step				**+ Adaptation**			
n	**Pre**	**Rec**	**F1**	**Std**	**Pre**	**Rec**	**F1**	**Std**
1,000	47.5	42.5	43.4	1.5	50.2	44.0	**45.7**	2.1
1,500	49.6	45.6	**46.8**	4.0	55.7	45.3	46.5	0.9
2,000	54.7	46.5	48.5	4.3	69.5	51.4	**55.6**	2.7
2,500	55.4	48.0	50.2	5.3	64.1	53.6	**56.7**	2.2
3,000	62.9	48.9	52.4	5.0	65.3	51.7	**55.6**	2.1
3,500	61.9	54.4	55.8	3.0	63.8	56.7	**58.5**	1.7
4,000	64.2	52.1	55.5	4.5	67.3	55.4	**59.3**	3.3

4 Results

4.1 FRENK

Table 2 shows the random and upper bound baselines on the FRENK dataset, whereas Table 3, 4, and 5 show the results achieved for AL with model re-initialization, checkpoint updating, and two-step learning, respectively. The upper bound F1-macro amounts to 59.1%, which was surpassed when using 2,500 training instance in an AL fine-tuning process with model re-initialization after task adaptation. Specifically, using 29.7% of the available labeled training data led to a performance of 59.4% F1-macro. When increasing the training data to 4,000 entries, 47.6% of all available training data, the F1-score reached 61.0%.

Overall, model re-initialization when fine-tuning after querying a new sample of annotated data achieves best results, followed by two-step learning, and finally incremental checkpoint updating. In the case of the latter, we even observe a small decrease in performance (from 55.3% to 54.8%) and high standard deviation of the F1-score (5.4%) when updating the model with 500 unseen entries after it was already fine-tuned cumulatively on 3,000 entries. This shows that updating model checkpoints with smaller samples of data is inefficient, cf. Schröder et al. (2022); Ein-Dor et al. (2020). For the best AL approach, i.e. model re-initialization, a McNemar test was conducted for each sample size to determine whether the improvements after task adaptation are statistically significant (McNemar, 1947), as shown in Table 3. These tests indicate that the improvements in 4 out of 7 experiments are statistically significant, while the other three still show substantial improvements.

To illustrate where most improvements were made after using task adaptation, Table 19 (Appendix C), shows the results per class after fine-tuning on 2,500 training entries using different sampling approaches (random sampling, standard AL, AL after task adaptation). In this table, it can be observed that the highest improvements were made in the most difficult and underrepresented class, namely the "violent speech" class. More precisely, results improved from 18.3% to 32.3% when using standard AL instead of random sampling, and to 47.1% after using task adaptation before AL. Noteworthy is that the performance for the "inappropriate speech" class dropped with 1.4% when using standard AL (compared to random sampling), but when using task adaptation, this performance drop was no longer observed.

Table 6: Macro-averaged results on **GoEmotions** using **random sampling**. The last row represents the upper bound baseline.

n	Pre	Rec	F1	Std
4,000	12.8	8.7	9.7	2.6
6,000	18.2	13.0	14.2	1.4
8,000	31.3	18.7	21.5	0.9
10,000	37.7	22.8	26.3	1.5
12,000	48.9	28.1	32.7	2.5
14,000	51.1	32.5	37.5	1.4
16,000	54.6	34.7	39.7	0.4
18,000	52.2	36.1	40.9	0.4
20,000	52.1	37.1	41.8	0.9
22,000	52.4	38.7	43.1	0.8
43,410	54.4	43.7	47.2	0.7

Table 7: Macro-averaged results on **GoEmotions** using AL with model **re-initialization**. Best F1 are in bold, experiments reaching the upper bound are in grey. See Appendix B, Table 18 for statistical significance per class / sample size.

	Re-initialization				+ Adaptation			
n	Pre	Rec	F1	Std	Pre	Rec	F1	Std
4,000	12.8	8.7	9.7	2.6	15.5	11.9	**12.9**	0.4
6,000	22.0	15.3	17.0	0.9	36.3	21.8	**24.7**	0.5
8,000	36.6	22.6	25.8	1.4	47.6	30.0	**34.3**	0.5
10,000	49.0	29.6	34.2	0.7	51.2	34.2	**39.3**	0.6
12,000	52.0	33.0	38.1	0.9	54.5	36.8	**41.9**	0.8
14,000	54.7	35.5	40.8	0.6	56.9	39.2	**44.2**	0.4
16,000	56.6	38.1	43.5	0.6	55.2	40.5	**45.2**	0.6
18,000	55.5	38.9	43.9	0.7	55.2	40.9	**45.5**	0.1
20,000	55.7	40.3	45.1	0.5	54.6	41.8	**46.2**	0.5
22,000	55.7	41.0	45.6	0.8	54.4	42.5	**46.3**	0.4

Table 8: Macro-averaged results on **GoEmotions** using AL with **checkpoint updating**. Best F1 are in bold, experiments reaching the upper bound are in grey.

	Checkpoint updating				+ Adaptation			
n	Pre	Rec	F1	Std	Pre	Rec	F1	Std
4,000	12.8	8.7	9.7	2.6	14.1	11.6	**12.6**	0.2
6,000	22.4	16.2	17.9	1.7	29.7	20.7	**22.9**	2.1
8,000	33.8	21.6	24.7	1.7	42.4	25.7	**29.1**	3.2
10,000	44.1	26.6	30.5	1.9	50.9	30.3	**34.9**	2.0
12,000	51.7	29.9	34.9	1.8	53.5	35.7	**40.6**	1.5
14,000	56.5	33.8	39.4	1.0	55.9	36.8	**42.0**	0.7
16,000	57.2	34.7	40.6	1.2	55.9	37.7	**42.7**	0.7
18,000	56.2	34.4	41.1	0.7	54.5	39.1	**43.6**	0.4
20,000	56.1	37.6	43.1	0.9	55.0	38.8	**43.6**	0.7
22,000	55.9	37.7	43.0	0.8	54.8	39.3	**43.7**	0.6

Table 9: Macro-averaged results on **GoEmotions** using AL with **two-step learning**. Best F1 are in bold, experiments reaching the upper bound are in grey.

	Two-step				+ Adaptation			
n	Pre	Rec	F1	Std	Pre	Rec	F1	Std
4,000	12.8	8.7	9.7	2.6	14.1	11.6	**12.6**	2.5
6,000	22.4	16.2	17.9	1.7	29.7	20.7	**22.9**	2.1
8,000	25.0	19.2	20.9	2.6	38.6	24.6	**27.5**	2.4
10,000	35.1	22.6	24.8	2.5	41.4	28.3	**31.4**	2.5
12,000	45.8	26.1	29.3	0.9	46.9	30.5	**34.2**	2.2
14,000	46.9	28.9	32.7	0.8	47.8	33.3	**37.3**	2.0
16,000	50.0	32.0	36.1	1.6	49.9	35.4	**39.6**	2.8
18,000	52.2	35.3	39.6	1.8	51.8	37.3	**41.3**	1.4
20,000	53.3	36.0	40.7	1.2	54.0	39.1	**42.8**	1.6
22,000	53.2	39.0	43.3	1.1	55.4	40.6	**44.6**	0.2

4.2 GoEmotions

The results of the GoEmotions dataset are shown in Table 6 (random and upper bound baselines), 7 (model re-initialization), 8 (model updating), and 9 (two-step learning). As shown, the upper bound baseline achieves an F1-macro score of 47.2% on the test set (using all 43,410 samples for fine-tuning). In contrast with the experiments on the FRENK dataset, we observe that the upper bound baseline could not be matched with the utilized sample sizes. Nevertheless, it can be observed that task adaptation improves results for all AL approaches and sample sizes. Similarly to the results of the FRENK dataset, model re-initialization yields better results than checkpoint updating (regardless of whether this is done incrementally or through two-step learning). In the case of AL by checkpoint updating, we even observe a decrease in performance compared to random sampling when using 22,000 entries cumulatively for fine-tuning. This evidences the inefficiency of incremental fine-tuning of pre-trained language models on small data samples. Statistical significance for the experiments with model re-initialization in this multi-label setting was determined by conducting a McNemar test for each individual class for each sample size. A summary of these tests can be found in Appendix B, Table 18. The three emotions where statistically significant improvements were observed the most frequently were "approval", "confusion", and "amusement", and in total, 6 out of 28 emotions were never predicted more accurately with statistical significance.

Finally, Table 20 (Appendix C) shows that the improvements are primarily found in the most difficult classes, namely those that were not predicted in the experiments with random sampling: "annoyance", "caring", "confusion", "desire", "disappointment", "excitement", "fear" and "surprise". The bulk of these did also not get predicted when using standard AL, but improvements could be observed when using task adaptation.

Table 10: Macro-averaged results on **20 News Groups** using **random sampling**. The last row represents the upper bound baseline.

n	Pre	Rec	F1	Std
1,000	56.7	55.4	51.8	0.9
1,500	59.6	60.0	58.0	1.0
2,000	63.1	63.4	62.4	0.5
2,500	64.0	63.8	63.1	1.0
3,000	65.4	64.7	64.3	0.7
3,500	66.0	65.6	65.3	0.3
4,000	66.1	65.5	65.2	0.3
4,500	66.8	66.1	65.9	0.4
5,000	67.4	66.5	66.5	0.4
5,500	68.0	67.1	67.1	0.4
10,182	69.8	68.9	69.1	0.5

Table 11: Macro-averaged results on **20 News Groups** using AL with model **re-initialization**. Best F1 are in bold, experiments reaching the upper bound are in grey.

	Re-initialization				**+ Adaptation**			
n	**Pre**	**Rec**	**F1**	**Std**	**Pre**	**Rec**	**F1**	**Std**
1,000	56.7	55.4	51.8	0.9	62.2	60.9	**58.3*****	0.7
1,500	56.9	57.6	54.7	1.3	62.8	62.2	**60.2*****	0.8
2,000	59.8	60.6	58.9	1.3	64.6	64.2	**63.4*****	1.1
2,500	62.7	62.2	61.4	1.5	66.5	65.6	**65.3*****	0.5
3,000	64.4	63.9	63.5	0.8	67.4	66.6	**66.5*****	0.5
3,500	65.9	65.2	65.1	0.5	68.3	67.5	**67.4*****	0.2
4,000	66.8	66.2	66.1	0.5	68.5	67.8	**67.7*****	0.3
4,500	67.4	66.4	66.3	0.6	68.8	68.1	**68.0*****	0.9
5,000	68.1	67.3	67.3	0.4	69.4	68.6	**68.7*****	0.9
5,500	68.8	68.0	68.1	0.4	69.9	69.0	**69.2*****	0.2

Table 12: Macro-averaged results on **20 News Groups** using AL with model **updating**. Best F1 are in bold, experiments reaching the upper bound are in grey.

	Checkpoint updating				**+ Adaptation**			
n	**Pre**	**Rec**	**F1**	**Std**	**Pre**	**Rec**	**F1**	**Std**
1,000	56.7	55.4	51.8	0.9	62.2	60.9	**58.3**	0.7
1,500	54.3	52.7	50.0	3.7	62.1	58.5	**56.4**	1.0
2,000	57.1	54.9	54.4	2.3	62.7	61.3	**60.0**	2.3
2,500	59.5	58.5	58.0	0.8	64.1	63.3	**62.4**	1.2
3,000	63.8	60.2	60.1	1.5	66.0	64.1	**63.4**	1.5
3,500	63.7	62.2	63.3	1.3	67.0	65.5	**65.7**	0.9
4,000	64.9	63.2	64.2	0.5	67.7	66.4	**66.4**	0.5
4,500	66.4	64.4	65.2	0.5	67.6	66.5	**66.5**	1.0
5,000	66.5	65.2	65.5	0.2	68.3	66.6	**66.7**	0.8
5,500	66.7	64.7	**66.3**	0.3	67.4	65.5	65.2	1.7

Table 13: Macro-averaged results on **20 News Groups** using AL with **two-step learning**. Best F1 are in bold, experiments reaching the upper bound are in grey.

	Two-step				**+ Adaptation**			
n	**Pre**	**Rec**	**F1**	**Std**	**Pre**	**Rec**	**F1**	**Std**
1,000	56.7	55.4	51.8	0.9	62.2	60.9	**58.3**	0.7
1,500	54.3	52.7	50.0	3.7	62.1	58.5	**56.4**	1.0
2,000	59.0	56.4	54.4	2.3	63.8	61.6	**60.6**	1.5
2,500	59.9	59.3	58.0	0.8	64.9	62.8	**62.3**	1.6
3,000	62.5	60.9	60.1	1.5	65.8	64.6	**64.4**	1.6
3,500	64.6	63.4	63.3	1.3	67.2	65.9	**65.0**	0.9
4,000	65.2	64.3	64.2	0.5	67.5	66.4	**66.3**	0.9
4,500	66.2	65.3	65.2	0.5	68.1	66.9	**67.1**	0.5
5,000	66.5	65.6	65.5	0.2	68.6	67.5	**67.6**	0.3
5,500	67.1	66.3	66.3	0.3	69.0	68.0	**68.2**	0.4

4.3 20 News Groups

As shown in Table 10, the upper bound F1-macro score achieved on the 20 News Groups dataset is 69.1%. This score was (only) achieved when using the AL protocol with model re-initialization and task adaptation after sampling 5,500 training entries, which equals 54.0% of the available training data.

Similarly to the experiments on the other datasets, task adaptation improves all three explored AL protocols, although model re-initialization remains the best of the three. Interestingly, however, AL without task adaptation yields lower results than random sampling. For AL with model re-initialization, for example, there is an average decrease of 0.8% in F1-macro across all sample sizes. This shows that the standard uncertainty-based AL makes worse sampling choices than random sampling, and that uncertainty is therefore a suboptimal metric for measuring informativeness in this particular dataset. A possible explanation for this observation is that language models are often ill-calibrated and tend to be overconfident, even if their prediction is false (Yuan et al., 2020; Park and Caragea, 2022). Additionally, low prediction confidence may indicate that an entry is noisy, not just difficult to predict. A qualitative analysis of the entries that are selected early in the AL process shows that this is the case for the 20 News Groups dataset: many sampled posts contain merely a few words that are irrelevant to the topic, whereas other posts are lengthy and discuss a multitude of (irrelevant) topics causing the low prediction confidence in the classifier. Since data selection based on prediction uncertainty collects more noise than a random selection of data in this case, it prevents the model from learning useful information, especially in earlier samples.

After task adaptation, however, there is an average improvement of 2.6% over the random baseline, and an average gain of 3.2% over standard AL (improvements for all sample sizes are statistically significant). This shows that although sample selection based on model prediction uncertainty is suboptimal on some datasets, task adaptation can act as a safety net to avoid performance impairment due to the use of suboptimal metrics for measuring informativeness.

Finally, and similarly to the results observed on the other datasets, the highest performance increases are observed in the most difficult classes. For example, "talk.religion.misc" improves from 8.9% (random sampling) to 22.4% (AL with model re-initialization and task adaptation), as shown in Appendix C (Table 21).

The results of the experiments until this point have shown that task adaptation has a positive effect on AL: Significant improvements in F1-score could be observed, and in two datasets, the upper bound could be reached with a fraction of the annotations, while still showing substantial gains in the third dataset. In the case of 20 News Groups, we observed that traditional AL had a negative effect on model performance, although task adaptation countered this effect. Finally, the results indicate that AL with model re-initialization leads to more stable fine-tuning than with model updating.

4.4 Ablation study

In this section, we investigate the effect of task adaptation in isolation, and whether using AL still has beneficial effects on model performance after task adaptation. In order to gain insights into this matter, task adaptation was used without AL, i.e. with random sample selection. The result of this experiment was then compared to the result achieved with random sampling, standard AL, and task adaptation combined with AL (as reported in the previous subsections).

The results of the above mentioned experiments can be found in Table 14: For FRENK, it can be concluded that task adaptation alone does not improve results when using random sample selection, although using task adaptation and AL leads to improvements of 4.6% on average. In comparison, standard AL leads to average improvements of 1.7%. This surprising result indicates that BERT does not learn new knowledge from task adaptation, but that this technique causes better sample selection during the AL stage. There may be different reasons why task adaptation has less effect on FRENK than on the other datasets. For example, the data in FRENK could resemble the data used to pre-train BERT more than is the case for the other datasets, so that less new information is

Table 14: Improvements (F1) of task adaptation and AL over random sampling across all sample sizes (first sample size was not included in the calculations for the standard AL experiments, since this experiment is identical to that of random sample selection).

FRENK					**GoEmotions**					**20 News Groups**				
n	**Random**	**Random + adaptation**	**AL**	**AL + adaptation**	**n**	**Random**	**Random + adaptation**	**AL**	**AL + adaptation**	**n**	**Random**	**Random + adaptation**	**AL**	**AL + adaptation**
1,000	43.4	**45.7 (+2.3)**	43.4 (+0.0)	**45.7 (+2.3)**	4,000	9.7	**12.9 (+3.2)**	9.7 (+0.0)	**12.9 (+3.2)**	1,000	51.8	**58.3 (+7.5)**	51.8 (+0.0)	**58.3 (+7.5)**
1,500	45.9	46.8 (+0.9)	46.9 (+1.0)	**48.7 (+2.8)**	6,000	14.2	19.9 (+5.7)	17.0 (+2.8)	**24.7 (+10.5)**	1,500	58.0	**62.7 (+4.7)**	54.7 (-3.3)	60.2 (+2.2)
2,000	48.0	48.3 (+0.3)	48.5 (+0.5)	**51.8 (+3.8)**	8,000	21.5	28.6 (+7.1)	25.8 (+4.3)	**34.3 (+12.8)**	2,000	62.4	**64.9 (+2.5)**	58.9 (-4.0)	63.4 (+1.0)
2,500	52.0	50.0 (-2.0)	54.5 (+2.5)	**59.4 (+7.4)**	10,000	26.3	35.9 (+9.6)	34.2 (+7.9)	**39.3 (+13.0)**	2,500	63.1	**66.3 (+3.2)**	61.4 (-1.7)	65.3 (+2.3)
3,000	54.2	53.2 (-1.0)	55.0 (+0.8)	**59.4 (+5.2)**	12,000	32.7	38.9 (+6.2)	38.1 (+5.4)	**41.9 (+9.2)**	3,000	64.3	**67.2 (+2.9)**	63.5 (-0.8)	66.5 (+2.2)
3,500	52.9	52.6 (-0.3)	57.1 (+4.2)	**59.3 (+6.4)**	14,000	37.5	40.5 (+3.0)	40.8 (+3.3)	**44.2 (+7.3)**	3,500	65.3	**67.5 (+2.2)**	65.1 (-0.2)	67.4 (+2.1)
4,000	56.4	56.2 (-0.2)	57.4 (+1.0)	**61.0 (+4.6)**	16,000	39.7	41.2 (+1.5)	43.5 (+3.8)	**45.2 (+5.5)**	4,000	65.2	67.4 (+2.2)	66.1 (+0.9)	**67.7 (+2.5)**
Avg. improvement		+0.0	+1.7	**+4.6**	18,000	40.9	43.3 (+2.4)	43.9 (+3.0)	**45.5 (+3.0)**	4,500	65.9	67.9 (+2.0)	66.3 (+0.4)	**68.0 (+2.1)**
					20,000	41.8	43.9 (+2.1)	45.1 (+4.0)	**46.2 (+4.4)**	5,000	66.5	68.0 (+1.5)	67.3 (+0.8)	**68.7 (+2.2)**
					22,000	43.1	44.5 (+1.4)	45.6 (+2.5)	**46.3 (+3.2)**	5,500	67.1	67.5 (+0.4)	68.1 (+1.0)	**69.2 (+2.1)**
					Avg. improvement		+4.2	+4.1	**+7.2**	**Avg. improvement**		**+2.9**	-0.8	+2.6

being learned. Determining the exact cause of this observation, however, is beyond the scope of this paper and can be the subject of future work.

With respect to the GoEmotions dataset, which yielded the lowest F1-scores compared to FRENK and 20 News Groups, the improvements of using both task adaptation and AL was the highest of all datasets: 7.2% on average. Individually, task adaptation and AL showed improvements similar to each other, namely 4.2% and 4.1%, respectively.

Regarding 20 News Groups, we observe that standard active learning has a negative effect on performance overall, especially in the smaller sample sizes, as mentioned before. Task adaptation, however, shows improvements of 2.9% and 2.6%, without and with the additional use of AL, respectively. This shows that when active learning has a negative effect on performance, task adaptation can negate this effect and still improve upon standard fine-tuning with random data selection.

5 Conclusion

Main findings The current work combines uncertainty-based AL with task adaptation in order to learn from the data that could not be labeled due to limited annotation budget. It investigates the following research questions:

- What is the the effect of task adaptation on AL?
- How much data is needed to achieve the same performance with the proposed approach as with standard fine-tuning on all data?

The results of the experiments conducted on multiple datasets provide the following answers to these questions:

- Task adaptation has a significantly positive effect on AL, regardless whether the model is re-initialized or updated (although re-initialization consistently leads to better results). An analysis of the performance per class shows that the improvements are highest in the most difficult or underrepresented classes, and that the most difficult tasks in general show the highest improvements.
- In our experiments, 29.7% and 54% of all annotated data in the FRENK and 20 News Groups dataset, respectively, was needed to achieve the upper bound baseline. Although the proposed approach did not reach the upper bound baseline in the GoEmotions dataset with 50% or less of the training data, these results show that the proposed combination of approaches can lower annotation costs substantially.

Future research directions As mentioned earlier, AL with pre-trained language models remains understudied, in spite of it being an efficient annotation cost reduction method. Future research directions for AL with language models may include investigating the effect of calibration quality on uncertainty-based AL. An additional direction worth investigating is combining AL with other methods, such as data augmentation, weak supervision and domain adaptation, which are until now topics that are studied more extensively in other machine learning fields, such as computer vision (Zhao et al., 2020; Biegel et al., 2021; Xie et al., 2021; Hao et al., 2021; Zhan et al., 2022).

Limitations

The current work focuses on AL with pre-trained language models based on lowest prediction confidence. In spite of the effectiveness of the strategy shown both in these experiments and in previous work (Schröder et al., 2022; Ein-Dor et al., 2020), neural models are often not calibrated well (Yuan et al., 2020; Park and Caragea, 2022), which implies that the output of the softmax function could be a suboptimal metric for measuring prediction confidence, i.e. informativeness, for a given training sample. Future work on this topic should therefore investigate whether other metrics work better for AL with pre-trained language models and whether these metrics also benefit from unsupervised task adaptation. Additionally, experiments could only be conducted on a limited amount of tasks and datasets. Future work should shed new light on the usefulness of the proposed approach in different settings.

Acknowledgements

This research was funded by the Flemish government under FWO IRI project CLARIAH-VL and under the Research Program Artificial Intelligence Flanders.

References

Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. 2019. Deep batch active learning by diverse, uncertain gradient lower bounds. *CoRR*, abs/1906.03671.

Salman Aslam. 2023. Twitter by the numbers: Stats, demographics fun facts. [Accessed 17 April 2023].

Samantha Biegel, Rafah El-Khatib, Luiz Otávio Vilas Boas Oliveira, Max Baak, and Nanne Aben. 2021. Active WeaSuL: Improving weak supervision with active learning. *CoRR*, abs/2104.14847.

Jeska Buhmann, Maxime De Bruyn, Ehsan Lotfi, and Walter Daelemans. 2022. Domain- and task-adaptation for VaccinChatNL, a Dutch COVID-19 FAQ answering corpus and classification model. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3539–3549, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Dave Chaffey. 2023. Global social media statistics research summary 2023. SmartInsights. [Accessed 17 April 2023].

David Cohn, Les Atlas, and Richard Ladner. 1994. Improving generalization with active learning. *Machine Learning*, 15(2):201–221.

David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. 1996. Active learning with statistical models. *CoRR*, cs.AI/9603104.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A dataset of fine-grained emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4040–4054, Online. Association for Computational Linguistics.

Li Deng. 2012. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah A. Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *CoRR*, abs/2002.06305.

Liat Ein-Dor, Alon Halfon, Ariel Gera, Eyal Shnarch, Lena Dankin, Leshem Choshen, Marina Danilevsky, Ranit Aharonov, Yoav Katz, and Noam Slonim. 2020. Active Learning for BERT: An Empirical Study. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7949–7962, Online. Association for Computational Linguistics.

Ruqian Hao, Khashayar Namdar, Lin Liu, and Farzad Khalvati. 2021. A transfer learning–based active learning framework for brain tumor classification. *Frontiers in Artificial Intelligence*, 4.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. 2011. Bayesian active learning for classification and preference learning.

Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Chin-Yew Lin, and Deepak Ravichandran. 2001. Toward semantics-based answer pinpointing. In *Proceedings of the First International Conference on Human Language Technology Research*.

Peiyun Hu, Zachary C. Lipton, Anima Anandkumar, and Deva Ramanan. 2018. Active learning with partial feedback. *CoRR*, abs/1802.07427.

Fariz Ikhwantri, Samuel Louvan, Kemal Kurniawan, Bagas Abisena, Valdi Rachman, Alfan Farizki Wicaksono, and Rahmad Mahendra. 2018. Multi-task active learning for neural semantic role labeling on low resource conversational corpus. In *Proceedings of the Workshop on Deep Learning Approaches for Low-Resource NLP*, pages 43–50, Melbourne. Association for Computational Linguistics.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In Armand Prieditis and Stuart Russell, editors, *Machine Learning Proceedings 1995*, pages 331–339. Morgan Kaufmann, San Francisco (CA).

David D. Lewis and Jason Catlett. 1994. Heterogeneous uncertainty sampling for supervised learning. In William W. Cohen and Haym Hirsh, editors, *Machine Learning Proceedings 1994*, pages 148–156. Morgan Kaufmann, San Francisco (CA).

David D. Lewis and William A. Gale. 1994. A sequential algorithm for training text classifiers. *CoRR*, abs/cmp-lg/9407020.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Nikola Ljubesic, Darja Fiser, and Tomaz Erjavec. 2019. The FRENK datasets of socially unacceptable discourse in slovene and english. *CoRR*, abs/1906.02045.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Quinn McNemar. 1947. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157.

Shikib Mehri, Mihail Eric, and Dilek Hakkani-Tür. 2020. Dialoglue: A natural language understanding benchmark for task-oriented dialogue. *CoRR*, abs/2009.13570.

Seo Yeon Park and Cornelia Caragea. 2022. On the calibration of pre-trained language models using mixup guided by area under the margin and saliency. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5364–5374, Dublin, Ireland. Association for Computational Linguistics.

Roi Reichart, Katrin Tomanek, Udo Hahn, and Ari Rappoport. 2008. Multi-task active learning for linguistic annotations. In *Proceedings of ACL-08: HLT*, pages 861–869, Columbus, Ohio. Association for Computational Linguistics.

Guy Rotman and Roi Reichart. 2022. Multi-task active learning for pre-trained transformer-based models.

Maarten Sap, Swabha Swayamdipta, Laura Vianna, Xuhui Zhou, Yejin Choi, and Noah A. Smith. 2022. Annotators with attitudes: How annotator beliefs and identities bias toxic language detection. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5884–5906, Seattle, United States. Association for Computational Linguistics.

Christopher Schröder, Andreas Niekler, and Martin Potthast. 2022. Revisiting uncertainty-based query strategies for active learning with transformers. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2194–2203, Dublin, Ireland. Association for Computational Linguistics.

Burr Settles. 2009. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Yanyao Shen, Hyokun Yun, Zachary C. Lipton, Yakov Kronrod, and Animashree Anandkumar. 2017. Deep active learning for named entity recognition. *CoRR*, abs/1707.05928.

Jack Shepherd. 2023. 30 essential facebook statistics you need to know in 2023. TheSocialShepherd. [Accessed 17 April 2023].

Shuo Shi, Yuhai Liu, Yuehua Huang, Shihua Zhu, and Yong Liu. 2008. Active learning for kNN based on bagging features. In *2008 Fourth International Conference on Natural Computation*, volume 7, pages 61–64.

Tianze Shi, Adrian Benton, Igor Malioutov, and Ozan İrsoy. 2021. Diversity-aware batch active learning for dependency parsing. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2616–2626, Online. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Neeraj Vashistha, Kriti Singh, and Ramakant Shakya. 2022. Active learning for neural machine translation.

Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747.

Binhui Xie, Longhui Yuan, Shuang Li, Chi Harold Liu, Xinjing Cheng, and Guoren Wang. 2021. Active learning for domain adaptation: An energy-based approach. *CoRR*, abs/2112.01406.

Michelle Yuan, Hsuan-Tien Lin, and Jordan Boyd-Graber. 2020. Cold-start active learning through self-supervised language modeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7935–7948, Online. Association for Computational Linguistics.

Xueying Zhan, Qingzhong Wang, Kuan hao Huang, Haoyi Xiong, Dejing Dou, and Antoni B. Chan. 2022. A comparative survey of deep active learning.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level Convolutional Networks for Text Classification. *arXiv:1509.01626 [cs]*.

Yuekai Zhao, Haoran Zhang, Shuchang Zhou, and Zhihua Zhang. 2020. Active learning approaches to enhancing neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1796–1806, Online. Association for Computational Linguistics.

Jingbo Zhu and Eduard Hovy. 2007. Active learning for word sense disambiguation with methods for addressing the class imbalance problem. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 783–790, Prague, Czech Republic. Association for Computational Linguistics.

Jingbo Zhu, Huizhen Wang, Tianshun Yao, and Benjamin K Tsou. 2008. Active learning with sampling by uncertainty and density for word sense disambiguation and text classification. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 1137–1144, Manchester, UK. Coling 2008 Organizing Committee.

A Label distribution per dataset

Table 15: Class distribution of the FRENK dataset after combining the violent and offensive classes.

Class name	Train	Val	Test
Violent language	96	11	15
Offensive language	1,487	165	477
Inappropriate language	1,490	165	410
Acceptable language	5,331	592	1,399

Table 16: Class distribution of the 20 News Groups dataset.

Class name	Train	Val	Test
alt.atheism	534	59	394
comp.graphics	538	60	398
comp.os.ms-windows.misc	508	56	376
comp.sys.ibm.pc.hardware	535	60	396
comp.sys.mac.hardware	532	59	393
comp.windows.x	520	58	385
misc.forsale	526	59	390
rec.autos	418	47	310
rec.motorcycles	526	58	389
rec.sport.baseball	491	55	364
rec.sport.hockey	534	59	395
sci.crypt	535	59	396
sci.electronics	539	60	398
sci.med	339	38	251
sci.space	535	59	396
soc.religion.christian	540	60	399
talk.politics.guns	531	59	392
talk.politics.mideast	432	48	319
talk.politics.misc	532	59	394
talk.religion.misc	537	60	397

Table 17: Class distribution of the GoEmotions dataset.

Class name	id	Train	Val	Test
Admiration	0	4,130	488	504
Amusement	1	2,328	303	252
Anger	2	1,567	195	197
Annoyance	3	2,470	303	286
Approval	4	2,939	397	318
Caring	5	1,087	153	114
Confusion	6	1,368	152	139
Curiosity	7	2,191	248	233
Desire	8	641	77	74
Disappointment	9	1,269	163	127
Disapproval	10	2,022	292	220
Disgust	11	793	97	84
Embarrassment	12	303	35	30
Excitement	13	853	96	84
Fear	14	596	90	74
Gratitude	15	2,662	358	288
Grief	16	77	13	6
Joy	17	1,452	172	116
Love	18	2,086	252	169
Nervousness	19	164	21	16
Optimism	20	1,581	209	120
Pride	21	111	15	8
Realization	22	1110	127	109
Relief	23	153	18	7
Remorse	24	545	68	46
Sadness	25	1,326	143	108
Surprise	26	1,060	129	92
Neutral	27	14,219	1,766	1,606

B GoEmotions significance per class

Table 18: Indices of emotions that were predicted significantly more accurately when using task adaptation before AL with model re-initialization (per sample size).

n	*	**	***
4,000	-	-	0, 1, 18
6,000	20	26	1, 2, 4, 7, 14, 17, 25
8,000	1, 8, 9, 26	24	13
10,000	6, 10, 24, 25, 26	27	4
12,000	2, 13, 20	-	4
14,000	14, 27	3, 6, 10	4, 12
16,000	6	9, 12, 22	4
18,000	3, 5, 6, 12, 14	-	4, 22
20,000	2, 6	4, 22	-
22,000	1, 22	-	-

C Results per class

Table 19: Comparison of results per class on the FRENK dataset after fine-tuning on 4,000 entries using different sampling approaches. The reported results are averaged across 5 runs with different random seeds. The best results are in bold.

	Random			**Re-initialization**			**+ Adaptation**		
Class name	**Pre**	**Rec**	**F1**	**Pre**	**Rec**	**F1**	**Pre**	**Rec**	**F1**
Acceptable speech	76.0	87.7	81.4	76.2	88.5	81.8	77.4	87.4	**82.1**
Offensive speech	56.4	44.5	49.5	58.1	46.7	51.3	59.5	50.3	**54.1**
Violent speech	73.3	10.7	18.3	54.0	24.0	32.3	61.2	38.7	**47.1**
Inappropriate speech	63.2	47.3	54.0	65.3	44.2	52.6	64.3	47.3	**54.3**

Table 20: Comparison of results per class on the GoEmotions dataset after fine-tuning on 22,000 entries using different sampling approaches. The reported results are averaged across 5 runs with different random seeds. The best results are in bold.

	Random			**Re-initialization**			**+ Adaptation**		
Class	Pre	Rec	F1	Pre	Rec	F1	Pre	Rec	F1
Admiration	71.8	60.8	65.8	70.2	63.8	66.8	68.3	66.0	**67.0**
Amusement	80.9	81.4	81.2	81.5	80.1	80.7	80.4	84.9	**82.6**
Anger	67.9	10.3	17.4	64.5	27.5	37.8	61.3	33.4	**43.0**
Annoyance	0.0	0.0	0.0	0.0	0.0	0.0	24.3	1.2	**2.4**
Approval	58.1	23.0	32.9	32.6	2.6	4.8	51.4	25.3	**33.2**
Caring	0.0	0.0	0.0	0.0	0.0	0.0	59.9	17.8	**26.0**
Confusion	0.0	0.0	0.0	0.0	0.0	0.0	47.2	4.8	**8.7**
Curiosity	58.2	36.7	**44.9**	45.2	24.8	31.5	54.8	35.8	43.3
Desire	0.0	0.0	0.0	63.7	9.4	15.9	67.1	20.2	**30.3**
Disappointment	0.0	0.0	0.0	0.0	0.0	0.0	53.7	4.9	**8.5**
Disapproval	38.5	6.7	10.8	34.5	3.1	5.4	46.9	19.4	**27.4**
Disgust	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Embarrassment	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Excitement	0.0	0.0	0.0	20.0	0.6	1.1	79.5	19.2	**30.5**
Fear	0.0	0.0	0.0	66.5	33.8	43.9	67.5	63.3	**65.3**
Gratitude	93.4	88.0	90.6	92.8	89.1	90.9	93.0	89.3	**91.1**
Grief	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Joy	73.2	30.8	42.8	72.4	43.5	53.9	69.1	48.7	**57.1**
Love	82.0	77.0	79.4	81.1	80.5	**80.8**	82.0	78.7	80.3
Nervousness	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Optimism	70.7	31.9	43.8	69.5	38.8	49.5	67.0	42.2	**51.7**
Pride	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Realization	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Relief	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Remorse	35.7	4.3	7.0	49.3	30.0	34.4	63.0	52.1	**56.9**
Sadness	77.0	14.6	23.6	74.4	37.3	49.2	68.1	40.4	**50.6**
Surprise	0.0	0.0	0.0	41.6	8.7	13.0	59.4	33.9	**42.8**
Neutral	67.7	56.6	61.6	65.9	59.2	**62.4**	67.4	57.3	61.8

Table 21: Comparison of results per class on the 20 News Groups dataset after fine-tuning on 4,500 entries using different sampling approaches. The reported results are averaged across 5 runs with different random seeds. The best results are in bold

	Random			**Re-initialization**			**+ Adaptation**		
Class name	**Pre**	**Rec**	**F1**	**Pre**	**Rec**	**F1**	**Pre**	**Rec**	**F1**
alt.atheism	39.3	57.0	**46.4**	44.0	47.0	45.1	46.5	47.0	46.1
comp.graphics	64.0	68.2	65.9	66.4	68.7	67.5	66.6	70.7	**68.4**
comp.os.ms-windows.misc	59.7	63.2	61.3	65.6	61.2	63.2	66.5	63.9	**65.1**
comp.sys.ibm.pc.hardware	61.3	57.1	59.1	59.8	64.7	61.9	64.8	67.3	**65.9**
comp.sys.mac.hardware	49.8	65.5	56.1	70.5	65.3	67.7	72.9	70.4	**71.5**
com.windows.x	80.4	73.2	76.5	82.4	74.7	78.3	81.9	76.6	**79.1**
misc.forsale	75.6	79.3	77.4	82.7	80.5	81.6	82.3	82.6	**82.3**
rec.autos	65.8	76.0	**70.1**	52.1	76.3	61.9	51.7	75.1	61.2
rec.motorcycles	71.1	65.9	68.3	71.8	66.7	69.1	76.5	68.7	**72.3**
rec.sport.baseball	88.9	82.5	85.5	85.8	82.1	83.8	90.1	82.7	**86.2**
rec.sport.hockey	93.7	84.1	**88.6**	90.6	83.1	86.7	89.9	86.8	88.2
sci.crypt	72.5	68.6	70.5	75.1	65.8	70.1	79.6	67.7	**73.1**
sci.electronics	58.2	57.7	57.9	57.9	60.0	58.9	62.9	60.5	**61.6**
sci.med	82.3	80.2	81.2	84.6	80.4	**82.4**	82.0	82.8	**82.4**
sci.space	76.0	73.8	**74.8**	70.2	76.3	73.1	72.7	76.4	74.5
soc.religion.christian	68.2	74.0	70.8	68.3	75.5	71.6	67.4	81.2	**73.6**
talk.politics.guns	53.2	63.6	57.9	53.5	63.9	58.1	56.4	65.5	**60.5**
talk.politics.mideast	89.3	73.9	**80.9**	87.2	73.6	79.7	83.0	74.4	78.2
talk.politics.misc	54.7	43.9	**48.6**	46.5	47.9	47.0	52.4	43.3	47.3
talk.religion.misc	30.1	5.5	8.9	32.1	13.8	18.5	29.9	18.2	**22.4**

Improving Dutch Vaccine Hesitancy Monitoring via Multi-Label Data Augmentation with GPT-3.5

Jens Van Nooten, Walter Daelemans
CLiPS (University of Antwerp)
Lange Winkelstraat 40, 2000 Antwerp (Belgium)
`firstname.lastname@uantwerpen.be`

Abstract

In this paper, we leverage the GPT-3.5 language model both using the Chat-GPT API interface and the GPT-3.5 API interface to generate realistic examples of anti-vaccination tweets in Dutch with the aim of augmenting an imbalanced multi-label vaccine hesitancy argumentation classification dataset. In line with previous research, we devise a prompt that, on the one hand, instructs the model to generate realistic examples based on the human dataset (gold standard) and, on the other hand, to assign one or multiple labels to the generated instances. We then augment our gold standard data with the generated examples and evaluate the impact thereof in a cross-validation setting with several state-of-the-art Dutch BERT models. This augmentation technique predominantly shows improvements in F1 for classifying underrepresented classes while increasing the overall recall, paired with a slight decrease in precision for more common classes. Furthermore, we examine how well the synthetic data generalises to human data in the classification task. To our knowledge, we are the first to utilise Chat-GPT and GPT-3.5 for augmenting a Dutch multi-label dataset classification task.

1 Introduction

In many text classification settings, the label distribution in datasets is imbalanced, which impacts the learning process of machine learning models and often leads to a degraded performance (Tarekegn et al., 2021). This phenomenon is especially inherent to multi-label datasets and therefore complicates multi-label classification further. Multiple approaches for tackling classification tasks where the data is imbalanced have been proposed. These approaches, which aim to alleviate the issue of class imbalance, comprise multiple categories and can be divided into data re-sampling, classifier adaptation and ensemble methods (Tarekegn et al., 2021). Bayer et al. (2022) describe multiple Data Augmentation (DA) methods, which encompass DA of the text itself at character, word, sentence/paragraph, or document level. These methods include noise induction by replacing and swapping characters or words, embedding replacement, replacing words with synonyms, translation, or even a combination of multiple methods.

With the popularisation of large generative models such as the GPT models (Radford et al., 2019), researchers have leveraged them for DA purposes, ranging from augmenting text translation datasets (Sawai et al., 2021), to generating examples for an Event Detection dataset (Pouran Ben Veyseh et al., 2021) and augmenting a COVID-19 Q&A dataset (Zhou and Zhang, 2021). However, most research is restricted to binary or multi-class classification, thus leaving much uncovered ground to research the potential benefits of augmenting multi-label data. Additionally, the cross-lingual capabilities of the State-of-the-Art (SotA) generative models for DA purposes remain under-explored. In this paper, we augment a Dutch dataset for vaccine hesitancy argumentation monitoring, as introduced in Lemmens et al. (2021) using SotA language models, including Chat-GPT, and build upon previous research by introducing these generative models in a realistic multi-label text classification setting. Additionally, we contextualise the performance with a strong back-translation baseline and extensively compare different GPT-based DA methods for multi-label text classification.

Our contributions are the following:

- We build upon previous research that utilises large language models to augment datasets by extending the SotA techniques to a multi-label setting and showing their effectiveness.
- We bring new insights into applying the SotA DA techniques for a real-world application of multi-label text classification.
- We extensively compare data generated by the two instances of the GPT-3.5 model.

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 251–270
July 14, 2023 ©2023 Association for Computational Linguistics

The present work is organised as follows: Section 2 provides an overview of research into data augmentation and the most recent generative models. Section 3 describes the datasets that are used for the experiments, including how the generative models were prompted, introduces the classification models used and outlines the evaluation methods. In Section 4, the results from the experiments are summarised and additional extensive analyses are conducted. Lastly, Section 5 briefly summarises the presented methodology and findings, along with outlining steps for future work.

2 Related Work

2.1 Generative Models

The GPT models are generative language models developed by OpenAI that have showcased an immense versatility in tasks that they can perform, ranging from classification and translation to summarisation (Radford et al., 2019). The models are auto-regressive, meaning that the models take previous outputs that they have generated into account for future outputs.

GPT-3.5 (or InstructGPT) is one of the most recent additions to OpenAI's roster of models, which is a fine-tuned version of GPT-3 using Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022). Chat-GPT[1] is the latest installment of GPT-3-based models and was released in December 2022. The web interface and API have already attracted many users and researchers for various use cases, such as text summarisation, creative and academic text generation, translation, and so on. Chat-GPT is quite similar to GPT-3.5, in that the model generates an answer, given an instruction as a prompt. Analogous to GPT-3.5's training method, Chat-GPT was trained using RLHF (Ouyang et al., 2022).

2.2 GPT Models as In-Context Learners

GPT models have already displayed impressive in-context learning and few-shot learning capabilities (Radford et al., 2019; Brown et al., 2020): based on a handful of gold standard examples that are included in the prompt, the models have shown SoTA performance on a variety of tasks. However, GPT-3's performance is heavily dependent on the selection of prompt examples, as mentioned in Liu et al. (2022) and Min et al. (2022). The former authors denote the in-context learning scenario with GPT-3 as a conditional text generation problem, where given context (which can include the task description and gold standard text examples), several target samples are generated. To find the optimal gold standard samples to include in the prompt as the context, the researchers leveraged RoBERTa (and variations thereof fine-tuned on NLI tasks) to embed training texts and retrieve the most similar ones to the test source. Their approach was applied to the SST-2 benchmark and showed an improvement of 6% accuracy compared to randomly selecting examples from the training data, in addition to superior performance compared to T5 on multiple Question-Answering datasets.

With these in-context learning capabilities in mind and taking inspiration from the MixUp learning technique in computer vision (Zhang et al., 2017), Yoo et al. (2021) introduced a multi-task prompt for GPT-3 that lets the model generate data and simultaneously labels the generated data. With the inclusion of examples from the training data in the prompt, the researchers aimed to generate examples that are close to the training distribution. With their multi-task approach, they reached superior results to other DA techniques on GLUE benchmark datasets.

In this paper, we implement a similar technique for augmenting a Dutch multi-label dataset, thus leveraging the in-context learning and cross-lingual abilities of Chat-GPT and GPT-3.5.

2.3 Data Augmentation

Data scarcity is a common problem in machine learning, and more so in commercial settings. This can manifest itself in data imbalance, leading to a degradation in performance on downstream classification tasks. DA aims to alleviate this problem by artificially enhancing the dataset through transformations or synthetic additions, while still preserving class information (Bayer et al., 2022). Consequently, DA also circumvents the high costs of human annotations (Bayer et al., 2022). More generally speaking, DA can be understood as "a strategy to prevent overfitting via regularization" (Shorten et al., 2021).

Bayer et al. (2022) describe three main categories of DA. First, augmentation in the data space, which is concerned with transformations of the data in its textual form on either character level, word level, phrase/sentence level, or document level. A wide array of different techniques and approaches

---

[1] `https://openai.com/blog/chatgpt`

have been proposed at each of these levels, such as noise induction in the shape of random swap and deletion of words (Wei and Zou, 2019). Other approaches include synonym replacement based on WordNet or thesauri (replacement by synonyms) (Kolomiyets et al., 2011; Li et al., 2017), embedding replacement (replacement by words with a similar latent representation) (Rizos et al., 2019) and back-translation (Xie et al., 2020). The latter describes the method of translating a text entry to another language and then translating it back to the source language, thus yielding a paraphrase or a slightly different variation of the original text and keeping the same gold-standard label(s).

A great deal of recent work on DA has explored the usage of generative models. Multiple researchers have explored SotA encoder/decoder models, such as BART (Kumar et al., 2020; Abonizio et al., 2022), while others focused on utilizing the popular GPT models for augmenting datasets, though few have leveraged them for augmenting multi-label datasets. For example, Zhang et al. (2020) have utilised GPT-2 for augmenting their dataset for extreme multilabel classification (XMC), a setting where a large pool of labels is available and to which data imbalance is inherent as a result. By comparing rule-based augmentation, Wordnet-based augmentation and augmentation based on text pairs with the same labels as a prompt in GPT-2, the authors found GPT-2 to be the most effective approach for improving the model's performance. Similarly, Dirting et al. (2022) generated synthetic Facebook comments that contain hate speech using GPT-2 to balance out a multi-label hate speech detection dataset, after which pseudo-labels were added to these comments by using the prediction of a fine-tuned BERT classifier.

3 Methodology

In this work, we aim to build upon existing research by augmenting a Dutch multi-label vaccine hesitancy detection dataset using novel generative models. Given the capabilities of the most recent models, we aim to generate realistic examples of vaccine-hesitant tweets to alleviate the data scarcity issue of the present application. We hypothesise that introducing new synthetic examples in the training data will improve the performance of large language models on the mentioned down-stream classification task.

3.1 Datasets

3.1.1 Vaccinpraat

For the experiments, we used the Vaccinpraat dataset for vaccine hesitancy argumentation classification, which was introduced in Lemmens et al. (2021). The dataset is used to classify the reason(s) why the writer of a tweet or Facebook comment is vaccine-hesitant. In total, there are nine labels: alternative medicine, conspiracy, criticism of vaccination strategy, development, efficacy, institutional motives, liberty, morality and safety. A description of these labels can be found in Lemmens et al. (2021). This dataset consists of 8,244 tweets and 3,917 Facebook comments in Dutch that are annotated with a binary label for vaccine stance (hesitant or not) and if the entry expresses vaccine hesitancy, the entries are annotated with one or multiple of the previously mentioned classes ("arguments"). For the present experiments, we only use the Twitter portion of the data [2].

The label distribution of this dataset and the generated datasets is visualised in Figure 1 and shows that the dataset is imbalanced, where the 'morality' and 'alternative medicine' are the least frequent classes. The dataset also contains 1,453 instances where no label is assigned.

3.1.2 Generated Datasets

Prompt Construction As mentioned previously, GPT-3.5 and Chat-GPT[3] were used as generative models. Taking inspiration from the method described in Yoo et al. (2021), an English prompt was devised that instructs the models to generate items in Dutch based on the description of labels in addition to examples from the gold standard data and assigns one or multiple labels to these items. The following paragraphs describe how the prompt was constructed.

The prompt itself consists of three main parts, namely the labels and their descriptions, the description of the task and the examples from the gold standard data.

- **Label Descriptions.** The label descriptions were retrieved from the Vaccinpraat website, which conform to the annotation guidelines for the gold standard dataset. These descriptions were then manually translated to En-

---

[2] This way we only have to generate synthetic tweets and do not have to mix different sources of text in training.

[3] We used the model provided in the API released on March 1 2023.

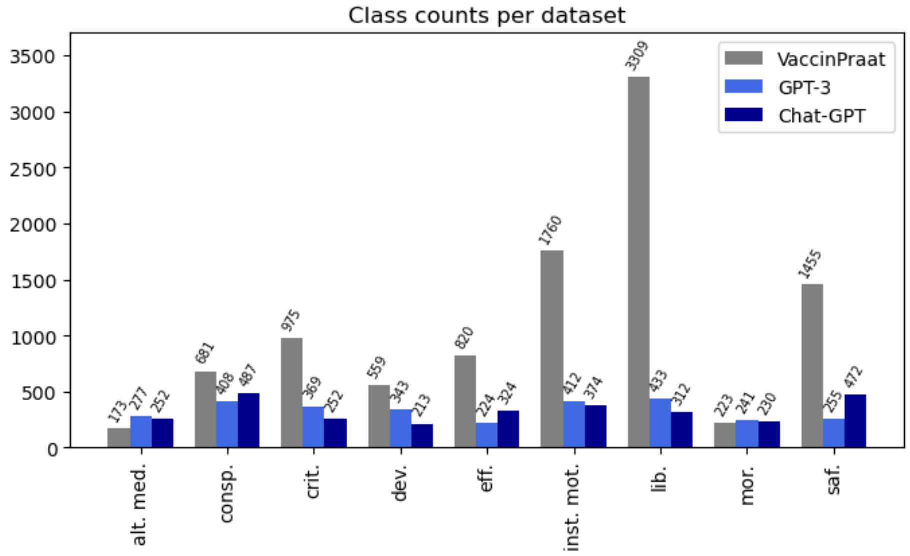


Figure 1: Distribution of the classes in the tweets portion of the Vaccinpraat dataset, in addition to the datasets generated with GPT-3.5 and Chat-GPT. The total number of instances does not correspond to the number of entries in the dataset, because the data is multi-label.

glish[4], after which clarifications of how the label is manifested in the data were added to the descriptions. In order to retrieve such clarifications, named entities, nouns and general phenomena for a certain label were selected manually from texts that belong to a certain label. For example, tweets with the label 'alternative medicine' often referred to ivermectine or hydroxochloroquine, so this was added to the label description as follows:

> "alternative_medicine. This label refers to alternatives to the COVID vaccine, such as natural remedies or other medicine, such as *ivermectine* or *hydroxychloroquine*."

- **Task Description.** The models were instructed (in English) to generate Dutch anti-vaccination tweets and label the data it generates at the same time. Additionally, we also prompt the model to generate instances that are similar to some provided examples.

- **Example Selection.** In order to select text-label pairs from the dataset, one multi-label and one single-label example were manually selected that represent the label(s) well to guide the model to produce text in the correct output format.

Generative Model Parameters For GPT-3, we used the *text-davinci-003* completion model, for which the temperature parameter was set to .6 [5], while the 'top p', 'frequency penalty' and 'presence penalty' were all set to 1. The model generated 500 tokens maximum.

For Chat-GPT, the standard implementation of the *gpt-3.5-turbo* model was used. No additional contextual messages besides the prompt were added to the conversation as history. Details about the generated datasets can be found in Table 1. With Chat-GPT and GPT-3.5, 1,700 instances were generated using the same prompt as described in Section 3.1.2. GPT-3.5 generated shorter and more lexically diverse tweets, as indicated by the higher Type-Token Ratio (TTR). Examples of the generated datasets can be found in Appendix 6 and 7. These examples also showcase some differences between two versions of the generative model: though GPT-3.5 generates slightly more lexically diverse data, the data is sometimes ungrammatical.

Generated Data The distributions of the two synthetic datasets differ slightly, in that GPT-3.5 generated fewer instances for 'efficacy' and 'safety' than Chat-GPT, but more for the 'liberty' and 'safety' classes. Moreover, GPT-3.5 generated 80 unique label combinations in total, while Chat-GPT generated 72 unique combinations. In comparison, the

---

[4] Experiments were also conducted where the label descriptions were in Dutch, though this did not yield a significant difference in performance or class distribution.

[5] We aimed for a balance between consistency and creativity. Any higher value for temperature yielded ungrammatical text after the model had generated a tweet.

Dataset	Length	TTR
GPT-3.5	27 (6)	.10
Chat-GPT	32 (7)	.07
Vaccinpraat	32 (14)	.07

Table 1: Average length of the generated data in tokens (with standard deviation) and average type-token ratio.

gold-standard data contains 104 unique label combinations. Additionally, GPT-3.5 generated slightly more examples with one assigned label than Chat-GPT (cf. Figure 2). The label co-occurrence matrices for each dataset can be found in Appendix 6 and 7.

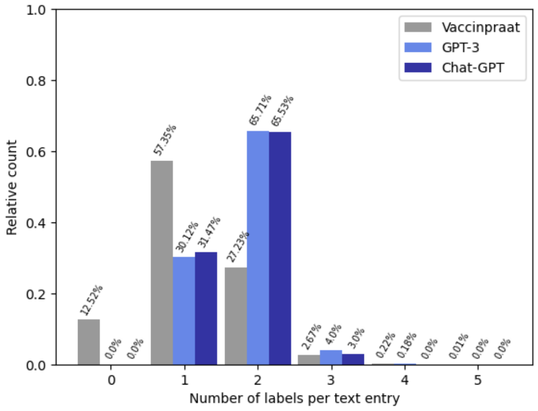


Figure 2: The number of labels per generated instance for both models.

3.2 Back-translation Baseline

As a baseline for the experiments, we opted for back-translation. The Dutch tweets were translated to English and then back to Dutch by employing an ALIGN transformers model (Garg et al., 2019)[6].

3.3 Classification Models

Several large language models that were trained on Dutch were utilised for the experiments:

1. **BERTje**: A Dutch version of BERT (de Vries et al., 2019).
2. **RobBERT-v2**: A Dutch version of RoBERTa (Delobelle et al., 2020).
3. **CoNTACT**: A domain-adapted version of RobBERT, specialised in COVID-related tweets (Lemmens et al., 2022).
4. **RobBERTje**: A distilled version of RobBERT (Delobelle et al., 2021).

All models were trained for 5 epochs and the best performing model based in terms of the loss on the validation set was chosen to make predictions on the test fold. For all models, a learning rate of 5e-5, a batch size of 16 and a maximum sequence length of 256 were used. As the input for the classification head of the models, we used the hidden state of the CLS token at the last layer of the model, thereby following the original implementation of BERT (Devlin et al., 2019). We used Binary Cross Entropy Loss (with a built-in sigmoid activation applied to the model logits), the AdamW optimizer and a learning rate scheduler with linear decay. These hyper-parameters remained consistent across all models.

3.4 Model Evaluation

The gold standard dataset was split into five stratified parts of equal length, out of which one fold was used as a test set for each repetition. The synthetic data was then added to the remaining training data. In order to prevent data leakage with the back-translation baseline, we selected 1,700 random samples from each training fold to be back-translated[7]. We repeated each experiment on each fold five times, each time with a different random seed. The performance of the models is reported in terms of the average precision, recall and F1 (all macro- and micro-averaged).

4 Results and Discussion

In this section, the results for the experiments are summarised. First, the effect of data augmentation will be discussed. Then, in order to gain more insight into the nature of the synthetic data and the generative models, additional sets of experiments were conducted. First, we perform a typicality study on the synthetic and human data. Then, we directly compare the data generated by Chat-GPT and the GPT-3.5 API by balancing the data and utilizing the balanced sets for the classification tasks in order to determine if there is a clear difference in quality between the two synthetic datasets. Additionally, we measure how consistent the generative models are in labeling the generated instances by training and testing RobBERT on the synthetic data only. Furthermore, we investigate how well the synthetic data generalises to human-written data by training models on the synthetic data only and

---

[6] We used the Helsinki-NLP/opus-mt-en-nl and Helsinki-NLP/opus-mt-nl-en model on the HuggingFace hub.

[7] Experiments were also conducted where all training data was back-translated.

testing on the cross-validation test sets. Finally, we evaluate the impact of the synthetic data in a cross-platform setting where the models are trained on Twitter data and tested on Facebook comments.

4.1 Effect of Data Augmentation

The results for each model are visualised in Figure 3[8]. The results per class for RobBERT can be found in Table 2 (more detailed results and the results for the other models can be found in Appendix 9 - 12). Overall, DA leads to a significant increase in performance across all models. Examining the results more closely, it can be observed that the performance increases are the most significant for under-represented classes in the dataset, such as 'alternative medicine', 'development' and 'criticism of vaccination strategy' (cf. Table 2). Most notably, the 'alternative medicine' class sees an increase of more than 40 F1 points across all models. Additionally, we observe a small, though statistically significant[9] increase for the 'morality' class for some models. However, it remains the most difficult class to classify for all models. The effect of DA is also less notable on classes that occur more frequently in the gold standard dataset, such as 'institutional motives' and 'liberty', where some models show no improvement or only a minor improvement[10]. In general, we observe a reduction in false positives and an increase in true positives across all classes. However, this is paired with an increase in false negatives as well for most classes (cf. Figure 12 - 16).

Even though the augmentation with generated data yields a significant performance increase for all models, back-translation yields a roughly equal performance to the GPT-based augmentation, except for RobBERT, where the GPT-based augmentation significantly outperforms back-translation. However, both methods seem to complement each other when they are combined, since this leads to the highest performance for almost every class across all models[11] (cf. Figure 9-12.).

4.1.1 Data Typicality

We also calculated the typicality for the synthetic and gold-standard datasets (Zhang, 1992). Typicality measures how well an instance represents instances from the same label in a dataset. This concept is especially meaningful for the synthetic datasets to measure how well the generative models can capture the patterns in the labels. Therefore, it serves as an indication for how prototypical the synthetic data is compared to itself and/or the human data. For this, we group the instances per label[12] and embed all instances with Sentence Transformers[13] (Reimers and Gurevych, 2019). Then, for each generated instance from dataset a, we average the cosine similarity (sim) between the generated instance and each entry from dataset a (or gold standard dataset b) belonging to the same label ($N_l^{(a)}$):

$$\frac{1}{N_l^{(a)}} \sum_{j=1}^{N_l^{(a)}} sim(\mathbf{a}_i, \mathbf{a}_j^{(l)})$$

This is then divided by the average cosine similarity between the generated instance and the cosine similarity with entries from the same dataset (or gold standard dataset b) from all other labels ($N_k^{(a)}$):

$$\frac{1}{(L-1)\sum_{k\neq l}^{L} N_k^{(b)}} \sum_{k\neq l}^{L} \sum_{j=1}^{N_k^{(a)}} sim(\mathbf{a}_i, \mathbf{a}_j^{(k)})$$

The results of this analysis are summarised in Table 3. On average, the instances from both synthetic datasets are more prototypical than the gold standard data (intra-dataset typicality), as can be derived from Table 1. Comparing the synthetic instances to the human data (inter-dataset typicality), the same observations are made. This could be explained by Chat-GPT and GPT-3.5 possibly not accessing knowledge about very specific events or political figures, thus generating more generic or prototypical texts compared to the gold standard. Ultimately, the generated data is a distillation of what the GPT models have been shown during training.

---

[8] The complete experimental matrix can be found in Appendix 8.

[9] The statistical significance was calculated with the McNemar test.

[10] Future work could explore adjusting the prompt to generate more diverse examples for these classes.

[11] An even higher performance for all models is yielded when the entire training set is back-translated and the GPT-generated data is added. The results for these experiments can be found in Table 8.

[12] Since this is a multi-label dataset, we copy a text n times for n number of labels assigned to that text.

[13] For this, the 'textgain/allnli-GroNLP-bert-base-dutch-cased' model on the Hugging Face hub was used.

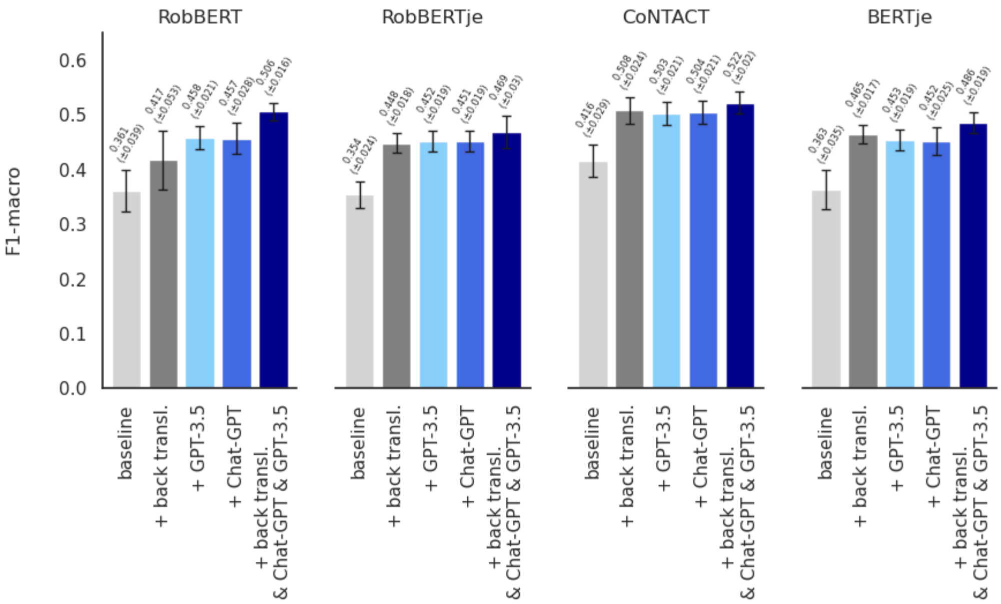


Figure 3: Performance (macro-averaged F1 scores) of the models, averaged across all folds and random seeds.

	Baseline	+ Back-transl.	+ GPT-3.5	+ Chat-GPT	+ Back-transl + GPT-3.5 + Chat-GPT
alt med	.01 (.02)	**.14** (.13)	**.5** (.09)	**.44** (.11)	**.49** (.08)
consp	.44 (.05)	**.48** (.04)	**.48** (.05)	**.49** (.06)	**.52** (.04)
crit	.15 (.1)	**.29** (.09)	**.26** (.06	**.28** (.06)	**.34** (.05)
dev	.2 (.14)	**.32** (.12)	**.36** (.1)	**.34** (.11)	**.42** (.08)
eff	.48 (.07)	**.51** (.09)	**.49** (.06)	**.51** (.06)	**.56** (.02)
inst	.58 (.05)	**.58** (.08)	.57 (.04)	**.58** (.02)	**.6** (.02)
lib	.76 (.02)	.75 (.02)	**.77** (.02)	**.76** (.02)	**.76** (.02)
mor	.0 (.0)	**.03** (.04)	**.06** (.07)	**.07** (.05)	**.19** (.05)
saf	.64 (.02)	**.65** (.02)	**.64** (.02)	**.64** (.02)	**.67** (.01)
micro avg.	.59 (.02)	**.6** (.02)	**.6** (.01)	**.6** (.01)	**.62** (.01)
macro avg.	.36 (.04)	**.42** (.05)	**.46** (.02)	**.46** (.02)	**.51** (.02)

Table 2: Classification results (F1) per class from RobBERT (averaged across folds and random seeds).

4.2 Comparing Chat-GPT and GPT-3.5

In order to provide a fair comparison between the two generated datasets to measure the text quality for the classification task, both the Chat-GPT and GPT-3 datasets were balanced by performing random undersampling so that each label occurs n times, where n is equal to the occurrence count of the least frequent class across both datasets. The undersampling in this case was necessary, as both generative models did not generate an equal distribution of data. In this case, both datasets were balanced to have 213 instances per label. The results of these supplemental experiments are summarised in the experimental matrix in Appendix 8. For all models, a statistically significant difference [14] in performance is observed between the two balanced setups, though the performance difference in terms of F1 between Chat-GPT and GPT-3 is negligible.

4.3 Labeling Consistency

In order to validate whether the GPT models label the generated instances consistently, we conducted an extra set of cross-validation experiments on the synthetic data only. If there are noticeable label inconsistencies in the generated datasets, the standard deviation of the performance scores on the folds should be high. The results of these experiments are summarised in Table 4. Comparing Chat-GPT, GPT-3 and the gold standard data with each other, RobBERT shows a smaller standard deviation between folds when it is trained on the Chat-GPT dataset, thus suggesting that the labels are more consistent in this dataset than the other datasets.

---

[14] $P < 0.001$ for RobBERT, RobBERTje and BERTje, $p = 0.03$ for ConTACT

Dataset	Intra-dataset typicality	Inter-dataset typicality	Cosine
GPT-3.5	1.23 (.19)	1.17 (.57)	.35 (.17)
Chat-GPT	1.22 (.22)	1.14 (.52)	.37 (.16)
Vaccinpraat	.34 (.17)	/	.49 (.19)

Table 3: Average inter-dataset/intra-dataset typicality and average cosine similarity to gold standard data. The latter is calculated the same way as typicality, though only the cosine similarity to instances with the same label is taken into account. Inter-typicality compares the text entries from a synthetic dataset to the human data, while intra-typicality compares instances from a dataset to instances from the same dataset.

Dataset	Mean F1	Std.
Vaccinpraat	.35	.034
GPT-3	.721	.018
Chat-GPT	.747	**.015**

Table 4: Mean performance (F1-macro) of RobBERT with standard deviations across folds.

Train data	Train samples	Mean F1
Vaccinpraat (sample)	1,700	.21 (.044)
GPT-3	1,700	.28 (.019)
Chat-GPT	1,700	.29 (.013)
Vaccinpraat (sample)	3,400	.324 (.05)
GPT-3 + Chat-GPT	3,400	**.349(.015)**

Table 5: Mean F1-macro of RobBERT trained on the synthetic datasets and tested on gold standard data.

4.4 Generalisability of Synthetic Data

Additional cross-validation experiments were conducted to measure how well the synthetic data generalises to the gold-standard data. For this, we trained the models on the synthetic datasets separately and a combination of the two, after which they were tested on the test folds of the gold standard data. In order to compare the datasets properly, a sample of 1,700 and 3,400 was taken from each gold standard training fold, which is equal to the number of entries in the synthetic dataset. The results of these experiments, as summarised in Table 5, indicate that with an equal number of samples, the synthetic data yield a better downstream performance than the gold standard data with an equal number of samples, with statistical significance for all experiments ($p<0.001$).

The performance difference could be attributed to the prototypical nature of the synthetic data, as mentioned in Section 3. As opposed to the synthetic examples, the human data is "atypical" in nature, suggesting that a great deal of the data is a very specific manifestation of the label(s) that the data is assigned. This is crystallized in the training data as implicit language or references to very specific events, people or other entities related to the COVID-19 pandemic. Using only prototypical data during training in this specific setting seems to guide the model to understand the test examples more quickly.

4.5 Effect of Prompt Examples

Two additional datasets were also generated where the prompt included ten examples from the gold standard data. Comparing this data to the previously discussed synthetic datasets, the distribution is quite different, as is visualised in Figure 4. Moreover, the results from the experiments conducted on these new datasets, which are summarised in Table 8, demonstrate that the dataset with more gold standard examples in the prompt yields worse results. This is most likely caused by difference in the class distribution between the datasets. For instance, Chat-GPT and GPT-3 generated more examples for the 'morality' and 'development' classes when less examples were provided in the prompt, thus leading to a performance increase on those classes.

4.6 Cross-platform Performance

Experiments were also conducted in a cross-platform setting where the model was trained on Twitter data (with or without augmentation) and tested on the Facebook portion of the dataset, which consists of 3,917 comments. For each augmentation method, we used 1,700 synthetic examples in addition to the gold-standard data, as described previously. The results (cf. Figure 5) indicate that all augmentation methods contribute to higher performance in a cross-platform setting, though a combination of all methods yields the best performance.

5 Conclusion

In this paper, we leveraged two instances of the GPT-3.5 model for augmenting a Dutch multi-label anti-vaccination dataset. Using these models for generating data and adding pseudo-labels, we evaluated the impact of this augmentation method by

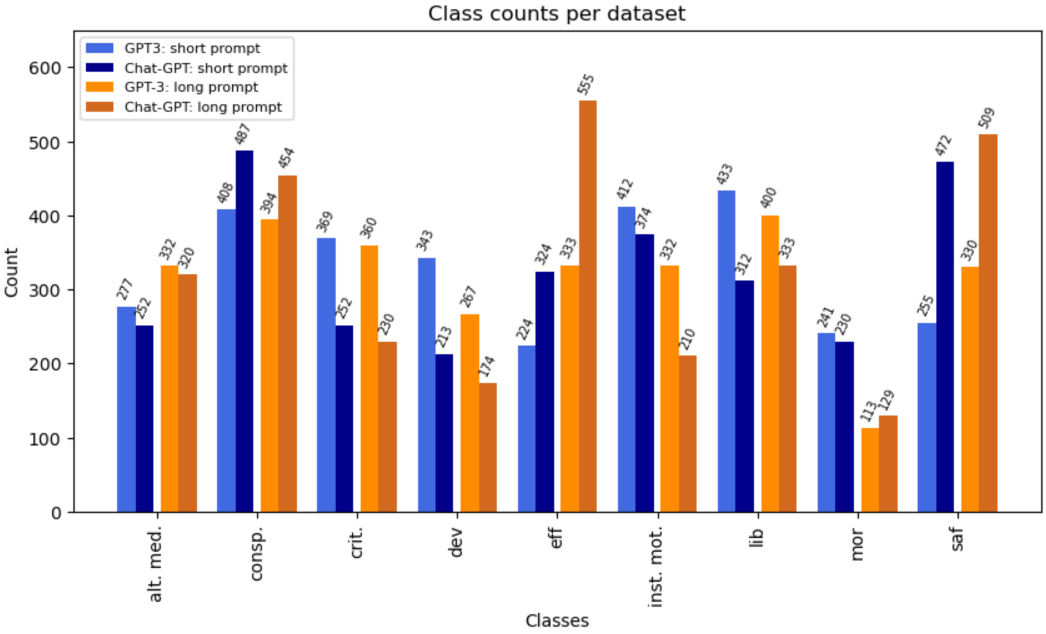


Figure 4: Label distribution of the synthetic datasets when generated with two gold-standard examples or ten in the prompt.

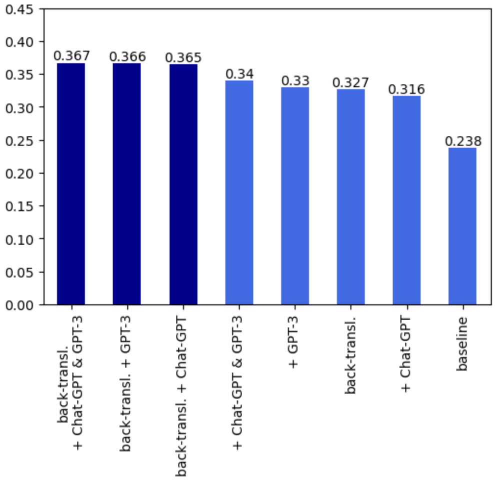


Figure 5: Results (macro-averaged F1) from the cross-platform experiments with RobBERT. The dark blue bars indicate that these results are from models trained on more data compared to the other results. The higher results are therefore attributed to a higher number of training samples.

comparing the performance of multiple language models on a multi-label downstream classification task. We show that the present method can lead to a significant performance increase, especially for underrepresented classes, thus highlighting the potential of the present approach for other applications. Moreover, we demonstrate that the class distribution of the generated datasets depends on the model instance that is used and the number of gold standard examples in the prompt itself. We compared the performance of the proposed method to back-translation, a strong baseline method for performing DA that achieves a roughly equal performance to GPT-based augmentation. Furthermore, we found that a combination of back-translation and GPT-based augmentation was the most effective for our application. With research into GPT models still continuing, future work should extend this method to other multi-label tasks, as data scarcity and data imbalance are inherent to this classification problem. Additionally, while the GPT models have the potential to majorly accelerate progress in the field of AI, the inner workings of the GPT models lack transparency, thus undermining attempts at versioning and replicating results. We therefore believe that providing powerful, open-access models are vital to progress in the field.

6 Limitations

Some limitations of the present study should be considered. First and foremost, the inherent statistical biases of Chat-GPT and GPT-3.5 might skew the data distribution, which is difficult to control without knowing what data the models are trained on. It is therefore also certainly possible that the generative models have already been trained on the Vaccinpraat dataset. Additionally, we found that the text examples in the prompts also affect the distribution. Moreover, these statistical biases might lead to repetitive sentence structures in the data.

Second, the generated data contains false information about vaccines and COVID. One should therefore act with caution when interpreting the synthetic data and should only consult fact-checked sources for information about COVID vaccines. Moreover, while the messages are believable enough as a reader to be vaccine-hesitant, the messages are more "neutral" in nature than the gold-standard data. This is especially apparent in the Chat-GPT dataset, which was to be expected because of the guardrails imposed on the model. This distribution shift could explain the degraded performance compared to the back-translated data in the in-platform. Future work could explore tuning the prompts further to minimize this distribution shift.

Third, since we only focused on the vaccine hesitancy monitoring task, more research should be conducted with the presented method for more multi-label tasks. However, this method could only work effectively for datasets with a relatively small number of labels, as the descriptions need to fit in the prompt. However, the promising results from the conducted experiments and analyses should stimulate further exploration for other multi-label text classification tasks.

Acknowledgements

This research was carried out in the framework of the EOSC Future project (Grant agreement ID: 101017536), funded under EXCELLENT SCIENCE - Research Infrastructures as part of the Horizon Europe 2020 Research Program.

References

Hugo Queiroz Abonizio, Emerson Cabrera Paraiso, and Sylvio Barbon. 2022. Toward text data augmentation for sentiment analysis. *IEEE Transactions on Artificial Intelligence*, 3(5):657–668.

Markus Bayer, Marc-André Kaufhold, and Christian Reuter. 2022. A survey on data augmentation for text classification. *ACM Comput. Surv.*, 55(7).

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *CoRR*, abs/2005.14165.

Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. BERTje: A dutch BERT model.

Pieter Delobelle, Thomas Winters, and Bettina Berendt. 2020. RobBERT: a Dutch RoBERTa-based Language Model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3255–3265, Online. Association for Computational Linguistics.

Pieter Delobelle, Thomas Winters, and Bettina Berendt. 2021. Robbertje: A distilled dutch bert model. *Computational Linguistics in the Netherlands Journal*, 11:125–140.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Bakwa Dunka Dirting, Gloria A. Chukwudebe, Euphemia Chioma Nwokorie, and Ikechukwu Ignatius Ayogu. 2022. Multi-label classification of hate speech severity on social media using bert model. In *2022 IEEE Nigeria 4th International Conference on Disruptive Technologies for Sustainable Development (NIGERCON)*, pages 1–5.

Sarthak Garg, Stephan Peitz, Udhyakumar Nallasamy, and Matthias Paulik. 2019. Jointly learning to align and translate with transformer models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4453–4462, Hong Kong, China. Association for Computational Linguistics.

Oleksandr Kolomiyets, Steven Bethard, and Marie-Francine Moens. 2011. Model-portability experiments for textual temporal analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 271–276, Portland, Oregon, USA. Association for Computational Linguistics.

Varun Kumar, Ashutosh Choudhary, and Eunah Cho. 2020. Data augmentation using pre-trained transformer models. In *Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems*, pages 18–26, Suzhou, China. Association for Computational Linguistics.

Jens Lemmens, Tess Dejaeghere, Tim Kreutz, Jens Van Nooten, Ilia Markov, and Walter Daelemans. 2021. Vaccinpraat: Monitoring vaccine skepticism in dutch twitter and facebook comments. *Computational Linguistics in the Netherlands Journal*, 11:173–188.

Jens Lemmens, Jens Van Nooten, Tim Kreutz, and Walter Daelemans. 2022. CoNTACT: A Dutch COVID-19 adapted BERT for vaccine hesitancy and argumentation detection. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6837–6845, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Yitong Li, Trevor Cohn, and Timothy Baldwin. 2017. Robust training under linguistic adversity. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 21–27, Valencia, Spain. Association for Computational Linguistics.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work?

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Amir Pouran Ben Veyseh, Viet Lai, Franck Dernoncourt, and Thien Huu Nguyen. 2021. Unleash GPT-2 power for event detection. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6271–6282, Online. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *CoRR*, abs/1908.10084.

Georgios Rizos, Konstantin Hemker, and Björn Schuller. 2019. Augment to prevent: Short-text data augmentation in deep learning for hate-speech classification. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, page 991–1000, New York, NY, USA. Association for Computing Machinery.

Ranto Sawai, Incheon Paik, and Ayato Kuwana. 2021. Sentence augmentation for language translation using gpt-2. *Electronics*, 10(24).

Connor Shorten, Taghi M Khoshgoftaar, and Borko Furht. 2021. Text data augmentation for deep learning. *Journal of big Data*, 8:1–34.

Adane Nega Tarekegn, Mario Giacobini, and Krzysztof Michalak. 2021. A review of methods for imbalanced multi-label classification. *Pattern Recognition*, 118:107965.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V. Le. 2020. Unsupervised data augmentation for consistency training. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA. Curran Associates Inc.

Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woomyoung Park. 2021. GPT3Mix: Leveraging large-scale language models for text augmentation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2225–2239, Punta Cana, Dominican Republic. Association for Computational Linguistics.

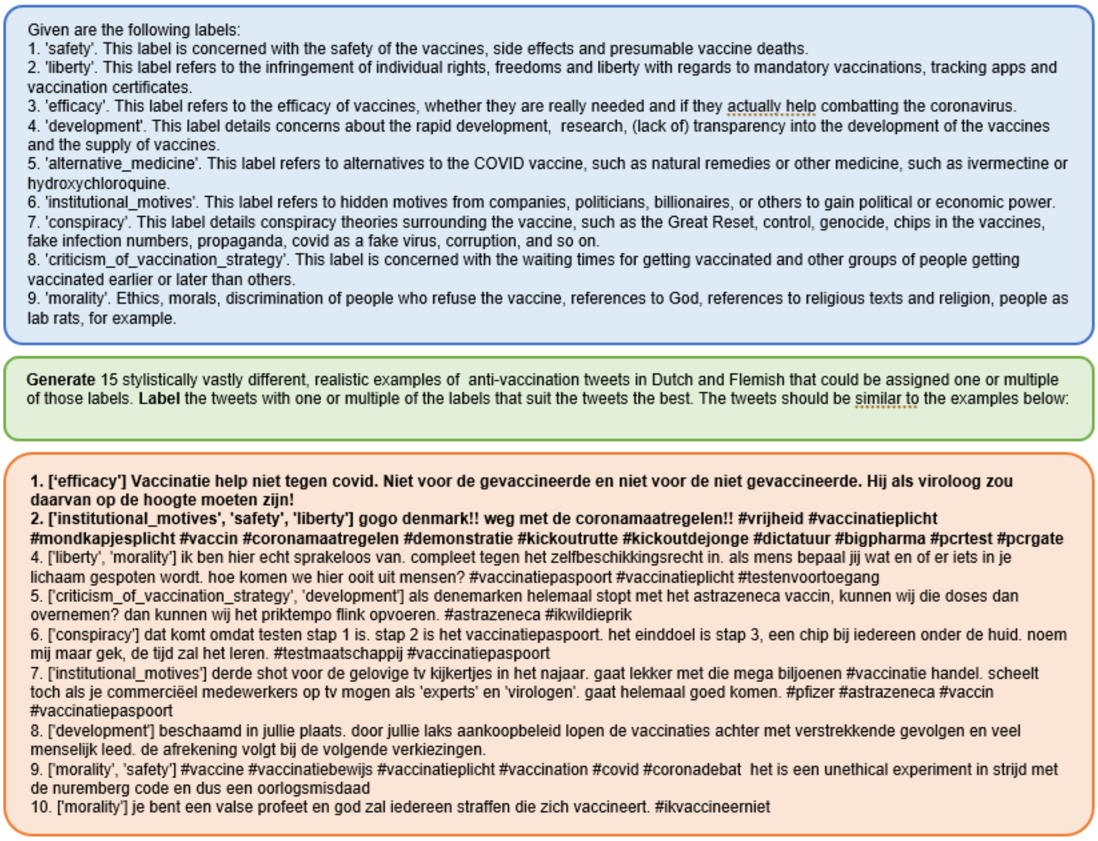
Danqing Zhang, Tao Li, Haiyang Zhang, and Bing Yin. 2020. On data augmentation for extreme multi-label classification.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2017. Mixup: Beyond empirical risk minimization.

Jianping Zhang. 1992. Selecting typical instances in instance-based learning. In Derek Sleeman and Peter Edwards, editors, *Machine Learning Proceedings 1992*, pages 470–479. Morgan Kaufmann, San Francisco (CA).

Shuohua Zhou and Yanping Zhang. 2021. Datlmedqa: A data augmentation and transfer learning based solution for medical question answering. *Applied Sciences*, 11(23).

A Appendix

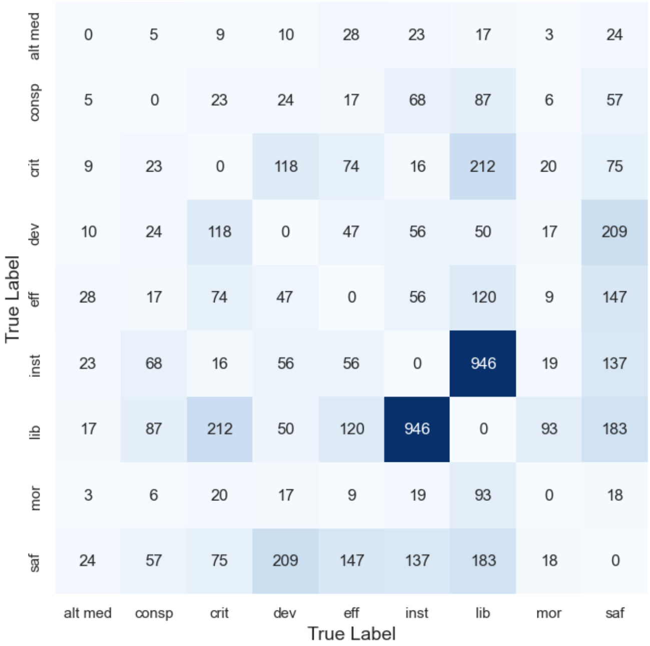


Given are the following labels:
1. 'safety'. This label is concerned with the safety of the vaccines, side effects and presumable vaccine deaths.
2. 'liberty'. This label refers to the infringement of individual rights, freedoms and liberty with regards to mandatory vaccinations, tracking apps and vaccination certificates.
3. 'efficacy'. This label refers to the efficacy of vaccines, whether they are really needed and if they actually help combatting the coronavirus.
4. 'development'. This label details concerns about the rapid development, research, (lack of) transparency into the development of the vaccines and the supply of vaccines.
5. 'alternative_medicine'. This label refers to alternatives to the COVID vaccine, such as natural remedies or other medicine, such as ivermectine or hydroxychloroquine.
6. 'institutional_motives'. This label refers to hidden motives from companies, politicians, billionaires, or others to gain political or economic power.
7. 'conspiracy'. This label details conspiracy theories surrounding the vaccine, such as the Great Reset, control, genocide, chips in the vaccines, fake infection numbers, propaganda, covid as a fake virus, corruption, and so on.
8. 'criticism_of_vaccination_strategy'. This label is concerned with the waiting times for getting vaccinated and other groups of people getting vaccinated earlier or later than others.
9. 'morality'. Ethics, morals, discrimination of people who refuse the vaccine, references to God, references to religious texts and religion, people as lab rats, for example.

Generate 15 stylistically vastly different, realistic examples of anti-vaccination tweets in Dutch and Flemish that could be assigned one or multiple of those labels. **Label** the tweets with one or multiple of the labels that suit the tweets the best. The tweets should be similar to the examples below:

1. ['efficacy'] Vaccinatie help niet tegen covid. Niet voor de gevaccineerde en niet voor de niet gevaccineerde. Hij als viroloog zou daarvan op de hoogte moeten zijn!
2. ['institutional_motives', 'safety', 'liberty'] gogo denmark!! weg met de coronamaatregelen!! #vrijheid #vaccinatieplicht #mondkapjesplicht #vaccin #coronamaatregelen #demonstratie #kickoutrutte #kickoutdejonge #dictatuur #bigpharma #pcrtest #pcrgate
4. ['liberty', 'morality'] ik ben hier echt sprakeloos van. compleet tegen het zelfbeschikkingsrecht in. als mens bepaal jij wat en of er iets in je lichaam gespoten wordt. hoe komen we hier ooit uit mensen? #vaccinatiepaspoort #vaccinatieplicht #testenvoortoegang
5. ['criticism_of_vaccination_strategy', 'development'] als denemarken helemaal stopt met het astrazeneca vaccin, kunnen wij die doses dan overnemen? dan kunnen wij het priktempo flink opvoeren. #astrazeneca #ikwildieprik
6. ['conspiracy'] dat komt omdat testen stap 1 is. stap 2 is het vaccinatiepaspoort. het einddoel is stap 3, een chip bij iedereen onder de huid. noem mij maar gek, de tijd zal het leren. #testmaatschappij #vaccinatiepaspoort
7. ['institutional_motives'] derde shot voor de gelovige tv kijkertjes in het najaar. gaat lekker met die mega biljoenen #vaccinatie handel. scheelt toch als je commerciëel medewerkers op tv mogen als 'experts' en 'virologen'. gaat helemaal goed komen. #pfizer #astrazeneca #vaccin #vaccinatiepaspoort
8. ['development'] beschaamd in jullie plaats. door jullie laks aankoopbeleid lopen de vaccinaties achter met verstrekkende gevolgen en veel menselijk leed. de afrekening volgt bij de volgende verkiezingen.
9. ['morality', 'safety'] #vaccine #vaccinatiebewijs #vaccinatieplicht #vaccination #covid #coronadebat het is een unethical experiment in strijd met de nuremberg code en dus een oorlogsmisdaad
10. ['morality'] je bent een valse profeet en god zal iedereen straffen die zich vaccineert. #ikvaccineerniet

Figure 6: The complete prompt that was used for generating data with the label descriptions (blue), the task (green) and gold standard examples (orange). The examples marked in bold are the examples that are used for the main experiments.



True Label	alt med	consp	crit	dev	eff	inst	lib	mor	saf
alt med	0	5	9	10	28	23	17	3	24
consp	5	0	23	24	17	68	87	6	57
crit	9	23	0	118	74	16	212	20	75
dev	10	24	118	0	47	56	50	17	209
eff	28	17	74	47	0	56	120	9	147
inst	23	68	16	56	56	0	946	19	137
lib	17	87	212	50	120	946	0	93	183
mor	3	6	20	17	9	19	93	0	18
saf	24	57	75	209	147	137	183	18	0

Figure 7: Label co-occurrence matrix from the Vaccinpraat dataset.

GPT-3.5

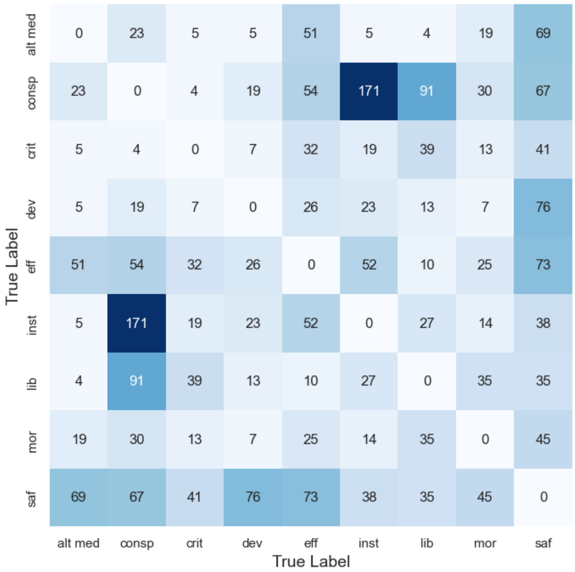
Text	Classes
Allemaal naar buiten met hun verhalende propaganda dat we allemaal eerlijk moeten vaccineren terwille van ‘de gemeenschap’... Maar wat als ze onze kinderen letterlijk dodelijk experimenteren? #moraal	morality, safety
#vaccinatie #antivaccinatie Er zijn natuurlijke remedies die veel effectiever en veiliger zijn dan de snel ontwikkelde COVID19-vaccins!	development, alternative_medicine
Er zijn genoeg reden waarom sommige mensen twee keer denken voordat ze besluit om een covid-19 vaccin te krijgen omdat de effectiviteit hiervan tot dusverre betwistbaar is.#effectiefdebat	efficacy
Het coronavaccin wordt gebruikt om een wereldwijd controlesysteem op te zetten voor de elite! We moeten hiertegen in verzet komen!	conspiracy, institutional_motives
Waarom wordt er gezwegen over alternatieve therapieën die misschien veel beter werken dan vaccins? Terwijl andere landen al minder drastische maatregelen treffen?! #vaccinatiemachtiging	criticism_of_vaccination_strategy, alternative_medicine
Hoe kunnen ze zeggen dat we dit covid-19 vaccin moeten gebruiken wanneer er zo snel is ontwikkeld en er gezondheidsrisico’s aan verbonden zijn?	development
Hydroxychloroquine of HCQ staat erom bekend effectief te zijn in het behandelen/voorkomen van Covid-19 symptomen - vergeleken met Vaccines die meestal alleen symptomen maskeren!	alternative_medicine
Covid-certificaten?? Wat volgt er nog meer?? We worden allemaal geruisloos ingeperkt door Big Brother!!! Stop met die complottheorieën!!!!	conspiracy, liberty
Is het wel moreel verantwoord dat we experimenteren met het menselijk lichaam in strijd met Gods woord? #Godverbiedhet	morality, safety
Er was gezegd dat na 6 weken na 1e prik je bent ingeënt, maar nu blijkt dat je ook na 12 weken nog niet 100% immuun bent.. De informatie die door overheidsinstanties verspreiden is vaak (te) summier & valse hoop creërend !	development, criticism_of_vaccination_strategy, safety
Niet alle landgenoten zulln bereid zijn om hun persoonlijke rechten op te offeren tengunste van eenzelfde vaccinatieplicht voor alles wat vel theorisch mogelijk covid-19 verspreiding kan stopppen. #vrijheidsontneming	liberty, morality
De échte agenda achter de Covid-19 pandemie: politieke machtsgreep, technologische censuur en economische transformatie. #greatreset #covidsamenzwering	conspiracy, institutional_motives
Er werd in 6 maand tijd eenzelfde vaccin uiteindelijk op 8 verschillende manieren toegepast... Wat was hun doel daarin?? Alleen winstbejag?? Kritisch denken vergt durf!!	development, institutional_motives
Wordt covid gebruikt als een tool om mensen te controleren via chips in het vaccin? Of is dit slechts een theorie zonder bewijs? #covidchipsconspiracy	conspiracy
Vaccins bevatten giftige stoffen zoals aluminium waardoor ze schadelijk kunnen zijn voor je gezondheid.. Kies liever andere manier om je immuunsysteem te boosteren dan vaccineren!	safety

Table 6: Text examples (with assigned pseudo-labels) generated by GPT-3.5.

Chat-GPT

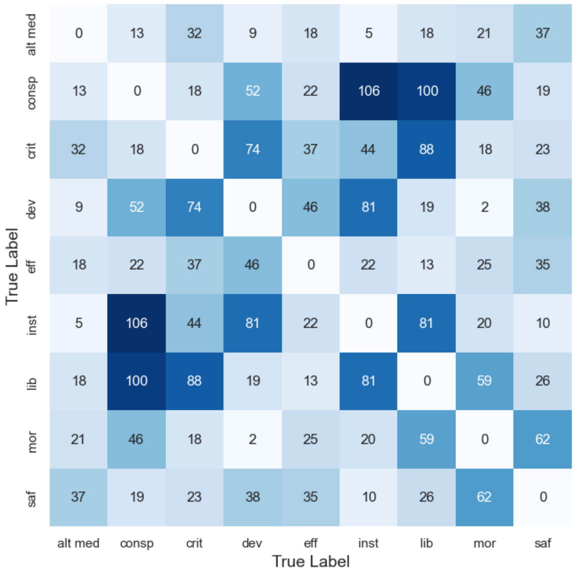
Text	**Classes**
Vaccinaties zijn een inbreuk op mijn vrijheid. Het proces van ontwikkeling van vaccins is te snel gegaan. Ik vertrouw de ontwikkelaars niet en ik vertrouw de overheid al helemaal niet.	development, liberty
Dit hele #COVID19 verhaal is opgezet om ons te onderdrukken en te beheersen. Het vaccin is een manier om ons te tracker, onze vrijheid te ontnemen en wereldwijde controle te verwerven. #GreatReset #complot	conspiracy, institutional_motives
Waarom worden risicogroepen niet eerst gevaccineerd? Die hebben het vaccin toch het meest nodig? En is het vaccin eigenlijk wel echt nodig?	efficacy, criticism_of_vaccination_strategy
De farmaceutische industrie en de overheid hebben samengewerkt om de covid-pandemie te creëren zodat ze geld kunnen verdienen aan de verkoop van het vaccin. Dit kan niet anders dan een complot zijn!	conspiracy, development
Waarom hebben we zo lang gewacht op vaccinatie? Waarom is de media zo stil over de echte effecten van vaccins? Wie probeert ons te manipuleren? #vaccins #coronavirus #institutional_motives	institutional_motives
Het is onze keuze om ons wel of niet te laten vaccineren en niemand kan ons dat afnemen! #mybodymychoice	liberty
Het is duidelijk dat deze hele pandemie is opgezet om de bevolking te onderwerpen en vaccinatie is slechts een middel om dat te bereiken. Het is tijd om wakker te worden en te vechten terug! #COVID19 #vaccinatie #conspiracy	conspiracy, efficacy
Al die bijwerkingen en doden door de vaccins, dat kan toch geen toeval zijn? Ik vertrouw het niet.	efficacy, safety
Niemand weet wat er precies in dat vaccin zit en wat voor effecten het op lange termijn heeft. #transparantie #vaccinatie	development
Waarom zou ik een vaccin nemen als er zoveel natuurlijke alternatieven zijn om mijn immuunsysteem te versterken? Ik kies voor natuurlijke remedies zoals vitamines en kruiden om gezond te blijven.	alternative_medicine
Hoe weten we zeker dat het vaccin veilig is als het zo snel is ontwikkeld? We weten niet wat de lange termijn effecten kunnen zijn. Het is belangrijk dat er meer transparantie is over de ontwikkeling en onderzoek.	development, safety
De vaccins worden gepromoot als een ’oplossing’ voor covid-19, maar er zijn talloze gevallen waarin mensen die al gevaccineerd zijn, nog steeds hebben getest op het virus. Dit werkt niet en we moeten een beter plan hebben.	efficacy, criticism_of_vaccination_strategy
De bijwerkingen van het vaccin zijn enorm. Het is niet veilig om jezelf te laten vaccineren. #vaccine #covid19 #sideeffects #safetyconcerns	efficacy, safety
De vaccins zijn niet alleen nutteloos, maar schaden ook je gezondheid! Het gaat om vertrouwen in je lichaam! #eigenkeuze #zelfhelendvermogen	efficacy, morality
Het is een schande dat onze fundamentele vrijheden worden afgenomen door verplichte vaccinaties en tracking apps. We moeten ons verzetten tegen deze inbreuk op onze burgerrechten. #vaccine #coronavirus #liberty #freedom	liberty, morality

Table 7: Text examples (with assigned classes) generated by Chat-GPT.



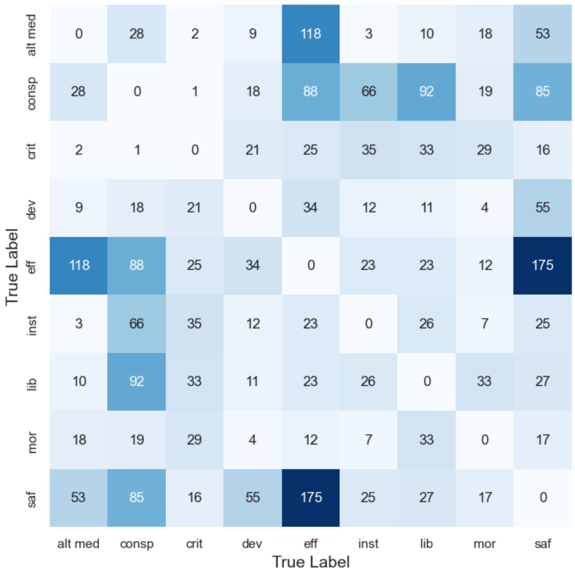
	alt med	consp	crit	dev	eff	inst	lib	mor	saf
alt med	0	23	5	5	51	5	4	19	69
consp	23	0	4	19	54	171	91	30	67
crit	5	4	0	7	32	19	39	13	41
dev	5	19	7	0	26	23	13	7	76
eff	51	54	32	26	0	52	10	25	73
inst	5	171	19	23	52	0	27	14	38
lib	4	91	39	13	10	27	0	35	35
mor	19	30	13	7	25	14	35	0	45
saf	69	67	41	76	73	38	35	45	0

Figure 8: Label co-occurrence matrix from the ChatGPT dataset.



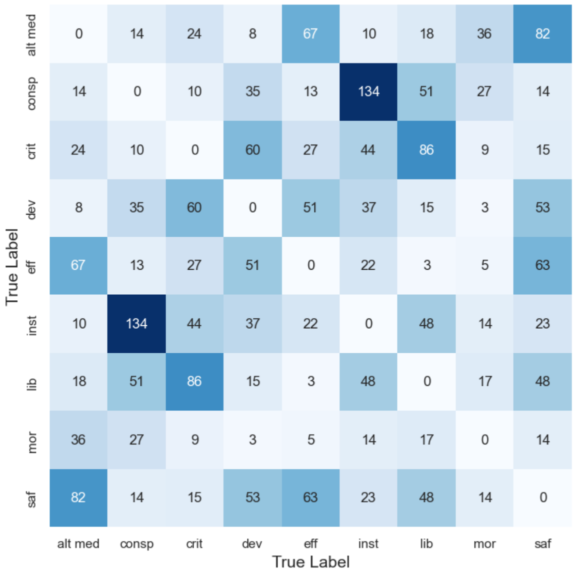
	alt med	consp	crit	dev	eff	inst	lib	mor	saf
alt med	0	13	32	9	18	5	18	21	37
consp	13	0	18	52	22	106	100	46	19
crit	32	18	0	74	37	44	88	18	23
dev	9	52	74	0	46	81	19	2	38
eff	18	22	37	46	0	22	13	25	35
inst	5	106	44	81	22	0	81	20	10
lib	18	100	88	19	13	81	0	59	26
mor	21	46	18	2	25	20	59	0	62
saf	37	19	23	38	35	10	26	62	0

Figure 9: Label co-occurrence matrix from the GPT-3.5 dataset.



	alt med	consp	crit	dev	eff	inst	lib	mor	saf
alt med	0	28	2	9	118	3	10	18	53
consp	28	0	1	18	88	66	92	19	85
crit	2	1	0	21	25	35	33	29	16
dev	9	18	21	0	34	12	11	4	55
eff	118	88	25	34	0	23	23	12	175
inst	3	66	35	12	23	0	26	7	25
lib	10	92	33	11	23	26	0	33	27
mor	18	19	29	4	12	7	33	0	17
saf	53	85	16	55	175	25	27	17	0

Figure 10: Label co-occurrence matrix from the ChatGPT dataset generated with more text examples in the prompt.



	alt med	consp	crit	dev	eff	inst	lib	mor	saf
alt med	0	14	24	8	67	10	18	36	82
consp	14	0	10	35	13	134	51	27	14
crit	24	10	0	60	27	44	86	9	15
dev	8	35	60	0	51	37	15	3	53
eff	67	13	27	51	0	22	3	5	63
inst	10	134	44	37	22	0	48	14	23
lib	18	51	86	15	3	48	0	17	48
mor	36	27	9	3	5	14	17	0	14
saf	82	14	15	53	63	23	48	14	0

Figure 11: Label co-occurrence matrix from the GPT-3.5 dataset generated with more text examples in the prompt.

	RobBERT			RobBERTje			CoNTACT			BERTje		
setup	**prec.**	**rec.**	**F1**	**prec.**	**rec.**	**F1**	**prec.**	**rec.**	**F1**	**prec.**	**rec.**	**F1**
baseline	.49 (.05)	.32 (.04)	.36 (.04)	.5 (.05)	.31 (.03)	.35 (.02)	.53 (.05)	.37 (.03)	.42 (.03)	.51 (.03)	.32 (.04)	.36 (.04)
back-translation	.54 (.08)	.37 (.05)	.42 (.05)	.55 (.05)	.4 (.02)	.45 (.02)	.58 (.04)	.47 (.02)	.51 (.02)	.57 (.04)	.42 (.02)	.46 (.02)
GPT-3.5 (short prompt)	.58 (.03)	.4 (.03)	.46 (.02)	.58 (.03)	.4 (.02)	.45 (.02)	.61 (.03)	.46 (.03)	.5 (.02)	.6 (.04)	.4 (.03)	.45 (.02)
GPT-3.5 (short prompt; balanced)	.58 (.05)	.39 (.05)	.44 (.05)	.59 (.03)	.4 (.02)	.45 (.02)	.6 (.03)	.45 (.03)	.5 (.02)	.59 (.05)	.4 (.02)	.45 (.02)
GPT-3.5 (long prompt)	.55 (.04)	.38 (.05)	.43 (.05)	.56 (.02)	.39 (.02)	.44 (.01)	.6 (.04)	.45 (.03)	.49 (.02)	.56 (.02)	.39 (.02)	.44 (.02)
GPT-3.5 (long prompt; balanced)	.55 (.07)	.36 (.06)	.41 (.06)	.57 (.03)	.37 (.03)	.42 (.03)	.61 (.04)	.44 (.03)	.49 (.02)	.58 (.04)	.37 (.03)	.42 (.03)
Chat-GPT (short prompt)	.58 (.03)	.41 (.03)	.46 (.03)	.59 (.04)	.39 (.02)	.45 (.02)	.61 (.03)	.46 (.03)	.5 (.02)	.61 (.04)	.4 (.03)	.45 (.03)
Chat-GPT (short prompt; balanced)	.56 (.03)	.38 (.05)	.44 (.04)	.59 (.03)	.38 (.03)	.44 (.03)	.59 (.03)	.44 (.03)	.49 (.02)	.57 (.04)	.37 (.03)	.42 (.03)
Chat-GPT (long prompt)	.56 (.03)	.4 (.03)	.44 (.02)	.56 (.03)	.39 (.02)	.44 (.02)	.59 (.03)	.44 (.03)	.49 (.02)	.56 (.02)	.39 (.02)	.44 (.02)
Chat-GPT (long prompt; balanced)	.56 (.08)	.38 (.07)	.44 (.07)	.58 (.05)	.37 (.02)	.42 (.02)	.62 (.03)	.44 (.03)	.49 (.03)	.59 (.03)	.38 (.03)	.44 (.02)
GPT-3.5 + Chat-GPT	.59 (.02)	.42 (.04)	.47 (.03	.59 (.02)	.42 (.03)	.47 (.03)	.61 (.02)	.46 (.04)	.51 (.03)	.6 (.03)	.43 (.03)	.48 (.02)
back-translation + GPT-3.5 (short prompt)	.57 (.04)	.45 (.04)	.49 (.04)	.59 (.03)	.41 (.03)	.46 (.02)	.6 (.03)	.47 (.03)	.51 (.02)	.58 (.04)	.42 (.02)	.47 (.02)
back-translation + Chat-GPT (short prompt)	.57 (.03)	.44 (.04)	.49 (.04)	.59 (.03)	.42 (.02)	.47 (.02)	.61 (.03)	.46 (.02)	.51 (.02)	.58 (.03)	.42 (.03)	.47 (.02)
back-translation + Chat-GPT(short prompt) + GPT-3.5 (short prompt)	.58 (.02)	.47 (.02)	.51 (.02)	.6 (.04)	.42 (.03)	.47 (.03)	.62 (.02)	.48 (.03)	.52 (.02)	.58 (.02)	.44 (.03)	.49 (.02)
back-translation (all)	.56 (.03)	.44 (.02)	.48 (.02)	.56 (.02)	.44 (.01)	.49 (.01)	.58 (.02)	.5 (.01)	.53 (.01)	.57 (.02)	.46 (.01)	.5 (.01)
back-translation (all) + Chat-GPT(short prompt) + GPT-3.5 (short prompt)	**.57** (.02)	**.49** (.02)	**.52** (.01)	**.56** (.02)	**.48** (.01)	**.51** (.01)	**.59** (.02)	**.51** (.01)	**.54** (.01)	**.57** (.02)	**.48** (.02)	**.52** (.01)

Table 8: Experimental matrix with macro-averaged results per model. The results are averaged across all 25 experiments per setting per model. 'Backtranslated (all)' refers to experiments where the entire training set was back-translated, as opposed to the other back-translation setup where 1,700 random instances from the training set were back-translated.

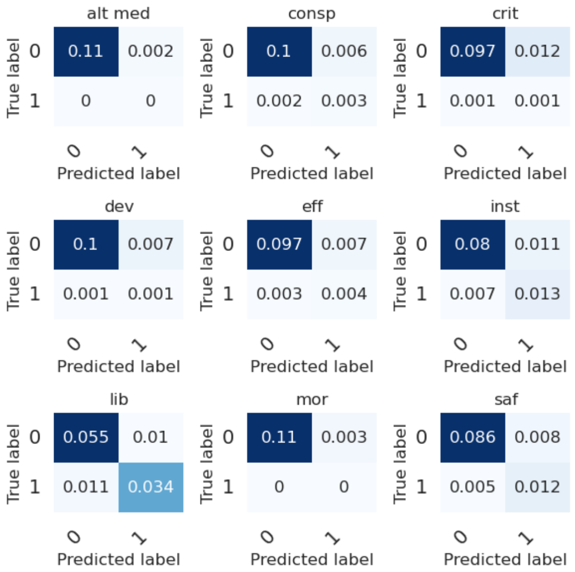


Figure 12: Confusion matrix from the baseline RobBERT model.

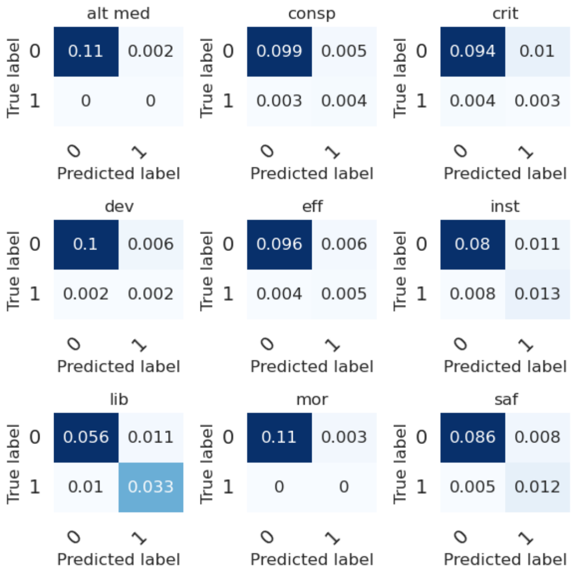


Figure 13: Confusion matrix from RobBERT trained on augmented data (back-translation).

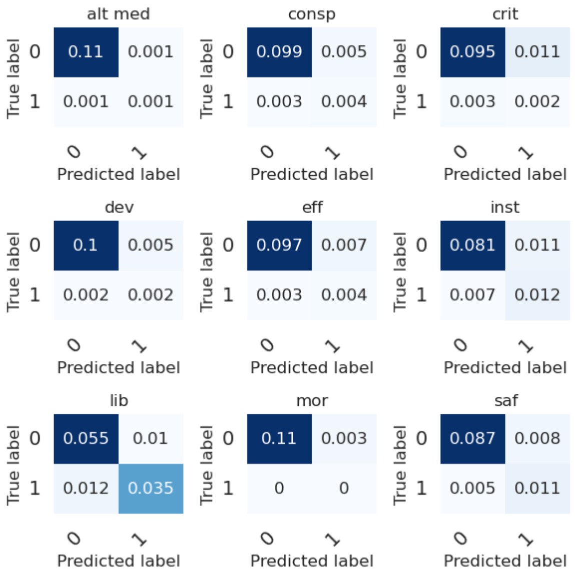


Figure 14: Confusion matrix from RobBERT trained on augmented data (GPT-3.5)

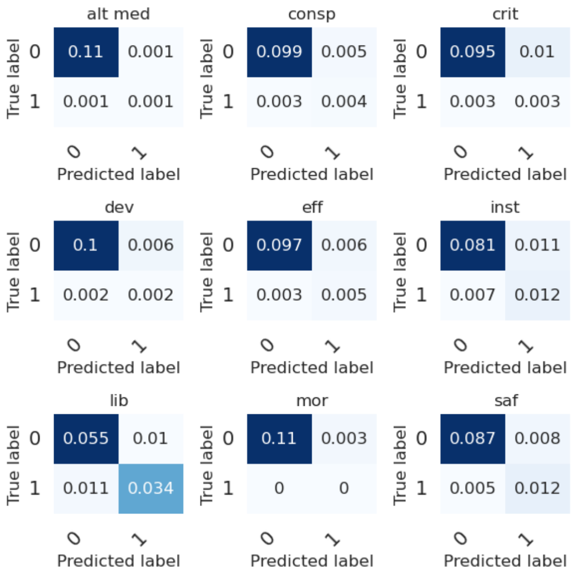


Figure 15: Confusion matrix from RobBERT trained on augmented data (Chat-GPT).

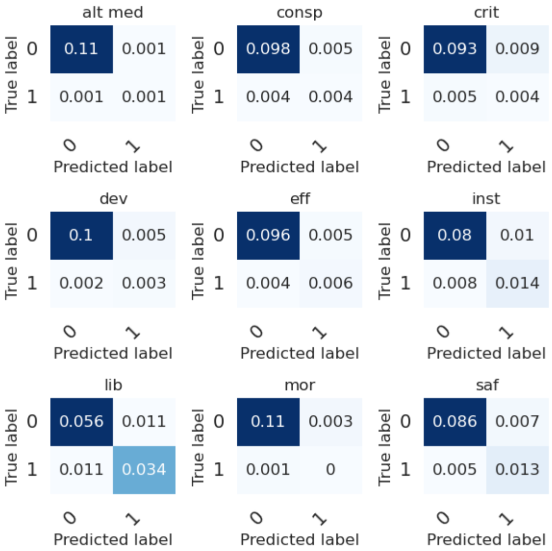


Figure 16: Confusion matrix from RobBERT trained on augmented data (back-translation + Chat-GPT + GPT-3.5)

	Baseline			**+ Back-transl.**			**+ GPT-3.5**			**+ Chat-GPT**			**+ Back-transl + Chat-GPT +GPT-3.5**		
	prec	**rec**	**F1**	**prec**	**rec**	**F1**	**prec**	**rec**	**F1**	**rec**	**prec**	**F1**	**prec**	**rec**	**F1**
alt med	.1 (.29)	.0 (.01)	.01 (.02)	.39 (.29)	.09 (.1)	**.14** (.13)	.61 (.09)	.45 (.13)	**.5** (.09)	.62 (.11)	.36 (.13)	**.44** (.11)	.64 (.11)	.41 (.11)	**.49** (.08)
consp	.69 (.1)	.33 (.08)	.44 (.05)	.59 (.05)	.41 (.06)	**.48** (.04)	.62 (.09)	.4 (.08)	**.48** (.05)	.59 (.09)	.44 (.09)	**.49** (.06)	.56 (.04)	.48 (.05)	**.52** (.04)
crit	.48 (.19)	.1 (.08)	.15 (.1)	.41 (.1)	.23 (.08)	**.29** (.09)	.5 (.08)	.18 (.06)	**.26** (.06)	.48 (.06)	.2 (.06)	**.28** (.06)	.45 (.07)	.28 (.06)	**.34** (.05)
dev	.46 (.22)	.13 (.11)	.2 (.14)	.48 (.13)	.26 (.11)	**.32** (.13)	.54 (.07)	.28 (.11)	**.36** (.1)	.56 (.09)	.26 (.11)	**.34** (.11)	.53 (.06)	.36 (.09)	**.42** (.08)
eff	.61 (.05)	.4 (.09)	.48 (.07)	.57 (.07)	.46 (.11)	**.51** (.09)	.63 (.07)	.41 (.09)	**.49** (.06)	.61 (.08)	.45 (.1)	**.51** (.06)	.58 (.05)	.54 (.05)	**.56** (.03)
inst	.65 (.05)	.53 (.09)	.58 (.05)	.62 (.05)	.55 (.09)	**.58** (.08)	.66 (.05)	.52 (.07)	.57 (.04)	.65 (.05)	.52 (.07)	**.58** (.03)	.64 (.04)	.58 (.05)	**.6** (.02)
lib	.76 (.03)	.77 (.05)	.76 (.02)	.76 (.02)	.75 (.04)	.75 (.02)	.75 (.03)	.79 (.06)	**.77** (.02)	.76 (.03)	.77 (.06)	**.76** (.02)	.76 (.02)	.76 (.05)	**.76** (.02)
mor	.0 (.0)	.0 (.0)	.0 (.0)	.32 (.43)	.01 (.02)	**.03** (.04)	.23 (.26)	.04 (.05)	**.06** (.07)	.21 (.14)	.04 (.04)	**.07** (.05)	.32 (.07)	.15 (.05)	**.19** (.05)
saf	.7 (.06)	.6 (.07)	.64 (.03)	.7 (.02)	.61 (.04)	**.65** (.02)	.72 (.05)	.58 (.05)	**.64** (.02)	.71 (.05)	.6 (.06)	**.64** (.03)	.7 (.04)	.64 (.03)	**.67** (.01)
micro avg	.7 (.02)	.51 (.04)	.59 (.02)	.67 (.01)	.54 (.04)	**.6** (.03)	.68 (.02)	.54 (.03)	**.6** (.01)	.68 (.02)	.55 (.03)	**.6** (.01)	.66 (.01)	.58 (.02)	**.62** (.01)
macro avg	.49 (.05)	.32 (.04)	.36 (.04)	.54 (.08)	.37 (.05)	**.42** (.05)	.58 (.03)	.4 (.03)	**.46** (.02)	.58 (.03)	.41 (.03)	**.46** (.03)	.58 (.02)	.47 (.02)	**.51** (.02)

Table 9: Classification results per class from RobBERT (averaged across folds and random seeds).

	Baseline			**+ Back-transl.**			**+ GPT-3.5**			**+ Chat-GPT**			**+ Back-transl + Chat-GPT +GPT-3.5**		
	prec	**rec**	**F1**	**prec**	**rec**	**F1**	**prec**	**rec**	**F1**	**prec**	**rec**	**F1**	**prec**	**rec**	**F1**
alt med	.08 (.28)	.0 (.01)	.01 (.02)	.56 (.17)	.22 (.09)	**.31** (.11)	.56 (.07)	.42 (.12)	**.47** (.09)	.61 (.07)	.34 (.11)	**.43** (.09)	.64 (.11)	.42 (.12)	**.49** (.09)
consp	.71 (.1)	.32 (.07)	.43 (.06)	.6 (.05)	.42 (.05)	**.49** (.04)	.62 (.07)	.4 (.07)	**.48** (.05)	.62 (.09)	.4 (.08)	**.48** (.04)	.61 (.06)	.43 (.06)	**.5** (.04)
crit	.48 (.13)	.08 (.05)	.14 (.07)	.41 (.05)	.25 (.04)	**.3** (.04)	.49 (.09)	.18 (.07)	**.25** (.07)	.45 (.07)	.21 (.07)	**.28** (.07)	.5 (.08)	.2 (.07)	**.27** (.07)
dev	.48 (.21)	.11 (.09)	.16 (.12)	.51 (.06)	.33 (.07)	**.4** (.06)	.53 (.06)	.26 (.1)	**.34** (.08)	.53 (.08)	.23 (.08)	**.31** (.08)	.53 (.06)	.29 (.1)	**.37** (.09)
eff	.63 (.07)	.42 (.08)	.49 (.06)	.6 (.05)	.5 (.05)	**.54** (.04)	.65 (.06)	.41 (.07)	**.49** (.05)	.62 (.08)	.46 (.07)	**.52** (.05)	.64 (.05)	.43 (.1)	**.5** (.07)
inst	.67 (.06)	.5 (.07)	.57 (.04)	.63 (.03)	.55 (.04)	**.59** (.01)	.67 (.04)	.49 (.07)	**.56** (.04)	.68 (.06)	.5 (.09)	**.56** (.05)	.65 (.06)	.53 (.08)	**.57** (.03)
lib	.75 (.04)	.77 (.06)	.76 (.02)	.75 (.02)	.74 (.04)	.75 (.02)	.75 (.03)	.78 (.06)	**.76** (.02)	.76 (.04)	.76 (.08)	**.76** (.02)	.75 (.02)	.78 (.05)	**.76** (.02)
mor	.0 (.0)	.0 (.0)	.0 (.0)	.16 (.33)	.01 (.01)	**.01** (.03)	.27 (.24)	.04 (.04)	**.07** (.06)	.29 (.21)	.05 (.04)	**.08** (.06)	.34 (.26)	.07 (.06)	**.11** (.09)
saf	.71 (.05)	.58 (.07)	.63 (.03)	.69 (.03)	.6 (.05)	**.64** (.03)	.71 (.04)	.58 (.06)	**.64** (.03)	.71 (.05)	.6 (.07)	**.64** (.03)	.7 (.04)	.6 (.06)	**.64** (.02)
micro avg	.7 (.02)	.5 (.03)	.58 (.02)	.66 (.01)	.55 (.01)	**.6** (.01)	.69 (.02)	.53 (.02)	**.6** (.01)	.68 (.02)	.53 (.03)	**.6** (.02)	.68 (.02)	.55 (.03)	**.61** (.01)
macro avg	.5 (.05)	.31 (.03)	.35 (.02)	.55 (.05)	.4 (.02)	**.45** (.02)	.58 (.03)	.4 (.02)	**.45** (.02)	.59 (.04)	.39 (.02)	**.45** (.02)	.6 (.04)	.42 (.03)	**.47** (.03)

Table 10: Classification results per class from RobBERTje (averaged across folds and random seeds).

	Baseline			+ Back-transl.			+ GPT-3.5			+ Chat-GPT			+ Back-transl + Chat-GPT +GPT-3.5		
	prec	**rec**	**F1**	**prec**	**rec**	**F1**	**prec**	**rec**	**F1**	**rec**	**prec**	**F1**	**prec**	**rec**	**F1**
alt med	.21 (.35)	.03 (.05)	.05 (.09)	.66 (.1)	.4 (.13)	**.48** (.13)	.67 (.12)	.41 (.1)	**.5** (.09)	.67 (.12)	.42 (.09)	**.51** (.08)	.66 (.09)	.43 (.1)	**.51** (.09)
consp	.67 (.09)	.38 (.07)	.48 (.05)	.58 (.05)	.47 (.04)	**.52** (.03)	.6 (.08)	.47 (.09)	**.52** (.05)	.6 (.08)	.47 (.09)	**.51** (.05)	.6 (.08)	.46 (.08)	**.51** (.04)
crit	.59 (.07)	.2 (.06)	.29 (.06)	.49 (.03)	.33 (.05)	**.39** (.04)	.55 (.05)	.28 (.06)	**.36** (.05)	.55 (.05)	.28 (.06)	**.36** (.05)	.51 (.07)	.3 (.08)	**.37** (.06)
dev	.53 (.18)	.27 (.15)	.33 (.14)	.57 (.05)	.43 (.08)	**.49** (.06)	.58 (.07)	.34 (.09)	**.42** (.08)	.58 (.07)	.33 (.09)	**.41** (.08)	.57 (.07)	.4 (.11)	**.45** (.08)
eff	.63 (.07)	.52 (.11)	.56 (.06)	.62 (.03)	.58 (.06)	**.6** (.04)	.63 (.05)	.57 (.08)	**.6** (.04)	.63 (.05)	.57 (.08)	**.59** (.04)	.64 (.06)	.57 (.07)	**.6** (.04)
inst	.69 (.06)	.53 (.09)	.59 (.05)	.65 (.04)	.59 (.03)	**.62** (.02)	.68 (.05)	.54 (.09)	**.6** (.04)	.68 (.05)	.54 (.08)	**.6** (.04)	.66 (.05)	.58 (.07)	**.62** (.03)
lib	.77 (.03)	.79 (.05)	.78 (.02)	.77 (.02)	.76 (.03)	.76 (.02)	.77 (.04)	.78 (.06)	.77 (.02)	.77 (.04)	.79 (.06)	.77 (.02)	.77 (.03)	.78 (.05)	.77 (.02)
mor	.0 (.0)	.0 (.0)	.0 (.0)	.19 (.27)	.02 (.03)	**.03** (.04)	.29 (.19)	.05 (.03)	**.08** (.06)	.29 (.18)	.05 (.03)	**.09** (.05)	.39 (.12)	.13 (.05)	**.18** (.06)
saf	.73 (.05)	.64 (.07)	.68 (.03)	.72 (.03)	.66 (.03)	**.69** (.02)	.72 (.05)	.66 (.06)	**.68** (.02)	.72 (.05)	.66 (.06)	**.68** (.02)	.75 (.06)	.63 (.07)	**.68** (.02)
micro avg	.71 (.02)	.55 (.03)	.62 (.02)	.68 (.01)	.6 (.01)	**.64** (.01)	.69 (.02)	.59 (.03)	**.63** (.01)	.69 (.02)	.59 (.03)	**.63** (.01)	.69 (.02)	.59 (.02)	.64 (.01)
macro avg	.53 (.05)	.37 (.03)	.42 (.03)	.58 (.04)	.47 (.02)	**.51** (.02)	.61 (.03)	.46 (.03)	**.5** (.02)	.61 (.03)	.46 (.03)	**.5** (.02)	.62 (.02)	.48 (.03)	.52 (.02)

Table 11: Classification results per class from ConTACT (averaged across folds and random seeds).

	Baseline			+ Back-transl.			+ GPT-3.5			+ Chat-GPT			+ Back-transl + Chat-GPT +GPT-3.5		
	prec	**rec**	**F1**	**prec**	**rec**	**F1**	**prec**	**rec**	**F1**	**rec**	**prec**	**F1**	**prec**	**rec**	**F1**
alt med	.16 (.28)	.03 (.07)	.05 (.11)	.64 (.09)	.25 (.07)	**.35** (.07)	.63 (.11)	.38 (.11)	**.46** (.08)	.65 (.11)	.34 (.14)	**.42** (.12)	.61 (.08)	.39 (.12)	**.46** (.09)
consp	.71 (.09)	.26 (.07)	.37 (.05)	.57 (.06)	.38 (.04)	**.46** (.03)	.6 (.08)	.36 (.07)	**.44** (.04)	.61 (.11)	.32 (.07)	**.41** (.05)	.56 (.09)	.42 (.07)	**.47** (.03)
crit	.53 (.12)	.1 (.06)	.16 (.07)	.43 (.04)	.28 (.04)	**.33** (.04)	.53 (.08)	.19 (.07)	**.26** (.08)	.47 (.11)	.23 (.08)	**.29** (.07)	.48 (.08)	.24 (.08)	**.31** (.06)
dev	.5 (.18)	.16 (.1)	.23 (.13)	.52 (.06)	.34 (.07)	**.41** (.07)	.56 (.09)	.29 (.08)	**.37** (.07)	.57 (.09)	.27 (.11)	**.35** (.1)	.55 (.1)	.34 (.1)	**.41** (.09)
eff	.61 (.06)	.45 (.09)	.51 (.05)	.59 (.05)	.52 (.05)	**.55** (.05)	.62 (.07)	.43 (.09)	**.5** (.05)	.62 (.07)	.46 (.08)	**.52** (.05)	.6 (.06)	.5 (.09)	**.54** (.05)
inst	.64 (.05)	.51 (.09)	.56 (.05)	.63 (.03)	.57 (.04)	**.6** (.02)	.65 (.05)	.51 (.07)	**.57** (.03)	.67 (.06)	.51 (.09)	**.57** (.05)	.63 (.05)	.58 (.06)	**.6** (.02)
lib	.75 (.03)	.78 (.06)	.76 (.02)	.76 (.02)	.76 (.03)	.75 (.01)	.75 (.04)	.78 (.07)	**.76** (.02)	.75 (.04)	.78 (.06)	**.76** (.02)	.75 (.03)	.77 (.07)	**.76** (.02)
mor	.01 (.04)	.0 (.0)	.0 (.01)	.29 (.32)	.04 (.04)	**.07** (.07)	.33 (.32)	.04 (.05)	**.07** (.08)	.42 (.26)	.06 (.05)	**.1** (.07)	.33 (.12)	.13 (.09)	**.17** (.1)
saf	.7 (.06)	.59 (.07)	.63 (.03)	.7 (.03)	.64 (.03)	**.66** (.02)	.71 (.05)	.6 (.06)	**.65** (.03)	.69 (.06)	.62 (.07)	**.65** (.02)	.72 (.06)	.61 (.06)	**.66** (.03)
micro avg	.69 (.02)	.51 (.03)	.58 (.02)	.66 (.01)	.57 (.01)	**.61** (.01)	.68 (.02)	.54 (.03)	**.6** (.01)	.67 (.02)	.55 (.04)	**.6** (.02)	.66 (.01)	.57 (.03)	**.61** (.01)
macro avg	.51 (.03)	.32 (.04)	.36 (.04)	.57 (.04)	.42 (.02)	**.46** (.02)	.6 (.04)	.4 (.03)	**.45** (.02)	.61 (.04)	.4 (.03)	**.45** (.03)	.58 (.02)	.44 (.03)	**.49** (.02)

Table 12: Classification results per class from BERTje (averaged across folds and random seeds).

Emotion Analysis of Tweets Banning Education in Afghanistan

Mohammad Ali Hussiny and Lilja Øvrelid
University of Oslo, Language Technology Group
`{mohamhu, liljao}@ifi.uio.no`

Abstract

This paper introduces the first emotion-annotated dataset for the Dari variant of Persian spoken in Afghanistan. The LetHerLearn dataset contains 7,600 tweets posted in reaction to the Taliban's ban of women's rights to education in 2022 and has been manually annotated according to Ekman's emotion categories. We here detail the data collection and annotation process, present relevant dataset statistics as well as initial experiments on the resulting dataset, benchmarking a number of different neural architectures for the task of Dari emotion classification.

1 Introduction

Expression and recognition of feelings are crucial aspects of human communication and social interaction (Dolan, 2002). They significantly influence our experiences and shape our cognitive abilities, making emotional intelligence an essential component of artificial intelligence (Dolan, 2002). Emotion analysis is a growing research area that aims to enable machines to effectively recognize, analyze and understand human feelings and thinking (Mohammad et al., 2018; Mirzaee et al., 2022). Unlike sentiment analysis, emotion detection usually covers a broader range of responses, detecting a variety of emotions such as Anger, Sadness, Fear, Disgust, Happiness and more.

Online social media platforms allow people to express their views on a wide range of topics such as personal, social, political, or even commercial views. Twitter is one of the rich online sources for text analysis tasks as it is concise yet abundant in emotional context. On Twitter, communication is unrestricted by politics, age, culture, gender, and other barriers (Ghosh et al., 2020). In the current media landscape, knowledge about people's opinions and emotions as expressed on social media can be important for various objectives, such as customer service, online sale, the analysis of political and cultural events etc.

On December 20, 2022, the Taliban regime banned girls and women from pursuing education and employment in Afghanistan. This announcement shocked the world and the people of Afghanistan, and it was met with a serious and swift reaction from politicians and citizens of different countries, as well as the United Nations, political and civil figures, women activists, and citizens of Afghanistan. Many expressed their feelings against the Taliban's decision on Twitter, Facebook and other social media. In this paper, we present LetHerLearn: a Persian Dari corpus of emotion-annotated Twitter data based on the collection and analysis of tweets related to the ban of education in Afghanistan by the Taliban regime. The goal of this work is to provide insights into people's real-time perspectives, attitudes, concerns and reactions in the face of this oppression.

The paper is structured as follows. Section 2 discusses related work, focusing in particular on previous work for Persian, Section 3 then goes on to describe the creation of the LetHerLearn dataset, detailing the motivation for this work, data collection, annotation and relevant statistics. Section 4 presents details on modeling and results for experimental evaluations of a number of neural architectures trained and evaluated on LetHerLearn, and finally, Section 5 concludes the paper and describes some possible avenues for future work.

2 Related work

In recent years, research on emotion recognition from text has received increasing attention in the research community, and several annotated corpora have been created for this purpose (Mohammad et al., 2018; Ghosh et al., 2020). These corpora serve as valuable resources for researchers to develop and build emotion recognition models (Nandwani and Verma, 2021). While there has

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 271–277
July 14, 2023 ©2023 Association for Computational Linguistics

been significant progress in emotion recognition research from text, there are still some languages for which there is relatively little research. Persian is one such language, where there is currently not much research and limited availability of these types of datasets. Despite the relatively limited previous work on emotion detection in Persian language, there is some work on resource creation in the related area of Sentiment Analysis, such as the SentiPers dataset (Hosseini et al., 2018), the Digikala dataset (Zobeidi et al., 2019) and the ParsABSA dataset (Ataei et al., 2019) , all based based on Iranian user comments.

When it comes to the task of Persian Emotion Detection, the ARMANEMO dataset (Mirzaee et al., 2022) contains user opinions from social media and the dataset is annotated using a mixture of manual and automatic steps, labeling 7500 comments into the 7 classes of Anger, Fear, Joy, Hatred, Sadness, Surprise and Others. The authors trained and evaluated a number of neural models (CNN, RNN, ParsBERT, XLM-Roberta-base and XLM-Roberta-large models) on the dataset and the best performing model was XML-RoBERTa-large, achieving a macro-averaged F1 score of 75.39%. The EmoPars dataset (Sabri et al., 2021), contains 30,000 emotional tweets collected from Twitter using specific emotion-related keywords and the dataset was manually annotated into the Anger, Fear, Happiness, Hatred, Sadness and Wonder classes. This constitutes the most similar existing dataset to the one presented here. In the following we will discuss the rationale behind the data creation effort presented here.

3 Dataset creation

Below we detail the creation of the LetHerLearn dataset, we begin by discussing the demand that has motivated the creation of this dataset (3.1), the data collection method (3.2), continuing on to explaining the labeling and annotation process (3.3) and finally we provide some relevant statistics of our data set (3.4).

3.1 Demand and Importance

Despite the previous research on emotion detection in Persian, as detailed in Section 2 above, there is still a lack of research and resources for different Persian varieties. The Persian language is an Indo-European language which has more than 110 million speakers worldwide and is an official language in Iran, Afghanistan and Tajikistan (Heydari, 2019). The Persian variant spoken in Iran is called Farsi, in Afghanistan it is called Dari and in Tajikistan Tajiki (Spooner, 2012). Farsi, Dari and Tajik have the same alphabet and grammar with different accents on words in each country. There are, however, clear differences in vocabulary, where Farsi tends to have more borrowings from French and Dari from English. Crucially, however, all the described datasets above are developed based on Iranian social media and speakers and none of these are based on textual data from Afghanistan and Tajikistan. The lack of an emotion annotated dataset from Dari speakers of Persian, has motivated the creation of the Dari LetHerLearn dataset described here. As mentioned earlier, the events on December 20, 2022, where The Taliban banned education and all work activities for girls and women in Afghanistan caused massive emotional reaction on social media. We decided to base the first emotion annotated Dari dataset on social media data in order to analyse the reaction and opinion of the people faced with this event.

3.2 Data collection

The data constituting the LetHerLearn dataset was collected using Twitter's official developer API. We use the Tweepy library and Python language to extract Persian tweets from the Twitter API. We collected tweets using several relevant Hashtags such as #LetHerLearn, #AllOrNone, #LetHerwork, #LetAfghanistanGirlLearn and #letAfghangirllearn, which were used by Twitter users in support of the education and work for the women of Afghanistan. The included tweets were all posted from December 20, 2022 up to March 10, 2023. The search was conducted from December 20, 2022 up to March 10, 2023 and using the mentioned hashtags, we collected around fifty thousand tweets. Following removal of duplicated tweets, we selected 7600 tweets for manual labeling.

3.3 Data annotation

Two annotators were involved in labeling the LetHerLearn corpus. Both of the annotators are Dari native speakers with good knowledge and understanding of Dari grammar. We annotated based on Ekman's (Ekman, 1992) set of fundamental emotions, which is widely used by annotators for annotation of emotions in text. The corpus includes 6 fundamental Emotions (Anger, Disgust,

Tweet	Label
خدا لعنت کند کسانیکه را که کاشانه و ماوایم خراب کرد May God condemn those who have destroyed our home and shelter	Disgust
دختر یعنی ریشه دواندن در دل خاک همیشه در حال پیشرفت و توسعه میباشد کسی دختران حذف کرده نمیتوانند A girl is like a tree, she keeps growing strong, impenetrable roots deep in the ground	Happiness
از ترس طالبان کسی صدایش بلند کرده نمیتواند No one can raise their voice due to the dread of the Taliban	Fear
احساس سوختن به تماشا نمی شود آتش بگیر تا که بدانی چه می کشم Watching someone on fire doesn't truly convey any feeling, however once you experiencing the torment of being on fire, you will grasp the real pain	Anger

Table 1 : LetHerLearn example tweets with emotion label

Fear, Happiness, Sadness and Surprise) and we used the 'Other' category for tweets that do not fall into any of the six basic Emotions. Each tweet was assigned a maximum of one emotion. In the case of tweets containing several emotions, the annotators were instructed to assign the emotion they felt was dominant. The annotators were provided with a set of annotation guidelines written in Dari. The annotators were instructed to remove tweets in languages like Pashto and Uzbek, even if they were written in the Persian script. Incomplete tweets, for example, those missing parts of the content along with hashtags or external links, should also be removed. The full set of guidelines (in Dari and English translation) are distributed along with the dataset, however we provide a brief summary of the guidelines below.

Annotation guidelines The guidelines provided to the annotators contain detailed descriptions of the six emotions with example words typically associated with the different emotions. For instance, the Anger class was described as comprising tweets reflecting emotions of anger, criticism, or frustration where the text may be confrontational, express strong negative feelings, or carry a tone of harsh criticism. Words symbolizing anger might include terms such as 'lying', 'spy', 'traitor', 'hypocrite', 'oppression' etc.

In addition to instructions describing each emotion class, care was taken to delimit the class of "Other" which represents tweets that do not display any particular emotion and convey a neutral tone. For instance, tweets about mundane activities or more fact-based posts would fall under this category. Annotators were further instructed to do their best to not let personal agreement or disagreement with the opinions stated in the tweets influence the labeling process and to label without any bias or directionality. Rather, they were instructed to depart from their interpretation of the speaker's emotional state and attempt to describe it as accurately as possible using one of the provided emotion labels.

Table 1 shows some examples of tweets (with English translations) from the LetHerLearn dataset to further illustrate the annotation effort.

Inter-annotator agreement We further assess the consistency of annotations and measure the agreement among two annotators using Cohen's Kappa (Cohen, 1960) for the double labeling of 100 tweets. The agreement attained over the 100 tweets was 0.80.

3.4 Dataset statistics

The total number of words in our dataset after removal of the tweet's Hashtag, URL and Mention is 88,875 words, where 16,276 words are unique and the average length of the tweets is 4.82 words long. Figure 1 shows the occurrences of tweets for each emotion class. Examining the content of the LetHerLearn dataset, we can see that Anger is the most observed emotion, followed by Happiness, and we find that Surprise is the least observed emotion, with only 503 occurrences. The dataset was further split into train-dev-test splits using a 80:10:10 split for experimentation. Table 2 shows the detailed class-wise distribution of train, validation, and test set.

Next, and in order to get some more insight into the contents of our dataset, we examine the distribution of most frequent words per class following stop word removal, as shown in table 3 which displays the top frequent words for each of the emotion classes. We observe that some words frequently occur in all classes such as 'Taliban', 'Afghanistan', 'girls', 'women', 'everyone'. There are also clear lexical indicators associated with each class, such as 'filthy' for Disgust, 'fear'

Type	Train	Dev	Test
Anger	1366	174	187
Disgust	462	50	57
Fear	483	64	59
Happiness	1266	179	152
Sadness	1032	120	128
Surprise	394	46	50
Other	1082	128	128
Total	6085	761	761

Table 2 : Data distribution for experiments

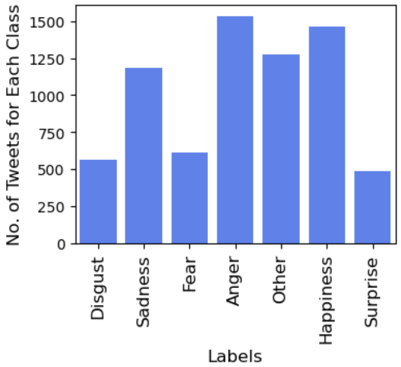


Figure 1 : Number of tweets for each emotion class in LetHerLearn.

Class	Words
Anger	‘work´, ‘should´, ‘islam´, ‘society´
Disgust	‘curse’, ‘tribe’, ‘damnation’, ‘filthy’
Fear	‘fear’, ‘explosion’, ‘escape’, ‘arrest’
Happiness	‘justice’, ‘hope’, ‘fight’, ‘rights’
Sadness	‘pain’, ‘close’, ‘forgot’, ‘tired’
Surprise	‘again’, ‘wish’, “someday’, ‘men’
Other	‘life’,‘world’ ‘further’, ‘iran’

Table 3 : Frequent words in each class following removal of stopwords.

Model	Precision	Recall	F1
LSTM	0.67	0.63	0.65
BiLSTM	0.66	0.63	0.64
GRU	0.65	0.62	0.60
CNN	0.66	0.60	0.62
Ensemble	0.69	0.64	0.66
ParsBERT	0.65	0.65	0.65
XML-RoBERTa	**0.70**	**0.70**	**0.70**

Table 4 : Macro Average Precision, Recall and F1 result of all models on the LetHerLearn test set.

for Fear and ‘pain’ for Sadness. We also observe lexical items describing the cause of emotion, e.g. ‘explosion’ and ‘arrest’ for Fear and ‘justice’ for Happiness.

4 Modeling

We evaluate a number of classic neural models on our dataset:

- Long Short-Term Memory Network (LSTM)
- Bi-directional Long Short-Term Memory Network(Bi-LSTM)
- Gated Recurrent Unit (GRU)
- Convolutional Neural Network (CNN)

All models made use of fastText (Grave et al., 2018) word embeddings with 300 dimensions for Persian. Further hyperparameters of the models are specified in Appendix A.

Ensemble Model After generating predicted probabilities from the LSTM, BiLSTM and GRU models, we develop an ensemble model (Dashtipour et al., 2021) using the scikit-learn library’s VotingClassifier (Leon et al., 2017) class to combine the predictions result of the LSTM, BiLSTM, and GRU models.

ParsBERT We use a pre-trained language model for Persian, ParsBERT (Farahani et al., 2021) which is a monolingual BERT model. Hyperparameters are found in Appendix A.

XLM-RoBERTa-large XLM-RoBERTa is a multilingual transformer-based language model pre-trained data from over 100 different languages (Conneau et al., 2019). Hyperparameters are specified in the appendix.

4.1 Results

The results of our experiments are summarized in Table 4, which shows the evaluation result of the different models described above. The results show that the ensemble model achieves better results compared to the LSTM, BilSTM, GRU and CNN models on their own, as has been shown also in previous work (Onishi and Natsume, 2014).

We further find that the XLM-RoBERTa-large model outperforms the other models. The per-

Tweet	True Label	Predicted Label
سنگدل سیری گرسنه های را نصیحت میکند درد گرسنگی تحمل کند A hard heart satiety advise the hungry to endure the pain of hunger	Anger	Sadness
یکی یکی ارزوی ما نیست میشوند I dreamt that my homeland had become prosperous and independent	Sadness	Anger
طالبان زیر فشار خارجی ها شریعت را فراموش میکنند The Taliban forget the Shariah under foreign pressure	Fear	Anger
وای چی دردهای جانسوزی Oh,What tragic and painful situation	Surprise	Happiness

Table 5 : Examples of misclassified tweet.

Class	Precision	Recall	F1_Score
Anger	0.52	0.57	0.54
Disgust	**0.86**	0.84	**0.85**
Fear	0.84	**0.86**	**0.85**
Happiness	0.67	0.71	0.69
Sadness	0.58	0.61	0.59
Surprise	0.82	0.85	0.84
Other	0.62	0.44	0.52
Macro Average	0.70	0.70	0.70

Table 6 : Individual class performance using XLM-RoBERTa-large model.

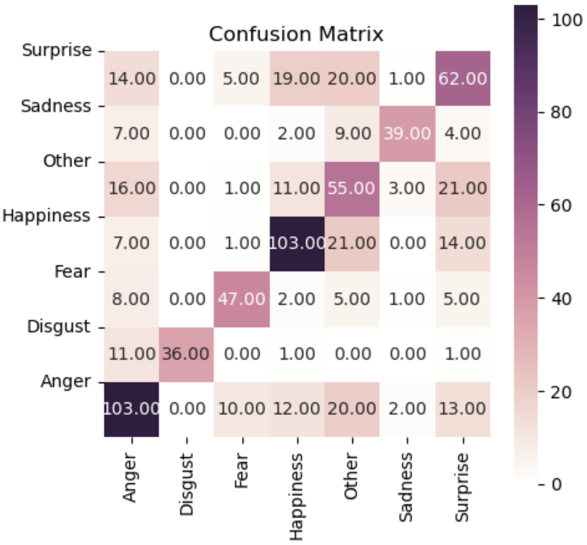


Figure 2 : Confusion matrix heat map.

class results, shown in Table 6 show that the scores vary for the different emotion classes, with the highest results obtained for the Disgust and Fear classes, and the most difficult classes being the Other class, as well as the Anger class.

4.2 Error analysis

We perform an error analysis on the outputs of our model in order to gain further insight into the classifications on the LetHerLearn dataset. It is clear that there is not a direct correlation between low-frequency classes (such as Disgust) and prediction performance. Figure 2 provides a confusion matrix heat map of the predictions. We find that Surprise is often mistaken for other categories, such as Happiness, Other and Anger. Not surprisingly perhaps, the Other class is also often mistaken for other classes.

Following our analysis of the misclassified predictions, we can infer some of the reasons: the assignment of a maximum of one emotion for each tweet is problematic for some of the tweets that have more than emotion. We also analyze the word overlap between the tweets and find that classes with a high degree of overlap tend to also suffer from misclassification. Table 5 shows some examples of misclassified predictions.

5 Conclusion

We have presented LetHerLearn: the first Dari emotion-annotated dataset of tweets collected following the Taliban's ban of women's education in 2022. All data and code will be made available.[1] In future work, we would like to experiment with cross-variant Persian emotion detection as well as multitask learning of sentiment and emotion.

---

[1] `https://github.com/ltgoslo/LetHerLearn`; due to the privacy restrictions applied by Twitter API, only tweet IDs along with annotations will be made available.

References

Taha Shangipour Ataei, Kamyar Darvishi, Soroush Javdan, Behrouz Minaei-Bidgoli, and Sauleh Eetemadi. 2019. Pars-absa: an aspect-based sentiment analysis dataset for persian. *arXiv preprint arXiv:1908.01815*.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Kia Dashtipour, Cosimo Ieracitano, Francesco Carlo Morabito, Ali Raza, and Amir Hussain. 2021. An ensemble based classification approach for persian sentiment analysis. *Progresses in Artificial Intelligence and Neural Systems*, pages 207–215.

Raymond J Dolan. 2002. Emotion, cognition, and behavior. *science*, 298(5596):1191–1194.

Paul Ekman. 1992. Facial expressions of emotion: an old controversy and new findings. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 335(1273):63–69.

Mehrdad Farahani, Mohammad Gharachorloo, Marzieh Farahani, and Mohammad Manthouri. 2021. Parsbert: Transformer-based model for persian language understanding. *Neural Processing Letters*, 53:3831–3847.

Soumitra Ghosh, Asif Ekbal, Pushpak Bhattacharyya, Sriparna Saha, Vipin Tyagi, Alka Kumar, Shikha Srivastava, and Nitish Kumar. 2020. Annotated corpus of tweets in english from various domains for emotion detection. In *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*, pages 460–469.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. *arXiv preprint arXiv:1802.06893*.

Mohammad Heydari. 2019. Sentiment analysis challenges in persian language. *arXiv preprint arXiv:1907.04407*.

Pedram Hosseini, Ali Ahmadian Ramaki, Hassan Maleki, Mansoureh Anvari, and Seyed Abolghasem Mirroshandel. 2018. Sentipers: a sentiment analysis corpus for persian. *arXiv preprint arXiv:1801.07737*.

Florin Leon, Sabina-Adriana Floria, and Costin Bădică. 2017. Evaluating the effect of voting methods on ensemble-based classification. In *2017 IEEE international conference on INnovations in intelligent Systems and applications (INISTA)*, pages 1–6. IEEE.

Hossein Mirzaee, Javad Peymanfard, Hamid Habibzadeh Moshtaghin, and Hossein Zeinali. 2022. Armanemo: A persian dataset for text-based emotion detection. *arXiv preprint arXiv:2207.11808*.

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. SemEval-2018 task 1: Affect in tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 1–17, New Orleans, Louisiana. Association for Computational Linguistics.

Pansy Nandwani and Rupali Verma. 2021. A review on sentiment analysis and emotion detection from text. *Social Network Analysis and Mining*, 11(1):81.

Akinari Onishi and Kiyohisa Natsume. 2014. Overlapped partitioning for ensemble classifiers of p300-based brain-computer interfaces. *PloS one*, 9(4):e93045.

Nazanin Sabri, Reyhane Akhavan, and Behnam Bahrak. 2021. Emopars: A collection of 30k emotion-annotated persian social media texts. In *Proceedings of the Student Research Workshop Associated with RANLP 2021*, pages 167–173.

Brian Spooner. 2012. 4. persian, farsi, dari, tajiki: Language names and language policies. In *Language Policy and Language Conflict in Afghanistan and Its Neighbors*, pages 89–117. Brill.

Shima Zobeidi, Marjan Naderan, and Seyyed Enayatallah Alavi. 2019. Opinion mining in persian language using a hybrid feature extraction approach based on convolutional neural network. *Multimedia Tools and Applications*, 78:32357–32378.

A Appendix

A.1 Model hyperparameters

Long Short-Term Memory Network(LSTM) Our model has 128 neurons with dropout and recurrent-dropout of 70%. The optimizer is adam and the number of epochs is 30 with learning rates of 0.01. and an output layer with 7 neurons, one for each class, batch size is 80.

Bidirectional Long Short Term Memory Network (Bi-LSTM) We use Bidirectional LSTMs with SpatialDropout1D of 0.2 and dropout and recurrent-dropout of 70%. We trained with different optimizers and achieved the highest result with adam, 30 epochs and a learning rates of 0.01 with batch size 64.

Gated Recurrent Unit (GRU) The GRU network has 4 layer in our proposed model and after feature extraction using fastText word embedding, the embedding layer of size (15130, 300) constitutes the input. We used adam optimizer with β1 = 0.9, β2 = 0.999 and learning rate of 0.01. Our model includes 64 neurons, SpatialDropout1D of 0.2 and dropout and recurrent-dropout of 65%.

Convolutional Neural Network (CNN) Our proposed CNN model has four layers, the Conv1D layer has 256 filters and a kernel size of 5 with relu activation function. The Dense layer has seven unit with softmax activation function, batch size is 96, Dropout value is 0.7 and we used the adam optimizer with β1 = 0.9, β2 = 0.999 and learning rate of 0.01.

ParsBERT The hyperparameters for the ParsBERT model fine-tuning was performed for seven epochs with a batch size of 32, specifying the maximum length of the encoded sequence to 128 and using the AdamW optimizer with β1 = 0.9, β2 = 0.999, learning rate scheduler is (2e-5) and linearly decreasing from the initial to 0 by the end of the last epoch.

XLM-RoBERTa-large The hyperparameters for the the XLM-RoBERTa-large is eight epochs, batch size of 32, learning_rate of 2e-5, optimizer of Adam and maximum length of 128. Table 6 shows the prediction results for each emotion class using this model.

Identifying Slurs and Lexical Hate Speech via Light-Weight Dimension Projection in Embedding Space

Sanne Hoeken, Sina Zarrieß and **Özge Alaçam**
Faculty for Linguistics and Literary Studies
University of Bielefeld, Bielefeld, Germany
`{sanne.hoeken, sina.zarriess, oezge.alacam}@uni-bielefeld.de`

Abstract

The prevalence of hate speech on online platforms has become a pressing concern for society, leading to increased attention towards detecting hate speech. Prior work in this area has primarily focused on identifying hate speech at the utterance level that reflects the complex nature of hate speech. In this paper, we propose a targeted and efficient approach to identifying hate speech by detecting slurs at the lexical level using contextualized word embeddings. We hypothesize that slurs have a systematically different representation than their neutral counterparts, making them identifiable through existing methods for discovering semantic dimensions in word embeddings. The results demonstrate the effectiveness of our approach in predicting slurs, confirming linguistic theory that the meaning of slurs is stable across contexts. Our robust hate dimension approach for slur identification offers a promising solution to tackle a smaller yet crucial piece of the complex puzzle of hate speech detection.[1]

1 Introduction

Recent years have seen an increase in attention towards hate speech detection due to the rising prevalence of online hate speech and its negative impact on society (Zhang and Luo, 2019). Current approaches to hate speech detection focus on identifying hate speech at the utterance level (Zampieri et al., 2020), which remains a challenging task due to the nuanced and complex nature of hate speech. Hate speech can take many different forms, especially in the context of social media platforms where language use is dynamic and constantly evolving (Davidson et al., 2017). This paper aims to tackle one aspect of hate speech detection by identifying hate speech at the lexical level, specifically through the identification of slurs based on their contextualized representations. Breaking down the problem into smaller pieces allows us to focus on specific aspects of hate speech, such as slurs, and understand how "hatefulness" is encoded as a dimension of meaning in the embedding space of language models. This, in turn, can inform the development of more robust hate speech detection methods.

Slurs can be defined as terms referencing a specific social group, and generally carry derogatory connotations, regardless of the situation in which they are used. Thus, the meanings of slurs remain relatively consistent across different contexts (Hess, 2021). In contrast, more subtle forms of hate speech such as dog whistles or expressions that depend on the speaker or audience can vary widely in their interpretation. For instance, the word "bitch" may be used as a derogatory term towards women, but among women, it can also be used casually in certain contexts (Davidson et al., 2019). Slurs are commonly included in hate speech lexicons, yet systematic study into automatically identifying them has barely been touched upon. Nevertheless, their semantic consistency across contexts makes slurs a promising target for identification based on their embedding representation.

Word embeddings have demonstrated their effectiveness in capturing various aspects of meaning, including relations like synonymy and antonymy, but also more abstract concepts like cultural or social bias. As a prominent example, previous studies have shown that gender bias can be detected by extracting the relevant semantic dimension from the embedding model (Bolukbasi et al., 2016; Garg et al., 2018). These "semantic dimension identification" techniques typically rely on a small set of carefully selected words or word pairs that only differ with respect to the semantic dimension of interest. Although this approach has demonstrated its capability to generalize to different dimensions of meaning, there are still open questions and chal-

---

[1] The code and accessible data used for the experiments can be found at `https://github.com/SanneHoeken/SlurDimension`

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 278–289
July 14, 2023 ©2023 Association for Computational Linguistics

lenges. For example, the task of selecting the right set of words or word pairs to capture a specific dimension of meaning is still indistinct.

Building on linguistic theories on slurs and the findings of dimension identification studies, we hypothesize that slurs have a systematically different representation than their neutral counterparts in the embedding space, and that this difference can be identified using existing methods for discovering semantic dimensions within word embedding models (Kozlowski et al., 2019). Specifically, our study addresses the following research questions: 1) can we identify slurs based on their contextualized word embeddings? 2) how do we leverage dimension-based methods for slur identification? 3) can we confirm existing work in linguistics which suggest that the meaning of slurs is stable across contexts? 4) can we use the hate dimension identified based on slurs for detecting other lexical units pertaining to hate speech?

In addressing these questions, we focus on methodological aspects such as the selection of lexical pairs and leveraging a pre-trained contextualized language model and incorporating multi-word expressions. In addition, our research puts emphasis on the robustness of the proposed methods across various hate speech domains and datasets without the need of big annotated data.

To sum up, this paper presents a more targeted and efficient method for detecting hate speech, that aims to identify and gain more insight into the use of slurs in online discourse. [2]

2 Related work

In this section, we review existing research on two key areas related to our study. First, we address hate speech at the lexical semantic level, with a particular focus on slurs. Second, we discuss previous work on semantic dimension identification and its applications in computational semantics.

2.1 Lexical semantics of hate speech

Hate speech can manifest itself in various forms at the lexical-semantic level, including both explicit and subtle expressions of derogatory language. Pejorative terms such as "nigger" or "faggot" fall into the former category, while more covert forms of hate speech include the use of code words and dog whistles like "inner-city" (referring to poor African-American) (Anderson and Barnes, 2022).

One prototypical (and explicit) form of hate speech is the use of slurs. Slurs are pejorative lexical items that refer to social groups defined by a factor such as race, ethnicity or religion, and convey derogatory attitudes toward those groups and their members (Hess, 2021). In his theoretical overview, Hess (2021) identifies several semantic and pragmatic properties of slurs. These properties include the observation that negative connotations of slurs persist even when used under negation, modals, or in conditionals, and that the derogatory meaning of a slur is independent of the speaker's intentions or attitudes. This means that every use of a slur is considered offensive. Additionally, most scholars agree that for every slur, there exists a neutral counterpart that can denote the same social group without causing offense (Falbo, 2021; Bach, 2018). For example, the term "beaners" in American English is generally understood as a derogatory term used to refer to "Hispanic people" regardless of context.

The lexical aspect of hate speech has been a key focus in hate speech detection models. Earlier feature-based classification systems relied on identifying specific words and phrases that are commonly associated with hate speech, such as slurs, by employing discrete hate speech lexica (Schmidt and Wiegand, 2017). However, the explicit modeling of slurs or slur detection has not been extensively explored in this field. Currently, the only notable work in this direction is presented by Wiegand et al. (2018) who proposed a method to automatically expand a base lexicon of abusive words through a feature-based classification system. Nevertheless, their engineered features are resource-intensive as they depend on multiple corpora and lexical resources. Their system also incorporates a lexical graph propagation framework, which has been previously applied in domains beyond abusive language detection. Hamilton et al. (2016) demonstrated its applicability in generating sentiment lexicons. However, creating a lexical graph requires a semantic space that is learned from a substantial corpus of data. Furthermore, Hamilton et al. (2016) demonstrated that their method is only effective for domain-specific applications.

The linguistic properties of slurs, i.e. having neutral counterparts and invariant offensiveness, make

---

[2] Please note that this paper includes the use of offensive language, solely for the purpose of illustrating theoretical concepts and our proposed methodology. We acknowledge that such language may be harmful and recognize that its use does not reflect our personal beliefs or values.

them potentially suitable for a domain-independent semantic dimension approach that does not necessitate extensive data. In the following section, we will discuss the computational linguistic aspects of a semantic dimension approach in more detail.

2.2 Semantic dimensions within word embeddings

Word embedding models have demonstrated their capacity to represent shared relationships fundamental to word analogies, as constant vector offsets between pairs of words (Mikolov et al., 2013). An increasingly important line of research focuses on detecting biases with and within word embeddings. Bolukbasi et al. (2016) proposed a method based on the concept of gender direction, which involves identifying the dimension in the embedding space that captures gender information. More precisely, they take the difference vectors of 10 curated word pairs and calculate their Principal Components (PC). Subsequently, the top PC is identified as the dimension vector. Garg et al. (2018) extended this work by proposing a more general method that can identify multiple types of biases, including those related to race and religion.

Kozlowski et al. (2019) also employ the semantic dimension approach, but with a focus on analyzing cultural meaning rather than revealing bias in word embeddings. They showed that identified dimensions capturing cultural information such as affluence and status, estimated as the mean difference vector of a set of word pairs, are consistent with human-rated associations measured by contemporary and historical surveys. We adopt the dimension identification technique by Kozlowski et al. (2019) for the purpose of detecting slurs.

The analysis of semantic dimensions by leveraging the geometrical properties of the vector space has traditionally been performed using static word embedding models. Bommasani et al. (2020) introduced a novel approach to identifying social biases in pre-trained contextualized language models, such as BERT (Devlin et al., 2019) and GPT-2 (Radford et al., 2019). The proposed technique involves reducing contextualized representations to static embeddings, which allows for the application of previously developed methods for social bias identification, as considered above.

In summary, our review of previous research has demonstrated the potential of using semantic dimension identification techniques to detect and analyze slurs, including contextualized language models. In the following section, we will describe our methodology for applying these techniques to the identification of slurs.

3 Methodology

Our approach can be divided into 1) the selection of pairs and extraction of their lexical representations from a language model, 2) the creation of a dimension vector based on those representations, and 3) the projection of lexical representations onto this dimension. We outline each of these components in Sections 3.1 through 3.3, respectively.

3.1 Extract representations

Instead of focusing on single words, we include both single words and multi-word expressions in our approach. The main motivation for this is the observation that not only slurs, but especially their neutral counterparts, are frequently encoded through multiple words. For instance, both the slur "porch monkey" and it's neutral counterpart "Black person" consist of two words. In the remainder of this paper, we therefore refer to the representations of *lexical units*, rather than words. We elaborate on our selection of lexical units in Section 4.1.

To obtain representations of lexical units from a pre-trained contextualized language model, we mainly follow techniques presented by Bommasani et al. (2020). The specifics of the particular model employed in our experiments is outlined in Section 4.3. For each lexical unit w, we sample n contexts from collected social media data that we detail in Section 4.1. We feed each context c to the model and extract the i hidden layers. Then we perform the following steps in order to compute a vector w_c for each lexical unit in each context:

1. **Layer selection & aggregation**. We average over all i hidden layers L encoding c: $L_c = mean\left(L_c^1, \ldots, L_c^i\right)$. The result is a matrix L_c of size x by d, with x being the number of subword-tokenized tokens and d the dimensionality of each layer in the model.
2. **Subword selection & aggregration**. Given all subtokens t in c, $L_c = [t_c^1, \ldots, t_c^x]$, we average over the k subword tokens generated for w_c starting at position s in c: $w_c = mean\left(t_c^s, \ldots, t_c^{s+k}\right)$. The resulting lexical representation is thus a d-dimensional vector which is the mean across all i hidden layers

and all k subtoken encodings that constitute the lexical unit w in c.

3.2 Dimension creation

Following Kozlowski et al. (2019), we calculate a semantic dimension by taking the mean of a set of pairs of lexical representations that we obtained in the previous step, whose semantic difference corresponds to the dimension of interest.

For dimension computation, we aggregate them into a single representation for each lexical unit w by taking the average representation across n contexts: $w = mean\left(w_{c_1}, \ldots, w_{c_n}\right)$. To obtain the final dimension vector v, we calculate the average difference between p pairs of static representations of a slur w^s and its neutral counterpart w^n: $v = mean\left(\left(w_1^s - w_1^n\right), \ldots, \left(w_p^s - w_p^n\right)\right)$.

3.3 Projection onto dimension

The degree of hate encoded in the embedding of a lexical unit can be determined by its projection onto the dimension. Given an embedding w, this projection onto the dimension is defined as the cosine distance between the lexical unit and dimension vector v (Kozlowski et al., 2019). Hateful lexical units should exhibit positive projection values, and neutral terms negative values.

4 Data & Experiments

The ingredients for our semantic dimension approach using a contextualized language model are lexical units and contexts in which they occur. The subsequent sections will introduce the data that we utilize to construct dimensions (4.1) and to assess them (4.2), respectively.

4.1 Data for dimension creation

Lexical Units. As discussed earlier, slurs and their neutral counterparts seem to form ideal *pairs* of lexical units that differ only with respect to the semantic dimension of hate. To create a set of such pairs, we utilized HateBase[3], a commonly used lexicon for hate speech detection. The English Hatebase lexicon contains 1565 hate terms, including but not limited to slurs. We filtered for slurs by identifying lexical units that refer to (members of) social groups, that are hateful in any use, and for which a neutral counterpart could be found.

To find neutral counterparts, we consulted the definition of the lexical unit and the annotation of the target group as provided in HateBase, as well as definitions from resources like Wictionary[4] and other online dictionaries if needed. Furthermore, our primary objective was to comply with the APA Style guidelines for bias-free language[5].

Next, out of the resulting 617 pairs, we filtered for pairs that appeared at least 10 times in the datasets used for context sampling (which we discuss next). From this filtered list of almost 70 pairs, we selected 15 pairs in such a way that each social group was represented by no more than one word pair. Additionally, we ensured that the selected pairs provided the best possible spread across target group categories such as ethnicity and religion. Table 1 presents the final set of pairs.

	Slur	Neutral counterpart	Category
1	beaners	Hispanic people	Ethnicity
2	gooks	Asian people	Ethnicity
3	injuns	Native Americans	Ethnicity
4	Argies	Argentinians	Nationality
5	limeys	British people	Nationality
6	pakis	Pakistanis	Nationality
7	feminazis	feminists	Gender
8	tranny	transgender people	Gender
9	whore	prostitute	Gender
10	kikes	Jews	Religion
11	muzzies	Muslims	Religion
12	darkies	Black people	Race
13	whitey	White person	Race
14	hillbillies	rural people	Class
15	libtard	Liberal person	Politics

Table 1: 15 pairs of slurs and their neutral counterparts, used for dimension creation, and the category of the social group they refer to.

Contexts. In order to obtain lexical representations from a contextualized language model, we provide the model with lexical units *within contexts*. Feeding the model with isolated units (i.e. without any context) would be an unnatural input to the model. To this end, we collect a set of user-generated web-data from Reddit, a social media platform that allows users to create communities (called subreddits) based on a wide range of topics and interests. Users can submit content, such as links, text posts, images, and videos. Reddit is often used as a resource in hate speech research (e.g. Saha et al. (2019); Rieger et al. (2021)) because it has a large user base and allows for anonymity,

---

[3] `https://hatebase.org`

[4] `https://www.wiktionary.org`

[5] `https://apastyle.apa.org/style-grammar-guidelines/bias-free-language`

which can encourage people to express controversial or offensive opinions.

To obtain a diverse range of data, we utilized the Pushshift API (Baumgartner et al., 2020) to scrape a random sample of 5.8 million comments from Reddit which spans from its inception in December 2005 up to March 2023. Additionally, we collected a second dataset from Reddit that is, in contrast to the first, highly domain-specific: around 10 million comments posted in the year 2016 on The_Donald subreddit. This subreddit was created to support the United States presidential campaign of Donald Trump and was eventually banned by Reddit in 2020 for violating its policies on hate speech and inciting violence (Yurieff, 2020). By comparing the results of the two different sets of data sources (for dimension creation), we aim to gain insights into what extent the source domain affects the quality of the resulting dimension.

4.2 Evaluation data

In order to assess the representational quality of a created dimension vector we project lexical representations of a set of test terms onto the dimension, and compare the projection values with human evaluations of the hatefulness of these terms.

Lexical Units First, as a preliminary check, we established a test set that also utilizes the HateBase lexicon as data source. We selected a set of lexical units consisting of slurs and neutral terms from the nearly 70 pairs that we formed before (see Section 4.1). We selected 40 slurs and 30 neutral terms, independently, that were not part of the 15 pairs selected for dimension creation (Table 1).

Second, we test our method on a more complex task, which entails assessing terms in a context-dependent manner and allows us to draw conclusions that is not limited to the HateBase source data. To this end, we leverage the HateXplain dataset which consists of over 20,000 posts from Twitter and Gab, annotated for hate speech (Mathew et al., 2021). Notably, annotators have marked parts of the post text that could be a potential reason for its perceived hatefulness. This information is provided as the "explanation rationale" for each post, which is a list that identifies marked tokens with a 1 (denoting hateful contribution) and unmarked with a 0. We identified all the unique tokens in the HateXplain dataset, and filtered out any non-stopword nouns[6] with a frequency of more than 10. For each of the resulting 2764 terms, we collected the rationale scores assigned by multiple annotators (often 3) to each instance of the term. We aggregated the scores by taking the majority score (0 or 1) for each token. For tokens in neutral posts, for which no explanation rationales were provided, we set the scores to 0.

For our first HateXplain-sourced test set, we selected 100 nouns that refer to persons and ensured that the selection included a proportional mix of both neutral and derogatory terms. To evaluate the extent to which our slur dimension is exclusively limited to persons, we gathered a second test set that encompasses all types of nouns. Specifically, we sampled 100 nouns from the HateXplain vocabulary with an approximately uniform distribution across the corresponding average rationale scores. All three lists of test terms described here are included in Appendix A.

Contexts Each final test input includes an online post containing a particular lexical unit to provide contextualized lexical representations for all test terms. For HateXplain-sourced items, the posts available in the HateXplain dataset serve as the context. For HateBase-based lexical units that are only available without any context, we use the (general) Reddit dataset we collected in previous steps (see Section 4.1) to obtain contextualized forms of the test items. In both settings, we include the contextualized representation of *each* occurrence of a lexical unit in the test data for projection.

4.3 Experimental set-up

In our default experimental set-up, a dimension vector is computed as the mean distance vector of 15 pairs of slurs and their neutral counterpart given in Table 1. In doing so, an average lexical representation for each pair part is generated across 10 contexts taken from our collected dataset, consisting of randomly sampled Reddit comments. For the generation of lexical representations, we use the pre-trained model DistilBERT (Sanh et al., 2019), which is a distilled version of BERT and consists of 6 layers of transformer blocks, each of which has 768 hidden units.[7] Each individual contextualized representation is extracted as the average of all DistilBert's hidden layers, limited to and averaged over the sentence positions of the subwords that

---

[6] We employed the Natural Language ToolKit (NLTK) for selecting nouns and excluding stopwords.

[7] We implemented 'distilbert-base-uncased' through Hugging Face's *transformers* library for Python (Wolf et al., 2020).

constitute the lexical unit.

Contextualized representations of test items are projected onto a computed dimension and compared to human assessments using our three different test sets, two utilizing the HateXplain dataset and one using the HateBase lexicon (see Section 4.2). In our evaluation, we employed a combination of token-level and type-level comparison using correlation and classification metrics. To classify lexical units as hateful or neutral, we used their projection values, with positive values indicating hateful and negative values indicating neutral. For classification accuracy, we used the chi-squared test to calculate its statistical significance.

For type-level evaluation, we calculated the average projection value across all contextualized instances of each test term, whereas for token-level evaluation, we considered each instance. This allowed us to assess the performance of our method in predicting terms within their context (made possible with the HateXplain dataset) as well as utilizing HateBase, a context-independent source of hatefulness ratings. In assessing projections of HateXplain test terms, we also measured correlation using the average rationale score (0 or 1) for each term across all instances. We compared these type-level scores with the type-level projection values, using Pearson's Correlation.

5 Results

In the following, we present the results of our default set-up on two test sets, as well as experiments examining the impact of the selection of pairs (5.1). Furthermore, we analyze the effect of the number of contexts included and the domain they are sourced from (5.2). Finally, we discuss our findings on our third test set that includes other categories besides persons (5.3).

5.1 Main results

Table 2 reports our main results. Overall, our dimension approach demonstrates effectiveness in predicting slurs as evidenced by the performance results. Specifically, our method achieves accuracy rates of around 0.90 on our HateBase test set and 0,77 on the HateXplain test set limited to person terms. The higher accuracy displayed by the HateBase test set may be explained by its utilization of the same data resources as the dimension data.

Pair selection To investigate the impact of the selection of pairs on performance, we utilized two set-ups: In the first, instead of a (more specific) co-hyponym, which applies to true counterparts, we replaced all neutral counterparts with the more general hypernym "person" or "people". The semantic difference between the two pair-parts here thus involves more than purely deragotary connotation, which seems to be reflected in the resulting dimension. The second and fourth boxplot in Figure 1 show that the projection values of hateful terms in the HateXplain test set are lower overall, indicating a weaker association with the dimension. As shown in Table 2, this change caused a drop in the recall of hateful tokens in the HateXplain test set. This effect was, however, not observed for the HateBase test set. Despite observing lower projection values for hateful terms, misclassifications did not increase. One possible explanation is that the test hate terms are more similar to the slurs used for dimension creation, thereby maintaining a positive association with the dimension vector.

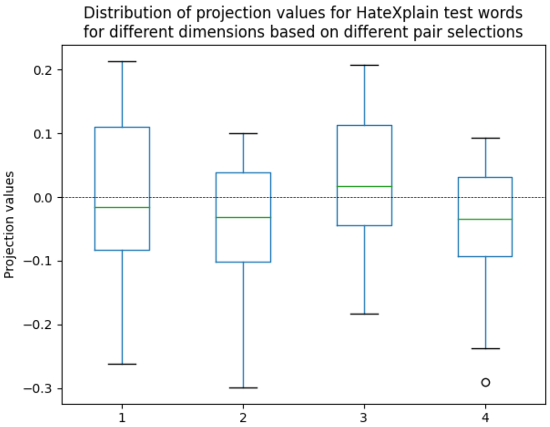


Figure 1: Effect of pair selection for dimension creation on projection values. Pair selections: *1 = all slurs & co-hyponym counterparts, 2 = all slurs & hypernym counterparts, 3 = nationality/ethnicity slurs & co-hyponym counterparts, 4 = nationality/ethnicity slurs & hypernym counterparts*

In the second set-up, we limited the set to only lexical units referring to social groups categorized by ethnicity and nationality, reducing the number of pairs to six (i.e. the first six pairs in Table 1). The resulting dimension represents a narrower spectrum of hate, which caused a significant decrease in the precision of predicting hateful terms correctly in the HateXplain data, but not in the HateBase test set. This discrepancy may be due to the majority of HateBase test terms referencing nationality or ethnicity categorized groups. Additionally, we found that combining hypernym counterparts with a restrictive set of slurs did not result in an increase

Pairs		Correlation	Classification report						
Slurs	Counter parts	Pearson's r (* = sig.)	Acc. (* = sig.)	Hateful			Neutral		
				Prec.	Recall	F1	Prec.	Recall	F1
HateBase test set									
All	Co-hyponyms	-	0.90*	1.00	0.82	0.90	0.81	1.00	0.90
All	Hypernyms	-	0.89*	0.97	0.82	0.89	0.81	0.97	0.88
Nat./Eth.	Co-hyponyms	-	0.91*	1.00	0.85	0.92	0.83	1.00	0.91
Nat./Eth.	Hypernyms	-	0.89*	0.97	0.82	0.89	0.81	0.97	0.88
HateXplain test set - persons									
All	Co-hyponyms	0.790*	0.77*	0.76	0.86	0.81	0.79	0.66	0.72
All	Hypernyms	0.755*	0.77*	0.80	0.80	0.80	0.74	0.74	0.74
Nat./Eth.	Co-hyponyms	0.770*	0.76*	0.71	0.95	0.82	0.90	0.51	0.65
Nat./Eth.	Hypernyms	0.737*	0.77*	0.80	0.79	0.80	0.74	0.75	0.74

Table 2: Performance results for dimensions with different pair selections and different test sets. (n = 15 for *All* slurs, n = 6 and for *Nat./Eth.* (Nationality/Ethnicity) slurs)

in false positives in either test set. This could be because the effect of hypernym counterparts in decreasing false positives is stronger.

Overall, these findings underscore that manipulating the hate specificity of the dimension by selecting different pairs does not significantly impact the overall accuracy and F1-scores. However, it does have a notable effect on the occurrence of false negatives or false positives, which is particularly relevant for hate speech detection.

5.2 Number & domain of contexts

Number of contexts Each lexical representation is produced based on the 10 contextual representations (as mentioned in Section 4.3). To evaluate the necessity of such data quantity, we conducted projection tests with dimensions based on less than 10 contextualized representations per lexical unit.

Figure 2 depicts the impact of the number of contexts on dimension performance, with detailed results presented in Table 6 in Appendix B. The results indicate that larger sample sizes result in greater stability in performance, as evidenced by reduced variation introduced by random sampling. Yet more importantly, our analysis suggests that the effectiveness of a dimension is not significantly influenced by the size of the context sample. This implies that accurate results could be obtained even with smaller amounts of data, thereby providing a more efficient and cost-effective method.

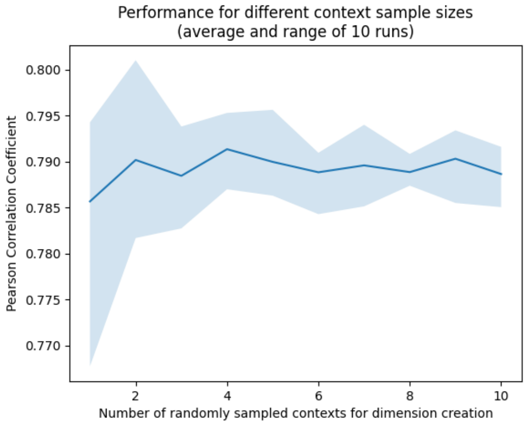


Figure 2: Effect of number of contexts on performance on HateXplain test words

Domain of contexts In addition to the quantity of contexts, we also tested the influence of the domain from which the contexts were sourced. Rather than sampling contexts from comments across the entire Reddit spectrum, we conducted an additional experiment using domain-specific comments from The_Donald subreddit, specifically those from 2016. This change did not reveal any significant effect on the dimension performance, as evidenced by the performance results in Table 3. These findings align with our previous results regarding sample size, and furthermore, they indicate that domain-specific data is also appropriate for our method, thereby increasing its versatility. Most importantly, the results confirm linguistic theory that the meaning of slurs is stable across contexts (Hess, 2021).

5.3 Generalizing to other categories

In our final experiment, we tested our dimension approach on another dataset comprising 100 random nouns from the HateXplain vocabulary in their respective contexts. We observe a significant drop in

Correlation	Classification report						
Pearson's r	Acc.	Hateful			Neutral		
(* = sig.)	(* = sig.)	Prec.	Recall	F1	Prec.	Recall	F1
0.790*	0.77*	0.76	0.87	0.81	0.79	0.65	0.71

Table 3: Performance results for a dimension based on domain-specific contexts on HateXplain test words

performance when predicting nouns that were not restricted to persons, like "stupidity" and "lottery". In particular, the correlation coefficient decreased by 0.3 (See Table 4).

Correlation	Classification report						
Pearson's r	Acc.	Hateful			Neutral		
(* = sig.)	(* = sig.)	Prec.	Recall	F1	Prec.	Recall	F1
0.482*	0.55*	0.43	0.53	0.47	0.66	0.56	0.61

Table 4: Performance results for dimension on HateXplain test set comprising random nouns

An analysis of the projections of the non-persons test terms (on token level) shows that most errors were false positives, with a significant number of neutral terms incorrectly predicted as hateful (See confusion matrix in Table 5.

		Projection	
		Hateful	Neutral
Gold	Hateful	1487	1337
	Neutral	1992	2562

Table 5: Confusion matrix for classification on HateXplain test set comprising random nouns

Many of these false positives were non-nouns, like "pay", "lmao" and "pro". This shows some inadequacies of the noun filtering method for the construction of the test set, as well as the need for greater part-of-speech robustness. Other false positives comprised nouns that did not refer to persons, such as "tweets" and "propaganda". Nonetheless, our methodology also demonstrated its ability to correctly label such terms, as evidenced by the correct prediction of e.g. "movement", "prison", and "knowledge".

Our analysis of false negatives has revealed limitations in using the HateXplain dataset as gold data for our specific purpose. We attribute this issue to the distinction between utterance-level and lexical-level purposes, which we have touched upon in the introduction. The human rationale scores in the HateXplain dataset reflect a word's contribution to the overall hateful meaning of the utterance. We, on the other hand, employed them as evaluations of the hatefulness of a specific lexical unit within a given context. This approach posed problems as demonstrated by the largest group of false negatives, which include terms that reference target groups but do not necessarily contain derogatory connotations at a lexical-semantic level, such as "feminist", "homosexuals" and "refugee".

Lastly, the results also indicated promising classification beyond the intended slur detection: Firstly, the method detected a hateful term that does not refer to persons, i.e. "holohoax". Secondly, the method detected the ambiguous term "fruit", that appeared to be used derogatorily to refer to LGBT people in certain contexts. For example: " *yep and he meets that satanic **fruit** every week how r... and g... is this man*". These findings suggest that our method has a potential wider application in detecting offensive language beyond just slurs.

6 Discussion & Future directions

The results indicate that our dimension approach is effective in predicting slurs based on their contextualized embedding, with the importance of selecting pairs carefully to create a robust hate dimension. Due to the lack of a universally agreed-upon definition of hate speech, the creation of hate speech datasets is difficult and prone to bias (Davidson et al., 2017; Waseem and Hovy, 2016). As a result, datasets are often limited in size and scope, making it challenging to train models that can effectively detect a wide range of hate speech in different domains. Our results demonstrated that the effectiveness of the dimension is not significantly influenced by the size of the context sample. This implies that our dimension approach is a promising cost-effective and domain-agnostic method for identifying slurs with low-data requirements.

6.1 Generalizability

When it comes to classifying *non-person* nouns, we observe a decline in the performance. However, our approach also shows promising results in detecting other categories than slurs, opening a possibility for extension beyond slurs. The further analysis indicated that many errors can be attributed to the quality of evaluation data rather than inherent limitations of the method itself (Section 5.3). Regarding the data employed in our study, we have selected a diverse yet bounded domain coverage, for the purpose of maintaining a systematic approach. Nevertheless, it is worth noting that our findings encourage further exploration of performance in alternative contexts. To illustrate this point, it would be interesting to observe how our method performs when faced with phenomena such as the non-derogatory use of the n-word slur within certain in-group contexts.

6.2 Technical considerations

Prior research on extracting lexical representations from models like BERT demonstrated significant effects of hidden layer selection on the efficacy of the derived representations for various lexical-semantic tasks (Vulić et al., 2020; Bommasani et al., 2020). While averaging all hidden layers generally yields beneficial representations, no single layer configuration stands out as the overall best. The optimal configuration appears to depend heavily on the task and methodology employed. Future research should investigate alternative layer configurations to improve the effectiveness of the representations for identifying slurs.

Moreover, it is important to experiment with different definitions of dimension computation in future research, such as PCA-based (Bolukbasi et al., 2016) or vector offset-based methods (Garg et al., 2018). This is particularly crucial since Bommasani et al. (2020) demonstrated the significant effect of the bias quantification method on the measured bias in lexical representations.

7 Conclusion

This paper addresses the complex puzzle of hate speech detection by breaking it down and concentrate on a smaller but crucial piece, the identification of slurs. We propose a novel approach that applies semantic dimension identification with contextualized embeddings to the detection of slurs. In this study, we set out to address several key research questions concerning the identification of slurs. First, we investigated whether slurs can be identified based on their contextualized word embeddings. Our experimental results demonstrated the effectiveness of our method in predicting slurs by leveraging contextual representations, thereby affirming their effectiveness. Simultaneously, we explored the application of dimension-based methods for slur identification. Our findings highlight the significance of carefully selecting lexical pairs while demonstrating that extensive data is not necessarily required. Additionally, we aimed to confirm existing work in linguistics, which suggests that the meaning of slurs is stable across contexts. Findings on our experiments across different domains and datasets align with linguistic theory, as evidenced by consistently strong prediction performance. Lastly, we explored the potential of utilizing the hate dimension identified based on slurs for detecting other lexical units related to hate speech. Our method exhibited promising results in detecting other categories of lexical hate speech, showcasing its broader applicability potential beyond slurs. In conclusion, our approach contributes to a more targeted and efficient method for detecting hate speech and sheds light on the use of slurs in online discourse.

Acknowledgements

The authors acknowledge financial support by the project “SAIL: SustAInable Life-cycle of Intelligent Socio-Technical Systems" (Grant ID NW21-059A), which is funded by the program “Netzwerke 2021" of the Ministry of Culture and Science of the State of Northrhine Westphalia, Germany.

References

Luvell Anderson and Michael Barnes. 2022. Hate Speech. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*, Spring 2022 edition. Metaphysics Research Lab, Stanford University.

Kent Bach. 2018. Loaded words: On the semantics and pragmatics of slurs. In *Bad Words: Philosophical Perspectives on Slurs*, pages 60–76. Oxford University Press Oxford.

Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. 2020. The pushshift reddit dataset. *Proceedings of the International AAAI Conference on Web and Social Media*, 14(1):830–839.

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *Advances in neural information processing systems*, 29.

Rishi Bommasani, Kelly Davis, and Claire Cardie. 2020. Interpreting Pretrained Contextualized Representations via Reductions to Static Embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4758–4781, Online. Association for Computational Linguistics.

Thomas Davidson, Debasmita Bhattacharya, and Ingmar Weber. 2019. Racial bias in hate speech and abusive language detection datasets. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 25–35, Florence, Italy. Association for Computational Linguistics.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. *Proceedings of the International AAAI Conference on Web and Social Media*, 11(1):512–515.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Arianna Falbo. 2021. Slurs, neutral counterparts, and what you could have said. *Analytic Philosophy*, 62(4):359–375.

Nikhil Garg, Londa Schiebinger, Dan Jurafsky, and James Zou. 2018. Word embeddings quantify 100 years of gender and ethnic stereotypes. *Proceedings of the National Academy of Sciences*, 115(16):E3635–E3644.

William L. Hamilton, Kevin Clark, Jure Leskovec, and Dan Jurafsky. 2016. Inducing domain-specific sentiment lexicons from unlabeled corpora. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 595–605, Austin, Texas. Association for Computational Linguistics.

Leopold Hess. 2021. *Slurs: Semantic and Pragmatic Theories of Meaning*, Cambridge Handbooks in Language and Linguistics, page 450–466. Cambridge University Press.

Austin C. Kozlowski, Matt Taddy, and James A. Evans. 2019. The geometry of culture: Analyzing the meanings of class through word embeddings. *American Sociological Review*, 84(5):905–949.

Binny Mathew, Punyajoy Saha, Seid Muhie Yimam, Chris Biemann, Pawan Goyal, and Animesh Mukherjee. 2021. Hatexplain: A benchmark dataset for explainable hate speech detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(17):14867–14875.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Diana Rieger, Anna Sophie Kümpel, Maximilian Wich, Toni Kiening, and Georg Groh. 2021. Assessing the extent and types of hate speech in fringe communities: A case study of alt-right communities on 8chan, 4chan, and reddit. *Social Media+ Society*, 7(4):20563051211052906.

Koustuv Saha, Eshwar Chandrasekharan, and Munmun De Choudhury. 2019. Prevalence and psychological effects of hateful speech in online college communities. In *Proceedings of the 10th ACM Conference on Web Science*, page 255–264. Association for Computing Machinery.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 1–10, Valencia, Spain. Association for Computational Linguistics.

Ivan Vulić, Edoardo Maria Ponti, Robert Litschko, Goran Glavaš, and Anna Korhonen. 2020. Probing pretrained language models for lexical semantics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7222–7240, Online. Association for Computational Linguistics.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on Twitter. In *Proceedings of the NAACL Student Research Workshop*, pages 88–93, San Diego, California. Association for Computational Linguistics.

Michael Wiegand, Josef Ruppenhofer, Anna Schmidt, and Clayton Greenberg. 2018. Inducing a lexicon of abusive words – a feature-based approach. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1046–1056, New Orleans, Louisiana. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Kaya Yurieff. 2020. Reddit bans pro-trump forum the_donald and other communities that promote hate. *CNN Business*.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. SemEval-2020 task 12: Multilingual offensive language identification in social media (OffensEval 2020). In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1425–1447, Barcelona (online). International Committee for Computational Linguistics.

Ziqi Zhang and Lei Luo. 2019. Hate speech detection: A solved problem? the challenging case of long tail on twitter. *Semant. Web*, 10(5):925–945.

A Lists of lexical units for evaluation

HateBase: *african americans , albanians, american indian, americans, anchor babies, arabs, armenians, australians, azn, bimbo, bint, canadians, chav, chinaman, chinese people, ching chong, cunt, dutch people, faggy, filipinos, french people, gay people, gaylord, german people, ghey, goatfucker, gypsy, honky, hoodrat, immigrant, indian people, inuit, irish people, italian women, leb, mexican immigrants, middle-class people, mongoloid, native hawaiian, newfies, nigger, oklahoman, pacific islanders, palestinian, polack, polish people, porch monkey, protestants, race traitor, redneck, refugees, roman catholics, scally, seppo, shemale, shyster, slut, spics, sub human, taqiyya, trailer trash, twat, waspy, wetback, white trash, wigger, women, yokel, zio, zog*

HateXplain - persons: *asshole, bigot, bitch, boomer, boyfriend, brother, buddy, captain, chinaman, citizens, clown, cocksucker, commies, coons, coward, cuckservatives, cunts, doctors, driver, dykes, faggot, farmers, fascist, followers, friends, fuckers, girls, goatfucker, governor, haters, heeb, hero, hoes, honky, idiot, jigaboo, jockey, journalists, kids, ladies, lawyer, leader, leftie, loser, manager, moron, mother, mudshark, mudslime, muzrat, negress, negros, nigger, officers, officials, parents, partners, patriots, pedos, politician, prayers, president, princess, professor, protesters, pussy, queen, racists, raghead, rapefugees, rapper, redneck, residents, retard, sandniggers, satan, satanist, savages, scumbag, sheboon, sheriff, shitlib, shitskin, sjw, slave, slut, spics, students, taxpayers, teachers, towelhead, traitors, twat, veterans, warriors, wetbacks, whore, wigger, workers, yid*

HateXplain - random nouns: *action, aids, aliens, ape, army, ass, banislam, bat, beaners, bitch, bread, brown, charge, chinaman, code, commit, crack, cum, degeneracy, dicks, dislike, dumbass, faggotry, feminist, filth, friday, fruit, fuckers, gap, ghetto, girls, goatfucker, goy, head, hebrew, holohoax, homophobic, homosexuals, husband, illegals, infidels, jewish, khan, knowledge, lit, lmao, lottery, mans, mexicans, monkey, moslem, movement, mudslimes, muslime, muzrat, muzrats, nazi, negress, nig, niglet, noise, paki, pakis, pay, pedophile, pedophiles, pedophilia, players, porch, posts, prayer, prison, pro, propaganda, rag, raghead, rapist, redneck, refugee, ricky, savages, sheboon, shitskin, socialists, sort, steal, stupidity, subhuman, subversive, thot, thots, thru, trans, tweets, values, weird, wetbacks, wigger, witch, yid*

B Additional results of experiments

Sample size	Correlation	Classification report						
	Pearson's r (* = sig.)	Acc. (* = sig.)	Hateful			Neutral		
			Prec.	Recall	F1	Prec.	Recall	F1
1	0.786*	0.76*	0.76	0.86	0.80	0.78	0.64	0.70
2	0.790*	0.77*	0.76	0.86	0.80	0.78	0.65	0.71
3	0.788*	0.77*	0.76	0.86	0.81	0.78	0.65	0.71
4	0.791*	0.77*	0.76	0.86	0.81	0.79	0.65	0.71
5	0.790*	0.77*	0.76	0.87	0.81	0.79	0.65	0.71
6	0.789*	0.77*	0.76	0.86	0.81	0.78	0.66	0.71
7	0.790*	0.77*	0.76	0.85	0.81	0.78	0.66	0.72
8	0.789*	0.77*	0.76	0.86	0.81	0.79	0.65	0.71
9	0.790*	0.77*	0.77	0.85	0.81	0.78	0.66	0.72
10	0.789*	0.77*	0.76	0.86	0.81	0.78	0.66	0.72

Table 6: Performance results (average over 10 runs) for dimensions with different context sample sizes

Sentiment and Emotion Classification in Low-resource Settings

Jeremy Barnes
HiTZ Basque Center for Language Technologies
Ixa NLP Group, University of the Basque Country UPV/EHU
`jeremy.barnes@ehu.eus`

Abstract

The popularity of sentiment and emotion analysis has lead to an explosion of datasets, approaches, and papers. However, these are often tested in optimal settings, where plentiful training and development data are available, and compared mainly with recent state-of-the-art models that have been similarly evaluated.

In this paper, we instead present a systematic comparison of sentiment and emotion classification methods, ranging from rule- and dictionary-based methods to recently proposed few-shot and prompting methods with large language models. We test these methods in-domain, out-of-domain, and in cross-lingual settings and find that in low-resource settings, rule- and dictionary-based methods perform as well or better than few-shot and prompting methods, especially for emotion classification. Zero-shot cross-lingual approaches, however, still outperform in-language dictionary induction.

1 Introduction

Affective computing, including sentiment and emotion classification, has been research focuses inside of the Natural Language Processing (NLP) community for many years (Mohammad, 2016; Poria et al., 2023). This has lead to an incredible number of research directions and papers published on these topics, ranging from rule-based and dictionary-based approaches Turney (2002); Lee et al. (2010); Taboada et al. (2011); Staiano and Guerini (2014), to supervised training of deep learning models (Xu et al., 2019; Barbieri et al., 2022; Samuel et al., 2022) and finally to few-shot and prompting of large language models (Brown et al., 2020; Min et al., 2022; Plaza-del Arco et al., 2022). This also means that a systematic comparison of the benefits and weaknesses of models has not been performed, as each often individual papers compare only against more recent state-of-the-art models, and do not take into account previous approaches.

Like many other research areas in NLP, sentiment and emotion classification are dependent on domain and language-specific training data for optimal performance and this high-quality task-specific data is always in short supply as we apply our models to a constantly evolving set of scenarios.

The objective of this paper is therefore to identify trends in sentiment and emotion classification, especially regarding low-resource settings. As such, we attempt to address the following research questions:

- **RQ1**: Given a limited number of examples per class (<100), what method currently performs best?
- **RQ2**: Do these methods suffer domain transfer equally?
- **RQ3**: How well do these results hold for languages other than English?

To address these questions, we perform experiments[1] on 10 sentiment classification datasets and two emotion classification datasets in 8 languages with a number of low-resource approaches. Specifically, we compare dictionary-based methods, rule-based methods, few-shot methods and prompting methods on the English datasets. We simultaneously test the out-of-domain performance for each of the methods that demand training data. Finally, we also perform cross-lingual experiments.

We find that rule- and dictionary-based methods often perform on par with few-shot approaches in low-resource settings, especially on emotion classification and are more robust to domain changes, while prompting similarly provides promising results. Zero-shot cross-lingual approaches, however, still outperform in-language dictionary induction for languages other than English, suggesting that more work could be done in this area.

---

[1] Code to reproduce the experiments available at `https://github.com/jerbarnes/low_resource_sa_emo`.

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 290–304
July 14, 2023 ©2023 Association for Computational Linguistics

2 Related Work

Current state-of-the-art models for sentiment and emotion classification are dominated by language models that have been pretrained on large corpora and then fine-tuned for each specific task (Sharma et al., 2020; Barnes et al., 2022). Although ELMO (Peters et al., 2018), BERT (Devlin et al., 2019), and its variants were the first to provide evidence for the usefulness of language modeling as a transfer learning objective, there has since been an explosion and it is somewhat difficult to navigate which current models give the best performance on many datasets.

Besides the fully supervised setup, many larger language models also show signs of being able to learn a task with less data, allowing for non-trivial zero- or few-shot performance. The most common way to achieve this zero or few-shot ability is by prompting a model using a Natural Language Inference model, trained to determine whether a premise is true/false, given a hypothesis. This model can then be applied to new tasks by reformulating the input and labels (Schick and Schütze, 2021; Min et al., 2022).

For few-shot prompting, we can make use of the generative abilities of language models by providing demonstrations input/label pairs and asking for a final label (Brown et al., 2020; Lin et al., 2022). More recently, the results of models trained using instruction tuning suggests that these models generalize well to unseen tasks (Chung et al., 2022).

The same kinds of large language models trained on multilingual corpora also allow for *zero-shot cross-lingual transfer*, where a model is fine-tuned on a task in a high-resource source language and then tested on an under-resourced language (Pires et al., 2019; Conneau et al., 2020). However, these approaches have rarely been compared to previous dictionary-based methods.

2.1 Rule and Dictionary-based methods

Rule and dictionary-based methods are common for sentiment and emotion analysis, in part due to their simplicity and interpretability. Early work focused on automatically inferring polarity dictionaries for categorizing words (Hatzivassiloglou and McKeown, 1997) or texts (Turney and Littman, 2003; Kamps et al., 2004). Taboada et al. (2011) propose *SoCal*, one of the most popular rule-based methods for sentiment analysis, which uses a set of dictionaries with sentiment scores for certain parts of speech (adjectives, adverbs, nouns, intensifiers, and verbs) plus rules for interacting with negation, irrealis, and other sentiment shifting phenomena.

For emotion classification, there has been a good deal of work on creating dictionaries. Mohammad and Kiritchenko (2015) use word-association measures with emotional hashtags to create a large emotion dictionary from social media text, while Mohammad (2018) instead use best-worst scaling to crowdsource an emotion intensity dictionary. Buechel et al. (2016) adapt affective lexicons to historical German texts and use these to characterize emotional trends in various genres of writing across several centuries. Buechel et al. (2020) furthermore develop methods for inducing emotion dictionaries for 91 languages, but do not make use of these dictionaries for emotion classification.

For dictionary induction, Hamilton et al. (2016) propose a method to automatically induce domain-specific dictionaries and show their effectiveness across a number of historical and modern text classification tasks. An et al. (2018) similarly propose a method to create a semantic axis, *SemAxis*, in an embedding space and successfully create dictionaries for tasks beyond sentiment analysis, despite having small amounts of data available. In this approach, we create an average vector for positive V^+ and negative V^- sentiment by averaging the vectors for seed words from an embedding space, such as Word2Vec or FastText. We can then define the axis vector as the difference of the two:

$$V_{axis} = V^+ - V^-$$

To use the semantic axis that we have created, we can measure the cosine distance of another embedding and the semantic axis.

$$score(w)V_{axis} = \text{cosine dist}(w, V_{axis})$$

If the score is positive, we can assume the word is positive and vice versa, and expand the positive and negative seed dictionaries to cover all lemmas in the test set, effectively creating a high-coverage dictionary. We then use this dictionary to generate the semantic orientation score of a text.

However, most of these techniques have not been recently compared to what are considered state-of-the-art models under low-resource settings.

	Dataset	lang	Train	Dev	Test
Sentiment	MPQA	EN	987	337	299
	SemEval	EN	3,737	413	1,791
	OpeNER	EN	1,210	174	347
	OpeNER	ES	1,029	147	296
	GermEval	DE	6,444	772	1,490
	ASTD	AR	2,468	353	706
	NoReC	NO	2,675	516	417
	MultiBooked	EU	789	113	227
	NArabizi	DZ	564	75	92
	Maltese	MT	595	85	171
Emotion	SSEC	EN	2,329	583	1,956
	EnISEAR	EN	720	80	201

Table 1: Statistics regarding the sentiment and emotion datasets.

3 Data

In this section we describe the datasets that are used for experimentation. The statistics are shown in Table 1 (see Tables 6 and 7 in the Appendix for further details).

Sentiment datasets As we want to explore how well methods work across a number of domains and languages, we choose to explore binary sentiment classification. We use the binary version of the following datasets, where any strong positive/negative has been mapped to positive/negative and neutral has been removed. Using only binary sentiment classification allows for us to compare across a larger number of datasets and languages.

MPQA: Wiebe et al. (2005) annotate English news wire texts with a complex set of annotation types. We map the polarities to sentences and keep those sentences that contain a majority of one polarity, such that we have only positive and negative sentence-level annotations.

SemEval: The SemEval 2013 Shared Task 2 (Nakov et al., 2013) collected tweets and annotated them as positive, negative, or neutral. We keep only the positive and negative tweets.

OpeNER: Agerri et al. (2013) annotate English and Spanish (among others) hotel reviews for structured and aspect-based sentiment. We use the script from Barnes et al. (2018) to map these to sentence-level binary sentiment classification. **ES** is the Spanish data from this dataset.

AR: Nabil et al. (2015) annotate Arabic (both Modern Standard Arabic and various dialects) tweets. We remove the neutral and mixed classes.

DZ: Touileb and Barnes (2021) annotate Northern African Arabizi social media posts for sentiment. In this case, we use the transliterated Arabic script version of the dataset and remove the neutral class.

MT: The data for Maltese (Dingli and Sant, 2016; Cortis and Davis, 2019) comes from the combination (Martínez-García et al., 2021) of two smaller datasets.

DE: The GermEval 2017 Shared Task (Wojatzki et al., 2017) released annotated data for several subtasks on German social media texts. We use the document-level data (task B) and remove mixed and neutral.

EU: Barnes et al. (2018) annotate Basque hotel reviews for structured sentiment. We map these to sentence-level binary sentiment classification, using the script provided with the data.

NO: Velldal et al. (2018) provide a collection of professional reviews from news outlets. We keep the binary document-level data.

Emotion datasets For emotion classification we use the SSEC (Schuff et al., 2017) and EnISEAR (Troiano et al., 2019) datasets. The SSEC dataset reannotates a stance and sentiment dataset of political tweets with crowd-sourced labels for eight emotions (anger, anticipation, disgust, fear, joy, sadness, surprise, trust). The EnISEAR dataset, on the other hand, crowd sources descriptions of events tied to emotions (anger, disgust, fear, guilt, joy, sadness, shame), as well as how readers perceive these events.

For the SSEC, we separate 583 examples from the training set for development. For EnISEAR, we split the fully labeled data into train (70%), dev (10%), and test (20%). For EnISEAR, we use the crowd sourced annotations for emotion labels, rather than the prior emotion to align with SSEC. For both datasets, we take the view that any number of annotations is valid (the 0.0 strategy in SSEC) and accept any label that has been assigned to an example by at least one annotator.

4 Experimental Setup

In this section, we describe the approaches for three experimental setups (monolingual English sentiment classification, monolingual English emotion classification, and cross-lingual sentiment classification) from most resource intensive to least.

4.1 Sentiment classification

Supervised: To provide an upper-bound of fully supervised in-domain models, we use DistilBERT (Sanh et al., 2019), and RoBERTa base and large (Zhuang et al., 2021). To simulate low-resource scenarios, we train the same models with varying amounts of training data (200, 100, and 20 examples). We finetune these models for 5 epochs, with a learning rate of 2e-5, a weight decay of 0.01, and a batch size of 16 on a single Tesla T4 GPU. We take the best model on the development set for testing.

Few-shot: In this scenario, we assume we have a development set and a limited number of training examples (200, 100, 20). We train the same models in the same way as fully supervised training, but with the reduced training set size. We again take the best model on the development set for testing.

Prompting: In this scenario, we assume we have only a few training examples. We explore few-shot prompting (concretely 2-shot) using two OPT models (Zhang et al., 2022): namely, the 125 million and 1.3 billion parameter versions. We prompt these models by giving them 2 positive and negative examples with the following template (an example from the hotel domain):

(1) `I didn't like the hotel. Label: negative. We loved the hotel. Label: positive. {text}. Label:`

We take the first predicted token as the predicted label.

Rule-based: In this scenario, we assume no training data whatsoever. We compare these models with the rule-based SoCal system (see details in Section 2.1). This approach requires a large initial effort to create the rules and dictionaries, but afterward can be applied to new data without retraining.

Dictionary-based: Finally, we also compare simpler dictionary-based approaches which do not include rules, and instead rely on a simpler scoring procedure for each text:

$$score(\text{text}, D) = \frac{1}{|D|}\frac{1}{|\text{text}|}\sum_{w \in \text{text}} s_e(w, D)$$

where D is a sentiment dictionary, either containing a list of words with positive orientation D_{pos} or negative D_{neg}, and s_e is a function that returns 1 if a word w is in D, otherwise 0. The $score$ function therefore returns the average score of a text, normalized by the length of the text and by the length of the dictionary D. To predict the aggregate semantic orientation (positive or negative), we divide the positive score by the negative score

$$\text{semantic orientation} = \frac{score(\text{text}, D_{pos})}{score(\text{text}, D_{neg})}$$

If this orientation is greater than a certain λ, we will assume that the orientation is positive and return 1, otherwise we will assume it is negative, and return 0.

We can then use available sentiment dictionaries to estimate the semantic orientation of a text. For all dictionary-based methods, we further preprocess the texts by tokenizing and lemmatizing the text using spaCy.[2] For sentiment dictionaries, we use the available HuLiu dictionary (Hu and Liu, 2004), the NRC Hashtag sentiment dictionary (Mohammad et al., 2013), and the MPQA subjectivity and sentiment dictionary (Wiebe et al., 2005).

Dictionary induction: Finally, it is also possible to automatically create a sentiment or emotion dictionary from a small seed dictionary. In this case, we use the SemAxis method (An et al., 2018) with a small seed dictionary of 10 words per class. We limit the expansion of the dictionaries to tokens found in the test set and allow only words which have a cosine ≥ 0.15 to reduce likely noisy.

We compare the use of three embedding spaces to induce the new dictionaries: 200 dimensional GloVe embeddings trained on Twitter data (Pennington et al., 2014), 300 dimensional FastText embeddings trained on Wikipedia data (Bojanowski et al., 2017), and 300 dimensional FastText embeddings trained on Wikipedia and the GigaWord corpus[3] (Fares et al., 2017).

---

[2] Found at `https://spacy.io/`.

[3] These can be found at `http://vectors.nlpl.eu/repository/20/22.zip`

4.2 Emotion classification

Supervised and Few shot: Given that both the SSEC and EnISEAR datasets are multi-label, we train the models using a one-vs-all approach, effectively creating a binary version of the dataset for each emotion and training a binary classifier. Like the sentiment experiments, we use DistilBERT, RoBERTa-base, and RoBERTa-large. The training procedure is the same as with sentiment. We perform experiments with 200, 100, and 20 training examples for the few shot experiments.

Prompting: For prompting, we use the Flan T5 models (Chung et al., 2022) (base and large), which are instruction tuned models. We performed initial experiments with the same OPT models used for sentiment analysis, but found that the multi-label nature of emotion classification was better covered using the Flan T5 models. For prompting the SSEC dataset, we use the following template:

(2) `What emotions are found in this text (Anger, Anticipation, Disgust, Fear, Joy, Sadness, Surprise, Trust, None)?: {text}`

where `text` is the text to be classified. For EnISEAR, we replace the emotions with anger, disgust, fear, guilt, joy, sadness, and shame. We assume that any mention of these words in the generated text is a predicted label.

Dictionary-based: As emotion classification in the datasets we use is a multi-label task, we cannot use the semantic orientation score as is. Instead, we set a threshold value $\lambda = 1$ and predict any label where $score(\text{text}, D_{\text{emotion}}) > \lambda$. This allows for our dictionary-based approach to predict multiple labels.

We use the NRC emotion dictionary as an emotion dictionary (Mohammad and Kiritchenko, 2015), which contains 16,862 entries with annotations for 8 emotions (anger, anticipation, disgust, fear, joy, sadness, surprise, trust), which were compiled semi-automatically using word–emotion association scores with hashtags.

Dictionary induction: Similarly, we can use an adapted version of the SemAxis method to induce emotion dictionaries. In this setting, we create a semantic axis vector for each emotion we wish to propogate. For example, to create a semantic axis for 'anger' V_{anger}, we create the positive pole vector $V^{+}{}_{anger}$ by averaging the vector representation of seed words for 'anger' and the negative pole vector $V^{-}{}_{anger}$ by averaging the vectors of all other seed words.

Once we have the semantic axis vectors for each emotion, we can expand the original seed dictionaries by taking any word whose vector representation has a positive cosine distance with the semantic axis. As with sentiment, we take a conservative estimate and allow only words which have a cosine > 0.15 to reduce noise.

We then use the same prediction procedure as with the dictionary-based approach.

4.3 Cross-lingual generalization

We also compare zero-shot cross-lingual performance of multilingual large language models (MLLMs), in this case XLM-RoBERTa base and large, to dictionary induction. For the MLLM experiments, we train on one of the three English corpora (MPQA, OpeNER, and SemEval) and test the best model on the English development data on all non-English corpora.

For the dictionary induction experiments, we use the SemAxis method with FastText embeddings (Bojanowski et al., 2017), as these have embeddings available for most languages. For NArabizi (DZ), we use the embeddings trained on modern standard Arabic as a proxy.

4.4 Evaluation

For both sentiment and emotion classification datasets, we evaluate using Macro F_1, as the distribution of labels is unbalanced and we are interested in knowing how well the models perform on the less frequent labels as well.

5 Results

In this section we detail the results for sentiment classification, out of domain performance, emotion classification, and cross-lingual transfer.

5.1 Sentiment classification

Table 2 shows the Macro F_1 of the sentiment classification approaches on the English datasets (MPQA, OpeNER, and SemEval), as well as the average of all results per each approach.

The fully supervised upper-bound achieves an average F_1 of 91.2, showing strong performance for this binary classification task.

		MPQA	OpeNER	SemEval	Avg.
Supervised	DB	86.3	92.7	90.1	91.2
	RBB	87.2	94.4	91.0	
	RBL	92.0	95.3	91.5	
FewShot-200	DB	84.7	77.4	70.9	80.9
	RBB	80.8	93.4	86.8	
	RBL	67.9	80.5	86.2	
FewShot-100	DB	59.0	65.3	66.5	56.5
	RBB	62.5	81.9	45.3	
	RBL	38.5	31.1	58.1	
FewShot-20	DB	49.0	23.7	47.4	40.5
	RBB	36.9	42.1	42.2	
	RBL	39.3	42.1	42.2	
Prompted	OPT-125m	34.0	52.4	51.8	56.9
	OPT-1.3B	59.7	84.1	59.5	
Rules	SoCal	74.9	83.9	74.0	77.6
Dictionary	HuLiu	61.4	71.4	59.3	61.6
	NRC Hash	52.7	67.4	68.6	
	MPQA	60.7	60.2	52.5	
Induced	Twitter	61.9	65.1	67.7	61.8
	NLPL22	58.2	61.8	59.6	
	FastText	53.6	66.8	61.4	

Table 2: Results on sentiment analysis (MacroF1). DB: DistilBERT, RBB: RoBERTa-base, RBL: RoBERTA-large.

In the low-resource scenario, FewShot-200 is the best performing approach (80.9), followed closely by the rule-based SoCal (77.6). The dictionary-induction methods (61.8) and dictionary-based methods (61.6) achieve quite similar results, followed by prompting (56.9) and the few-shot methods using 100 examples (56.5) and 20 (40.5).

In general the RoBERTa-large model suffers more in the few-shot scenarios, losing 3.4-20 percentage points (pp) compared to RoBERTa-base. For prompting, however, the opposite is true, as the 1.3 billion parameter model performs 21.7 pp better than the 125 million parameter model. This ties in well with research indicating that the size of the language model leads to better few-shot performance (Brown et al., 2020).

Surprisingly, dictionary-based methods perform better than FewShot-100 or prompting large language models. Even more surprising is that inducing a sentiment lexicon from as few as 10 labeled words can outperform careful hand-designing of these dictionaries.

Approach	Avg. In	Avg. Out	$\overline{TL}_{A\rightarrow B}$
Supervised	91.2	84.6	13.3
FewShot-200	80.9	70.6	20.7
FewShot-100	56.5	47.4	18.1
FewShot-20	40.5	31.5	18.2

Table 3: We show the average in-domain results (Avg. In), average out-of-domain results (Avg. Out) and average domain transfer loss ($\overline{TL}_{A\rightarrow B}$) for the supervised models on English sentiment analysis.

Therefore, revisiting **RQ1**, we can say for binary sentiment classification, *fine tuning a model on as few as 100 examples per class gives competitive in-domain performance*. For anything less, *rule-based methods perform better*.

5.2 Out of domain performance of sentiment classification

Unlike prompting and dictionary-based approaches, supervised and few-shot methods are tied heavily to the domain they are trained with. In order to quantify the loss in performance of supervised models, we measure *domain transfer loss*, which is defined in Equation 1:

$$TL_{x\rightarrow y} = S_{x\rightarrow x} - S_{x\rightarrow y} \tag{1}$$

where $TL_{x\rightarrow y}$ is the difference of the Macro F_1 score $S_{x\rightarrow x}$ of a model fine-tuned on domain x and tested in the same domain, and the score $S_{x\rightarrow y}$ of the model fine-tuned on x and tested on domain y.

As we have two test domains $B = \{b_{domain1}, b_{domain2}\}$ for each training domain x, we average over these using Equation 2:

$$\overline{TL}_{x\rightarrow B} = \frac{1}{N_B}\sum_{\substack{i\in B\\ i\neq x}} S_{x\rightarrow x} - S_{x\rightarrow i} \tag{2}$$

		SSEC	EnISEAR	Ave.
Supervised	DB	74.6	72.1	67.6
	RBB	71.3	56.0	
	RBL	68.1	63.2	
FewShot-200	DB	55.5	62.8	54.1
	RBB	53.0	55.6	
	RBL	50.2	47.7	
FewShot-100	DB	45.6	47.3	45.8
	RBB	42.1	57.2	
	RBL	39.0	43.3	
FewShot-20	DB	42.8	43.3	41.6
	RBB	39.4	43.3	
	RBL	37.5	43.3	
Prompted	FlanT5-base	51.5	58.9	57.7
	FlanT5-large	47.6	72.6	
Seed Dict.		37.4	47.9	42.7
Dictionary	NRC	52.2	46.4	49.3
Induced	Twitter	62.0	53.2	54.4
	NLPL22	53.0	45.7	
	FastText	53.9	58.8	

Table 4: Macro averaged F_1 for emotion classification results on the SSEC and EnISEAR datasets. DB: DistilBERT, RBB: RoBERTa-base, RBL: RoBERTA-large.

Finally, we compute the average domain transfer loss for all models of a certain approach $A = \{\text{supervised, few shot}, \ldots\}$ by computing the average of the domain transfer losses $\overline{TL}_{x \to B}$ for all models in the approach:

$$\overline{TL}_{A \to B} = \frac{1}{N_A} \sum_{i \in A} \overline{TL}_{i \to B} \tag{3}$$

Table 3 shows the average in-domain results (Avg. In), average out-of-domain results (Avg. Out) and average domain transfer loss (TL) for the supervised models (the full results table can be found in Table 8 in Appendix A). Models finetuned in a supervised fashion achieve the best in-domain (91.2) and out-of-domain (84.6), with the smallest transfer loss (13.3).

Although FewShot-200 achieves relatively good in-domain performance (80.9), it has the largest transfer loss (20.7), with the out-of-domain performance dropping to 70.6, 7 pp. below SoCal. This suggests that it is highly dependent on the few training examples seen being in-domain and that it cannot be safely applied out-of-domain.

Finally, both FewShot-100 and FewShot-20 have similar transfer losses (18.1/18.2), although the already low in-domain performance (56.5/40.5) means that using these models either in-domain or out-of-domain is impractical.

In contrast, the prompting, rule-based, and dictionary-based approaches do not suffer from this and perform more consistently across domains.

Therefore, the answer to **RQ2** is that *rule-based methods perform better across domains that few-shot supervision methods.*

5.3 Emotion classification

Table 4 shows the Average Macro F_1 scores for all approaches on the two emotion classification datasets, as well as the averaged score per approach (results for each emotion can be found in Tables 9 and 10 in the Appendix).

Again, the fully supervised upper bound achieves the best F_1 (67.6), where DistilBERT achieves much better performance than either RoBERTa model. RoBERTA-base achieves poor performance on EnISEAR, RoBERTa-large consistently performs quite poorly, suggesting that it requires either more data or more careful fine-tuning than was used here.

The best performing method in the low-resource setting is prompting (57.7), followed by dictionary induction (54.4) and Few-shot 200 (54.1). The dictionary-based method, as well as the FewShot-100 and -20 approaches, perform quite poorly (49.3, 45.8, and 41.6 respectively), with the latter achieving worse performance than the 10 word per emotion seed dictionary (42.7).

In contrast to prompting OPT models for sentiment analysis, the FlanT5-large model does not consistently improve over the base model, achieving a quite low score on the SSEC dataset (47.6).

Similar to the sentiment experiment, the induced emotion dictionaries perform as well or better than previously compiled emotion dictionaries (NRC).

Returning to **RQ1**, *for emotion analysis prompting or dictionary induction perform better than few shot approaches.*

5.4 Cross-lingual sentiment classification

The results of the cross-lingual experiments can be seen in Table 5. In general, the XLM-RoBERTa models perform much better than the dictionary induction approaches (10-20 pp). However, this depends heavily on the source language corpus used to train, as several XLM-RoBERTa results are lower than their respective dictionary induction

Model	train		Test Lang							Avg. on Test
		self	DE	ES	AR	NO	EU	DZ	MT	
maj. baseline			46.2	45.2	33.5	41.8	45.8	39.1	39.4	41.6
XLM-RoBERTa-base	MPQA	87.1	65.9	89.9	68.8	74.3	80.2	52.8	54.9	69.5
	OpeNER	93.0	73.3	90.8	72.4	75.5	79.0	57.5	58.3	72.4
	SemEval	88.9	71.0	89.0	73.1	75.1	82.4	71.3	58.8	74.4
XLM-RoBERTa-large	MPQA	89.1	62.7	84.0	62.3	74.2	80.3	50.9	30.9	63.6
	OpeNER	95.6	72.8	93.8	77.1	82.9	87.2	72.2	40.2	75.2
	SemEval	90.9	67.6	88.4	75.0	77.0	83.6	76.9	51.2	74.2
FlanT5-base			69.9	77.9	36.3	43.9	14.5	26.4	44.5	44.8
FlanT5-large			73.1	93.4	89.7	86.7	90.9	97.6	82.6	87.7
dictionary induction			50.1	59.9	62.9	41.8	45.8	58.7	50.0	52.7

Table 5: Results on cross-lingual sentiment analysis (MacroF1).

approach (large trained on MPQA and tested on AR, DZ, or MT for example).

Curiously, the large version performs worse than the base version when trained on MPQA or SemEval. Like with the previous experiments, this may suggest that the larger models need more data or require more careful tuning than we performed in our experiments. In either case, it is important to note that simply increasing the size of the cross-lingual model will not necessarily result in better results.

Finally, the results of all models are generally worse for Narabizi (DZ) and for Maltese (MT), which is unsurprising, as they have little or no pretraining data in XLM-RoBERTA. The one exception is the Flant5-large, which achieves very good results on both. It is unclear what exactly causes this difference in multilingual ability, especially for low-resource languages like Narabizi and Maltese, although larger models are known to memorize training data (de Wynter et al., 2023) and both of these datasets are available in text format. Therefore, we cannot rule out data contamination as the source of such a jump in performance.

Finally, the cross-lingual models achieve an average of 71.5, compared to 66.3 for prompting or 52.7 for dictionary induction. Thus, we can cautiously venture that for **RQ3**, cross-lingual methods allow for the best results, although prompting larger multi-lingual LLMs may also provide good results in the future.

6 Conclusion and future work

In this paper we have performed experiments on 10 sentiment datasets and two emotion classification datasets in 8 languages with a number of low-resource approaches (dictionary-based methods, rule-based methods, few-shot methods and prompting methods). The main experiments were performed on the English language datasets (3 sentiment and 2 emotion), while further experiments were performed in 7 additional languages.

These results confirm that under ideal circumstances, fully supervised models perform much better than low-resource approaches. However, in low-resource settings (lack of training data, domain shift), these same models quickly lose performance and rule-based and dictionary-based approaches perform on par or even better if there is a domain shift involved.

While prompting achieved impressive performance in our experiments, given that the models were not explicitly trained, this came at a price. Namely, such approaches for languages other than English are currently not available or not on par with English versions. This area will surely be explored in the near future, but this current gap is nonetheless a product of the over-reliance on English in NLP.

The strong cross-lingual performance of the XLM-RoBERTa models suggests that cross-lingual approaches, especially those designed for adapting to new languages, scripts (Pfeiffer et al., 2021), or generally enabling ever more multilingual pretraining (Lauscher et al., 2020; Pfeiffer et al., 2022).

We find conflicting evidence on the importance of model size for low-resource performance. On the one hand, prompting the larger OPT model for sentiment classification gave consistently better results. On the other hand, RoBERTa-large suffered much more in out-of-domain classification and generally performed worse than RoBERTA-base on emotion classification in all data regimens. For prompting in emotion classification, FlanT5-large did not lead to consistent gains over the base version and finally, XLM-RoBERTa-large similarly performed worse than the base version on cross-lingual sentiment classification. This finding seems to indicate that some of the promised few-shot performance found in large language models is either lacking or requires careful tuning.

In the future, it would be interesting to expand this comparison to other dictionary induction methods, such as cross-lingual propagation (Buechel et al., 2020), or high-coverage expansion (Köper and Schulte im Walde, 2016). Given the promising results from the simple prompting approaches we used in our experiments, further research on how to expand these models to new languages and tasks would be of great use.

Finally, multi-lingual few-shot approaches (Lauscher et al., 2020) could also be compared, as it is often possible to use a few examples in the target language.

7 Limitations

In this paper, we only explore binary sentiment classification, as it is enables cross-lingual experiments to be somewhat comparable. However, this is a simplified task, which should be taken into account when interpreting the results. Our multilingual datasets also come from various domains and, although we try to control for this in English, this does lead to some effect in the results. Finally, for emotion detection, we only experiment in English.

We also chose only a few representative methods for each approach (few-shot, prompting, rule-based, etc). This was a necessary simplification given the large number of available models, and care was given to choose truly representative methods for each approach. However, some relevant methods may not be represented here.

Finally, we only report the results for a single run for the supervised models, rather than the average of 5-10 runs as is common. We compensate by averaging over results on several datasets and across several methods.

Acknowledgements

This work has been partially supported by the HiTZ center and the Basque Government (Research group funding IT-1805-22).

We also acknowledge the funding from the following projects: DeepKnowledge (PID2021-127777OB-C21) project funded by MCIN/AEI/10.13039/501100011033 and by FEDER Una manera de hacer Europa.

References

Rodrigo Agerri, Montse Cuadros Cuadros, Seán Gaines, and German Rigau. 2013. Opener: Open polarity enhanced named entity recognition. *Procesamiento del Lenguaje Natural*, 51(0):215–218.

Jisun An, Haewoon Kwak, and Yong-Yeol Ahn. 2018. SemAxis: A lightweight framework to characterize domain-specific word semantics beyond sentiment. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2450–2461, Melbourne, Australia. Association for Computational Linguistics.

Francesco Barbieri, Luis Espinosa Anke, and Jose Camacho-Collados. 2022. XLM-T: Multilingual language models in Twitter for sentiment analysis and beyond. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 258–266, Marseille, France. European Language Resources Association.

Jeremy Barnes, Toni Badia, and Patrik Lambert. 2018. MultiBooked: A corpus of Basque and Catalan hotel reviews annotated for aspect-level sentiment classification. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Jeremy Barnes, Laura Oberlaender, Enrica Troiano, Andrey Kutuzov, Jan Buchmann, Rodrigo Agerri, Lilja Øvrelid, and Erik Velldal. 2022. SemEval 2022 task 10: Structured sentiment analysis. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 1280–1295, Seattle, United States. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Sven Buechel, Johannes Hellrich, and Udo Hahn. 2016. Feelings from the Past—Adapting affective lexicons for historical emotion analysis. In *Proceedings of the Workshop on Language Technology Resources and Tools for Digital Humanities (LT4DH)*, pages 54–61, Osaka, Japan. The COLING 2016 Organizing Committee.

Sven Buechel, Susanna Rücker, and Udo Hahn. 2020. Learning and evaluating emotion lexicons for 91 languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1202–1217, Online. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Keith Cortis and Brian Davis. 2019. A social opinion gold standard for the Malta government budget 2018. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 364–369, Hong Kong, China. Association for Computational Linguistics.

Adrian de Wynter, Xun Wang, Alex Sokolov, Qilong Gu, and Si-Qing Chen. 2023. An evaluation on large language model outputs: Discourse and memorization.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Alexiei Dingli and Nicole Sant. 2016. Sentiment analysis on maltese using machine learning. In *Proceedings of The Tenth International Conference on Advances in Semantic Processing (SEMAPRO 2016)*, pages 21–25.

Murhaf Fares, Andrey Kutuzov, Stephan Oepen, and Erik Velldal. 2017. Word vectors, reuse, and replicability: Towards a community repository of large-text resources. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 271–276, Gothenburg, Sweden. Association for Computational Linguistics.

William L. Hamilton, Kevin Clark, Jure Leskovec, and Dan Jurafsky. 2016. Inducing domain-specific sentiment lexicons from unlabeled corpora. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 595–605, Austin, Texas. Association for Computational Linguistics.

Vasileios Hatzivassiloglou and Kathleen R. McKeown. 1997. Predicting the semantic orientation of adjectives. In *35th Annual Meeting of the Association for Computational Linguistics and 8th Conference of the European Chapter of the Association for Computational Linguistics*, pages 174–181, Madrid, Spain. Association for Computational Linguistics.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*.

Jaap Kamps, Maarten Marx, Robert J. Mooken, and Maarten de Rijke. 2004. Using wordnet to measure semantic orientations of adjectives. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, LREC '04, pages 1115–1118.

Maximilian Köper and Sabine Schulte im Walde. 2016. Automatically generated affective norms of abstractness, arousal, imageability and valence for 350 000 German lemmas. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2595–2598, Portorož, Slovenia. European Language Resources Association (ELRA).

Anne Lauscher, Vinit Ravishankar, Ivan Vulić, and Goran Glavaš. 2020. From zero to hero: On the limitations of zero-shot language transfer with multilingual Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4483–4499, Online. Association for Computational Linguistics.

Sophia Yat Mei Lee, Ying Chen, and Chu-Ren Huang. 2010. A text-driven rule-based system for emotion cause detection. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 45–53, Los Angeles, CA. Association for Computational Linguistics.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, Veselin Stoyanov, and Xian Li. 2022. Few-shot learning with multilingual generative language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9019–9052, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Antonio Martínez-García, Toni Badia, and Jeremy Barnes. 2021. Evaluating morphological typology in zero-shot cross-lingual transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3136–3153, Online. Association for Computational Linguistics.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Saif Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. In *Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013)*, Atlanta, Georgia, USA.

Saif M. Mohammad. 2016. Sentiment analysis: Detecting valence, emotions, and other affectual states from text. In Herb Meiselman, editor, *Emotion Measurement*. Elsevier.

Saif M. Mohammad. 2018. Word affect intensities. In *Proceedings of the 11th Edition of the Language Resources and Evaluation Conference (LREC-2018)*, Miyazaki, Japan.

Saif M. Mohammad and Svetlana Kiritchenko. 2015. Using hashtags to capture fine emotion categories from tweets. *Computational Intelligence*, 31(2):301–326.

Mahmoud Nabil, Mohamed Aly, and Amir Atiya. 2015. ASTD: Arabic sentiment tweets dataset. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2515–2519, Lisbon, Portugal. Association for Computational Linguistics.

Preslav Nakov, Sara Rosenthal, Zornitsa Kozareva, Veselin Stoyanov, Alan Ritter, and Theresa Wilson. 2013. SemEval-2013 task 2: Sentiment analysis in Twitter. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 312–320, Atlanta, Georgia, USA. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Jonas Pfeiffer, Naman Goyal, Xi Lin, Xian Li, James Cross, Sebastian Riedel, and Mikel Artetxe. 2022. Lifting the curse of multilinguality by pre-training modular transformers. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3479–3495, Seattle, United States. Association for Computational Linguistics.

Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2021. UNKs everywhere: Adapting multilingual language models to new scripts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10186–10203, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Flor Miriam Plaza-del Arco, María-Teresa Martín-Valdivia, and Roman Klinger. 2022. Natural language inference prompts for zero-shot emotion classification in text across corpora. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6805–6817, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, and Rada Mihalcea. 2023. Beneath the tip of the iceberg: Current challenges and new directions in sentiment analysis research. *IEEE Transactions on Affective Computing*, 14(1):108–132.

David Samuel, Jeremy Barnes, Robin Kurtz, Stephan Oepen, Lilja Øvrelid, and Erik Velldal. 2022. Direct parsing to sentiment graphs. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages

470–478, Dublin, Ireland. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Hendrik Schuff, Jeremy Barnes, Julian Mohme, Sebastian Padó, and Roman Klinger. 2017. Annotation, modelling and analysis of fine-grained emotions on a stance and sentiment detection corpus. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 13–23, Copenhagen, Denmark. Association for Computational Linguistics.

Chhavi Sharma, Deepesh Bhageria, William Scott, Srinivas PYKL, Amitava Das, Tanmoy Chakraborty, Viswanath Pulabaigari, and Björn Gambäck. 2020. SemEval-2020 task 8: Memotion analysis- the visuo-lingual metaphor! In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 759–773, Barcelona (online). International Committee for Computational Linguistics.

Jacopo Staiano and Marco Guerini. 2014. Depeche mood: a lexicon for emotion analysis from crowd annotated news. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 427–433, Baltimore, Maryland. Association for Computational Linguistics.

Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2011. Lexicon-based methods for sentiment analysis. *Computational Linguistics*, 37(2):267–307.

Samia Touileb and Jeremy Barnes. 2021. The interplay between language similarity and script on a novel multi-layer Algerian dialect corpus. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3700–3712, Online. Association for Computational Linguistics.

Enrica Troiano, Sebastian Padó, and Roman Klinger. 2019. Crowdsourcing and validating event-focused emotion corpora for German and English. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4005–4011, Florence, Italy. Association for Computational Linguistics.

Peter Turney. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 417–424, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Peter D. Turney and Michael L. Littman. 2003. Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems*, 21(4):315–346.

Erik Velldal, Lilja Øvrelid, Eivind Alexander Bergem, Cathrine Stadsnes, Samia Touileb, and Fredrik Jørgensen. 2018. NoReC: The Norwegian review corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2-3):165–210.

Michael Wojatzki, Eugen Ruppert, Sarah Holschneider, Torsten Zesch, and Chris Biemann. 2017. Germeval 2017: Shared task on aspect-based sentiment in social media customer feedback. In *Proceedings of the GermEval 2017 – Shared Task on Aspect-based Sentiment in Social Media Customer Feedback*, pages 1–12, Berlin, Germany.

Hu Xu, Bing Liu, Lei Shu, and Philip Yu. 2019. BERT post-training for review reading comprehension and aspect-based sentiment analysis. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2324–2335, Minneapolis, Minnesota. Association for Computational Linguistics.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

A Appendix

Dataset	lang	domain	Train	Dev	Test	Pos %
MPQA	EN	news	987	337	299	48.5
SemEval	EN	social media	3,737	413	1,791	72.2
OpeNER	EN	hotel reviews	1,210	174	347	72.7
OpeNER	ES	hotel reviews	1,029	147	296	82.6
GermEval	DE	social media	6,444	772	1,490	18.2
ASTD	AR	social media	2,468	353	706	50.2
NoReC	NO	reviews	2,675	516	417	67.1
MultiBooked	EU	hotel reviews	789	113	227	84.7
NArabizi	DZ	social media	564	75	92	52.0
Maltese	MT	social media	595	85	171	31.8

Table 6: Sentiment dataset statistics, including the percentage of positive examples for the sentiment datasets.

	lang	Train	Dev	Test	Anger	Anticipation	Disgust	Fear	Guilt	Joy	Sadness	Shame	Surprise	Trust
SSEC	EN	2,329	583	1,956	16.9	15.7	12.7	10.7	–	12.0	15.4	–	6.5	10.0
EnISEAR	EN	720	80	201	17.5	–	11.5	11.8	17.0	10.5	17.3	14.5	–	–

Table 7: Emotion dataset statistics, including the relative distribution of labels for the emotion classification datasets are also shown.

	Train	MPQA			OpeNER			SemEval		
	Test	MPQA	OpeNER	SemEval	MPQA	OpeNER	SemEval	MPQA	OpeNER	SemEval
Fully Supervised	DistilBert	86.3	84.4	82.3	77.5	92.7	85.6	66.8	91.5	90.1
	RoBERTa-base	87.2	90.3	87.1	79.7	94.4	88.4	82.7	94.0	91.0
	RoBERTa-large	92.0	90.4	86.3	75.2	95.3	87.2	78.6	94.8	91.5
FewShot-200	DistilBert	84.7	86.0	83.0	64.8	77.4	57.3	38.7	37.8	70.9
	RoBERTa-base	80.8	84.8	84.9	71.9	93.4	80.6	77.3	92.4	86.8
	RoBERTa-large	67.9	46.0	48.9	70.8	80.5	78.2	72.7	94.4	86.2
FewShot-100	DistilBert	59.0	54.6	52.7	59.3	65.3	60.8	47.3	38.4	66.5
	RoBERTa-base	62.5	44.3	44.8	57.9	81.9	66.4	29.3	42.1	45.3
	RoBERTa-large	38.5	42.0	42.4	45.4	31.1	46.8	43.2	35.4	58.1
FewShot-20	DistilBert	49.0	21.5	23.2	38.4	23.7	24.4	36.5	22.4	47.4
	RoBERTa-base	36.9	21.5	21.3	29.3	42.1	42.2	29.3	42.1	42.2
	RoBERTa-large	39.3	24.4	46.8	29.3	42.1	42.2	29.3	42.1	42.2

Table 8: Cross-domain results on sentiment analysis (Macro F_1).

		Anger	Anticipation	Disgust	Fear	Joy	Sadness	Surprise	Trust	Ave.
Supervised	DistilBERT	76.6	62.6	80.5	73.5	78.7	75.4	69.9	79.8	74.6
	RoBERTa-base	78.2	70.6	73.3	68.3	72.8	68.9	63.5	75.0	71.3
	RoBERTa-large	80.7	68.4	76.9	66.2	74.3	69.2	69.6	39.5	68.1
FewShot-200	DistilBERT	76.7	46.8	72.1	58.2	38.0	57.9	42.2	51.7	55.5
	RoBERTa-base	50.5	57.1	60.8	48.6	62.0	63.1	42.2	39.5	53.0
	RoBERTa-large	78.4	38.5	73.2	53.8	38.0	37.7	42.2	39.5	50.2
FewShot-100	DistilBERT	57.7	30.8	61.5	46.0	38.1	45.4	42.2	42.9	45.6
	RoBERTa-base	57.4	38.1	42.7	43.6	38.0	35.2	42.2	39.5	42.1
	RoBERTa-large	38.9	46.7	34.8	37.1	38.0	35.2	42.2	39.5	39.0
FewShot-20	DistilBERT	39.4	44.0	46.8	37.1	38.1	55.0	42.2	39.4	42.8
	RoBERTa-base	38.9	38.1	31.8	37.1	38.0	49.5	42.2	39.5	39.4
	RoBERTa-large	26.7	38.1	34.8	37.1	38.0	43.4	42.2	39.5	37.5
Prompted	FlanT5-base	64.1	38.4	43.9	58.7	49.9	52.1	47.9	57.0	51.5
	FlanT5-large	33.7	29.1	68.7	48.4	72.4	35.0	47.3	46.4	47.6
dictionaries	seed dictionary	29.4	35.0	35.9	40.3	40.0	32.0	43.6	43.3	37.4
	NRC	58.6	51.5	56.1	48.9	56.6	53.9	46.4	45.4	52.2
Induced	SemAxis	77.8	76.2	63.6	58.1	55.8	70.3	42.5	51.6	62.0
	NLPL22	48.6	55.1	51.6	55.8	61.0	49.7	47.4	54.7	53.0
	FastText	60.4	38.1	58.7	55.8	61.1	50.9	53.5	53.1	53.9

Table 9: Per class and Macro averaged F_1 for emotion classification results on the SSEC dataset.

		Anger	Disgust	Fear	Guilt	Joy	Sadness	Shame	Ave.
Supervised	DistilBERT	77.8	80.1	74.6	77.9	82.1	66.6	45.4	72.1
	RoBERTa-base	84.6	81.6	74.1	78.7	89.4	74.7	65.6	78.4
	RoBERTa-large	82.8	83.7	65.7	41.7	55.0	70.2	43.5	63.2
FewShot-200	DistilBERT	70.4	57.5	44.3	71.8	76.3	64.1	55.4	62.8
	Roberta-base	57.9	45.1	44.3	72.7	85.0	40.7	43.5	55.6
	Roberta-large	72.9	45.1	44.3	41.7	45.8	40.7	43.5	47.7
FewShot-100	DistilBERT	70.2	45.1	44.3	41.7	45.8	40.7	43.5	47.3
	Roberta-base	73.6	45.1	44.3	70.4	82.9	40.7	43.5	57.2
	Roberta-large	41.7	45.1	44.3	41.7	45.8	40.7	43.5	43.3
FewShot-20	DistilBERT	41.7	45.1	44.3	41.7	45.8	40.7	43.5	43.3
	Roberta-base	41.7	45.1	44.3	41.7	45.8	40.7	43.5	43.3
	Roberta-large	41.7	45.1	44.3	41.7	45.8	40.7	43.5	43.3
Prompted	FlanT5-base	60.3	54.8	62.9	43.7	80.9	64.8	44.8	58.9
	FlanT5-large	53.0	66.3	82.7	77.4	91.9	81.1	55.4	72.6
dictionaries	seed dictionary	41.4	45.2	59.2	48.1	50.7	44.6	45.8	47.9
	NRC	50.6	48.7	40.6	48.1	39.7	51.2	45.8	46.4
Induced	NLPL22	54.5	50.9	50.2	50.2	53.7	58.3	54.9	53.2
	FastText	22.4	70.6	18.6	55.9	40.7	61.9	50.0	45.7
	Twitter	49.7	64.8	71.2	53.3	57.5	55.5	59.8	58.8

Table 10: Per class and Macro averaged F_1 for emotion classification results on the enISEAR dataset.

Class	Seed Words									
Positive	good	nice	happy	beautiful	wonderful	enjoy	love	best	terrific	great
Negative	bad	mean	terrible	sad	ugly	hate	dislike	disgusting	worst	stressful
Anger	angry	mad	annoyed	hate	annoying	furious	upset	irritated	irritating	displeased
Anticipation	want	wanting	desire	anticipate	anticipating	wait	waiting	expect	expecting	hope
Disgust	yuck	disgusting	nasty	revolting	repulsive	despicable	nauseated	repugnant	shocking	vile
Fear	scared	afraid	fear	worried	worry	scary	dangerous	dark	panic	terror
Joy	happy	content	joyful	fun	cheerful	cheerfulness	cheer	delighted	ecstatic	elated
Sadness	sad	unhappy	melancholy	sorrowful	sorrow	gloomy	gloom	pessimistic	heartbroken	depressed
Surprise	wow	surprise	surprised	amazed	gobsmacked	stunned	shocked	dazed	astonished	startled
Trust	trust	trustworthy	confidence	confident	sure	faith	conviction	convinced	belief	truthful
Guilt	guilt	guilty	culpability	disgrace	regret	remorse	penitence	remorseful	sorry	wrong
Shame	ashamed	embarrassed	embarrassing	humiliating	humiliated	stigma	scandal	scandalous	shame	shameful

Table 11: Seed dictionaries for each class.

Analyzing Subjectivity Using a Transformer-Based Regressor Trained on Naïve Speakers' Judgements

Elena Savinova and **Fermín Moscoso del Prado Martín**
Centre for Language Studies, Radboud University, Nijmegen, the Netherlands
{elena.savinova, fermin.moscoso-del-prado}@ru.nl

Abstract

The problem of subjectivity detection is often approached as a preparatory binary task for sentiment analysis, despite the fact that theoretically subjectivity is often defined as a matter of degree. In this work, we approach subjectivity analysis as a regression task and test the efficiency of a transformer RoBERTa model in annotating subjectivity of online news, including news from social media, based on a small subset of human-labeled training data. The results of experiments comparing our model to an existing rule-based subjectivity regressor and a state-of-the-art binary classifier reveal that: 1) our model highly correlates with the human subjectivity ratings and outperforms the widely used rule-based *pattern* subjectivity regressor (De Smedt and Daelemans, 2012); 2) our model performs well as a binary classifier and generalizes to the benchmark subjectivity dataset (Pang and Lee, 2004); 3) in contrast, state-of-the-art classifiers trained on the benchmark dataset show catastrophic performance on our human-labeled data. The results bring to light the issues of the gold standard subjectivity dataset, and the models trained on it, which seem to distinguish between the origin/style of the texts rather than subjectivity as perceived by human English speakers.

1 Introduction

The task of subjectivity detection refers to identifying opinions, attitudes, beliefs and private states in a given text. Subjectivity detection as a task has received a lot of attention over the past decades, resulting in an abundance of methods and tools for subjectivity analysis. While in the earlier works, subjectivity was detected using rule-based approaches employing subjectivity lexicons (Riloff et al., 2003; Wiebe and Riloff, 2005), in the more recent studies, subjectivity detection is often approached with machine learning classifiers trained on existing gold standard datasets annotated for subjectivity (Huo and Iwaihara, 2020; Zhao et al., 2015).

Despite a relatively large body of literature on the topic, subjectivity detection has often been perceived as a preparatory step for sentiment analysis, that is, detection of positive or negative polarity of texts (Chaturvedi et al., 2018; Liu, 2010). Accurate sentiment analysis relies primarily on subjective fragments of the text. For this reason, subjectivity detection has been mostly viewed as a binary classification task. However, given the complex nature of the interplay of viewpoints in texts and numerous ways of expressing oneself with varying intensity, subjectivity can also be considered a gradual measure. To the best of our knowledge, the most widely used tool for subjectivity analysis that offers a gradual subjectivity estimate is *pattern* regressor (De Smedt and Daelemans, 2012), which is rule-based, and therefore, less accurate compared to state-of-the-art systems.

In the present paper, we approach the problem of identifying subjectivity as a regression task and use a semi-supervised approach to train a task-agnostic transformer model (RoBERTa) to produce sentence-level subjectivity scores based on a small subset of human annotations. The data that we use for training are a dataset of news articles and social media news posts produced by major UK news sources, with a small subset of it labeled by native English speakers. We describe the training procedure and compare the results of the model to the average native speaker's judgements, and to the widely used rule-based *pattern* regressor (De Smedt and Daelemans, 2012) to demonstrate that our model converges with native speaker intuitions and outperforms the rule-based regressor. The model is further evaluated as a binary classifier on our dataset and on the benchmark subjectivity dataset (Pang and Lee, 2004), showing good performance and generalizability to other discourse types. Finally, we show that our model generalizes better

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 305–314
July 14, 2023 ©2023 Association for Computational Linguistics

to other discourse types than the current state-of-the-art systems trained on the gold standard subjectivity dataset (Pang and Lee, 2004). These results highlight the importance of relying on actual human annotations rather than automatic labeling for compilation of subjectivity datasets and open further discussion about the nature of subjectivity models trained on the gold standard dataset.

2 Related work

The problem of sentence-level subjectivity detection in the previous literature has been approached in two ways. The traditional approach is rule-based and uses subjectivity lexicons and linguistic pattern extraction to define subjective and objective text fragments (Riloff and Wiebe, 2003; Riloff et al., 2003; Wiebe and Riloff, 2005; De Smedt and Daelemans, 2012). A more recent state-of-the-art approach uses machine learning based text classification algorithms to detect subjectivity (e.g., Hube and Fetahu, 2019; Huo and Iwaihara, 2020; Lin et al., 2011; Sagnika et al., 2021; Zhao et al., 2015). Although training deep neural models can give impressive results, they require large annotated datasets and substantial computational resources, which are not always available. To overcome this issue, recent studies on subjectivity detection started employing pretrained language representation transformer models, such as BERT (Devlin et al., 2019), and fine-tuning them for subjectivity classification task, which showed very promising results (Huo and Iwaihara, 2020; Kasnesis et al., 2021; Pant et al., 2020). For instance, Kasnesis et al. 2021 repot an impressive accuracy of 98.3% in subjectivity detection in the benchmark subjectivity dataset (Pang and Lee, 2004) using a method based on ELECTRA-large transformer (Clark et al., 2020).

In the present work, we adopt the approach of using a pretrained task-agnostic language model and fine-tuning it on subjectivity detection task for our own news dataset. The dataset consists of Facebook news posts and online news articles produced by four major UK news sources, with the total size of 7,751 sentences. Subjectivity of utterances is known to be a gradual factor in the cognitive theoretical accounts of subjectivity (Langacker, 1990; Traugott, 1995); some utterances are perceived as more subjective than others. For this reason, in contrast to many previous studies on automatic subjectivity analysis, we approached the problem of subjectivity detection as a regression rather than a binary classification problem. To do so, we trained our model on a subset of our data that have been annotated for the degree of subjectivity by human raters using a 7-point scale.

In the previous literature, benchmark datasets used for training subjectivity detection models were often obtained via annotations based on certain guidelines. For example, one of the earliest resources, namely MPQA Opinion Corpus (Wiebe et al., 2005), was compiled following a precise event- and entity-level annotation scheme for what is considered subjective and objective. In languages other than English, e.g., in Italian corpus *subjectivITA* (Antici et al., 2021), sentence-level subjectivity annotations were also obtained by asking annotators to follow specific guidelines on what should or should not be considered subjective. Although the guidelines are often theory-based, it is difficult to estimate how they relate to the actual native speakers' intuitions. For example, telling annotators to label third person attitudes and beliefs as objective reflects a certain theoretical choice but may not reflect language users' perceptions (e.g., *"According to the guests, the show was extremely unprofessional"*). Even more difficult to relate to human judgements are automatically collected subjectivity datasets, such as the benchmark SUBJ dataset (Pang and Lee, 2004), which is a widely used dataset for model training and evaluation. This dataset contains 5,000 movie review snippets that are automatically labeled as subjective and 5,000 sentences from plot summaries that are automatically labeled as objective. However, a closer look at this dataset reveals many cases where objectivity of the sentences taken from the movie plot summaries is questionable (e.g., *"What better place for a writer to pick up a girl?"* is considered as being objective). In the present work, we train our model on subjectivity annotations by native speakers who were not asked to follow any guidelines except for brief definitions of subjective ("expressing opinions, attitudes and beliefs") and objective ("stating factual information"), which means that our model results represent how subjectivity would be perceived by naïve language users. Similar approach to obtaining annotations with only definitions of subjective and objective as guidelines was used in the compilation of a Czech subjectivity dataset (Přibáň and Steinberger, 2022).

3 Method

3.1 Dataset

The dataset contains articles and Facebook posts on the topics of “crime” and “Covid-19” by four major UK news sources: two “popular” newspapers focused on soft news content (*Daily Mail* and *Metro*) and two “quality” sources focused on hard news (*BBC News* and *Sky News*). Since the dataset was collected for the purpose of analyzing subjectivity in the news across different types of sources (quality and popular) and media channels (articles on the websites and Facebook posts), the topics of "crime" and "Covid-19" were chosen to ensure comparability between the popular and the quality sources, as these topics are covered by both types of sources. First, we acquired Facebook posts of the four sources using Facepager app (Jünger and Keyling, 2019). Around 2,000 posts per news source page were randomly selected, setting the app limits to 100 posts per page, 20 pages and a one-year time limit, meaning the posts were published between December 7, 2020 and December 7, 2021. The information collected included the text of the post, the news headline (if present), the link to the original news item (if present), date and time of publication. Topic selection for "crime" and "Covid-19" topics was performed using keywords (see Appendix A). During topic selection for crime news, several keywords for exclusion of items were used to make sure that the news items did not include stories about natural disasters or TV soap operas. At the preprocessing stage, the posts were split into sentences using the Python package NLTK (Bird et al., 2009).

After the Facebook posts were selected, 84 news articles were randomly chosen out of those posts that had a corresponding link to the original news item (21 articles per source, equal distribution of topics), in an attempt to match the articles and the Facebook posts subparts of the dataset in the number of words. The text of the news articles was scraped from the websites (using the Python package *beautifulsoup*; Richardson, 2007). The articles were also preprocessed and split into sentences using NLTK (Bird et al., 2009). The resulting dataset includes 4,778 sentences taken from Facebook posts (65,058 words) and 2,973 sentences taken from articles on the websites (72,236 words), including headlines in both cases.

3.2 Annotation experiment

A random subset of 400 sentences (controlled for equal distribution of topics, news sources and media channels) was selected from the dataset for the annotation experiment. We used Prolific to collect annotations from 20 native speakers of English. We semi-randomly split the subset of 400 sentences into 20 experimental lists matching the number of annotators, in such a way that every annotator received 100 sentences for labeling and every sentence was labeled by 5 different speakers. The participants were instructed to evaluate subjectivity of the sentences on a 7-point scale, with extremes marked as “objective” and “subjective”. They were informed that the sentences were taken from news articles on the newspaper websites and news posts on social media, and that some sentences are headlines. The participants were given simple conceptual definitions of the terms, namely, they were told that “subjective” meant “expressing personal opinions, emotions, feelings and tastes, hopes and wishes, self-made conclusions (e.g., “This is awful”)”, while “objective” meant “reporting facts, events, conclusions supported by data (e.g., “The President had a meeting with the Prime minister”)”. There were four attention checks asking participants to select a specific answer option and four comprehension checks representing clearly subjective (“This is very beautiful”) and objective (“London is the capital of the UK”) sentences that were expected to be rated with 7 and 1, respectively. Only those participants who passed all the attention checks were paid for participation (4 GBP) and only those who also passed the comprehension checks were included in the dataset. One participant failed to pass the attention checks; additionally, two participants failed to pass the comprehension checks. After rejecting a participant, their list was reposted to Prolific until all 20 lists were successfully annotated. The mean age of participants in the final dataset was 36 (SD=15, range 19-67). The experiment was approved by the Ethics Assessment Committee Humanities of Radboud University (reference number 2022-9393).

Since our participants each rated a different subset of 100 sentences from all other participants, in order to estimate the inter-rater agreement, we computed the correlation of each participant’s ratings with the mean of the remaining participant’s ratings. We chose a correlation score of r=.4 as an inclusion threshold, leaving out one participant whose score was lower than .4. The mean correlation score of the remaining 19 raters was *r*=.64. We also excluded two sentences from the annotated subset as those were discovered to be duplicates (although in the full dataset these sentences come from different news items, they share the same text: “BREAKING”). For the remaining 398 sentences rated by 19 subjects, we computed mean scores and standard deviations. Figure 1 shows the distribution of standard deviations over the scores: as expected for this type of data, the more extreme scores have smaller deviations since people tend to agree on what is clearly subjective and objective, while the scores towards the middle have larger deviations reflecting weaker agreement among raters.

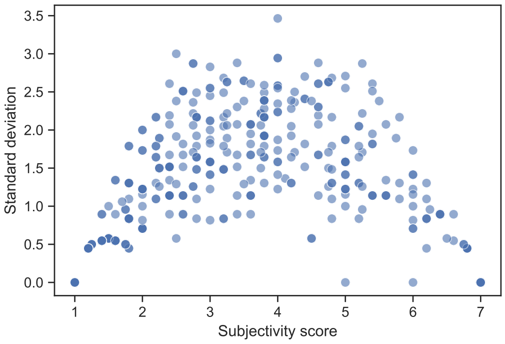


Figure 1: The distribution of standard deviations over mean subjectivity scores for annotated sentences.

3.3 Model training

In order to improve the performance of our text classifier/regressor, we began by fine-tuning the robustly optimized BERT transformer RoBERTa-base (Liu et al., 2019), which is based on dynamic masked priming, by adapting it to the unlabeled part of our dataset using the *simpletransformers* library (Rajapakse, 2019). Doing this for just a single epoch provided a small improvement in the final regression/classification results. Subsequently, we trained a text regression model on the labeled subset of our data using our version of RoBERTa-base fine-tuned to our specific dataset. We split the labeled data into a training set (298 sentences), a validation set (50 sentences) and a test set (50 sentences). For each of the 298 sentences, the model was trained to produce an average rating for that sentence provided by the human annotators. The human rating scores were normalized from the 7-point scale into a [0-1] scale. The convergence of the evaluation loss indicates that 20 epochs are sufficient training for this model (Figure 2). After testing the model’s performance, it was applied to the full dataset to obtain subjectivity scores per sentence. The dataset with sentence-level subjectivity scores predicted by our model is available online.

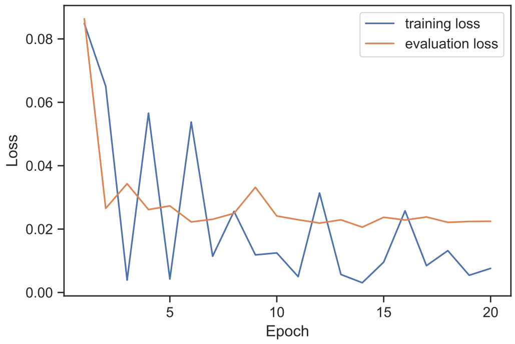


Figure 2: Training and evaluation loss.

4 Results

4.1 Evaluating the model as a text regressor

Our model’s predictions on the test set show that there is a very high correlation with the average human ratings (*r*=.79), accounting for over 62% of the variance. Figure 3 shows a plot of the correlation between the model’s prediction and the true human ratings. Beyond being quite a good correlation, this is above the correlation achieved by any of the raters with the average of the remaining raters (the maximum achieved by the raters was *r*=.76, the average correlation was *r*=.64, and the median correlation was *r*=.67). In other words, our model is a better match to the average human rater than any of the human raters was.

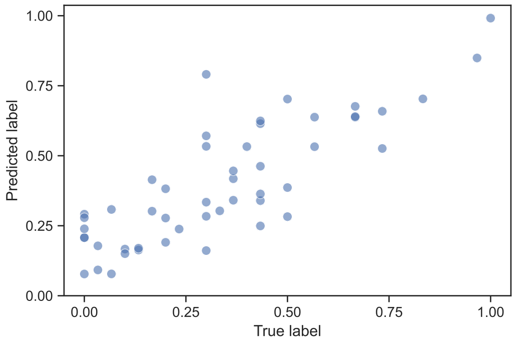


Figure 3: Correlation between our model’s predictions and the human ratings on the test set.

For comparison, we provide subjectivity annotation with TextBlob (using *SpacyTextBlob*; Loria,

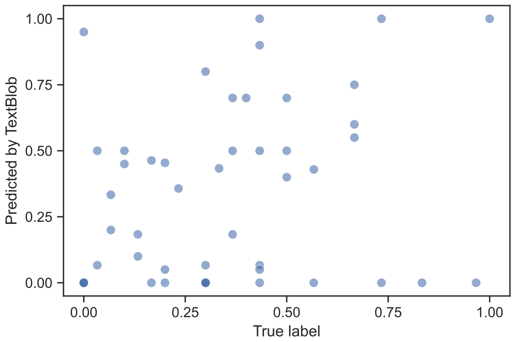


Figure 4: Correlation between *pattern*'s (TextBlob) predictions and the human ratings on the test set.

2018) as a baseline performance. TextBlob uses the rule-based sentiment valence and subjectivity tagger from the *pattern* library described in De Smedt and Daelemans (2012). This is one of the most popular sentiment analyzers for English. In addition to sentiment, it provides subjectivity ratings on a scale from 0.0 (totally objective) to 1.0 (totally subjective). We evaluate *pattern*'s performance on our test set by comparing *pattern*'s subjectivity ratings with the average ratings provided by our annotators. The correlation plot is presented in Figure 4. As we can see from the plot, *pattern*'s predictions correlate very poorly with human raters (r=.28), accounting for barely 8% of the variance in the rating means. Thus, our model substantially outperforms a widely used tool for subjectivity annotation based on regression. To give an example for comparison, a sentence *"Rose West was convicted of 10 murders in November 1995 and is serving life"* from our test set received an average score of 0 by the human annotators (which corresponds to 1 on the 7-point scale), suggesting that it was perceived as objective. While our model predicted a similar subjectivity score of .08 for this sentence, the *pattern* regressor estimated it as very subjective with a score of .95. From the above, we can conclude that using our model for tagging a corpus should result in an annotation that would be at least as good as an annotation that would be obtained if a single person rated all sentences for subjectivity.

4.2 Additional test set

In order to test our model's performance further, we collected an additional human-labeled test set by randomly selecting 100 sentences from the unlabeled part of our dataset. We obtained subjectivity ratings for theses sentences from 5 native English speakers (M=29, SD=7, range 19-35) using the same procedure as in the first annotation experiment (Section 3.2). Comparing each participant's ratings and their correlation with the mean of the other participants' ratings led to exclusion of one outlying participant, whose correlation with the others was below .4 threshold. The mean correlation score of the remaining participants was r=.60. Since all participants rated the same set of items in this experiment, we also computed an intraclass correlation coefficient (ICC) to estimate inter-rater agreement between participants. The ICC estimate at 95% confidence level using a two-way random-effects model (using the *pingouin* Python package; Vallat, 2018) was .41 for a single rater, suggesting fair bordering with poor reliability of raters as individuals. The ICC estimate for the average of raters was .74 indicating moderate bordering with good inter-rater agreement. The correlation between the average human ratings and our model's predictions are presented in Figure 5. The correlation score was r=.61, explaining 37% of the variance. Although this score is lower than the score obtained for the original test set, it is just above the mean correlation between the raters of this additional set (r=.60). Therefore, as in the previous tests, our model is indeed a very good model of the average human rater. Notice that one should not expect the model to show very high correlation scores with the raters' evaluations when the raters themselves do not agree on the evaluation of these sentences, as is indicated by the low inter-rater agreement scores. Further research is needed to investigate whether there are natural clusters among the raters which would imply that there are different ways of understanding what subjectivity is among English speakers. We believe that the performance of our model on the additional test set is not surprising given the low level of agreement among human raters themselves, and together with relatively good performance on the benchmark dataset (see Section 4.4 below), indicates that our model is a good subjectivity predictor.

4.3 Evaluating the model as a binary classifier

Although we have trained the model as a text regressor, it can also be used as a classifier, by discretizing the continuous scores on one or more thresholds for both the true and the predicted labels. Given that the human annotators were instructed to rate subjectivity on a 7-point scale, it is clear

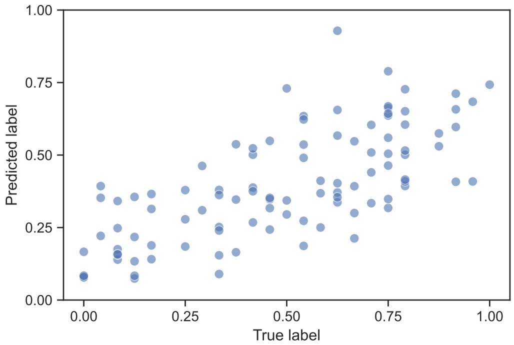


Figure 5: Correlation between our model's predictions and the human ratings on the additional test set.

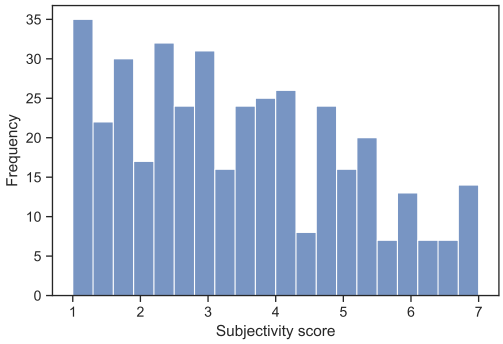


Figure 6: The distribution of subjectivity scores per sentence in the labeled set.

that anything with an average rating above .5 (i.e., above 4 on the 7-point scale), was considered as more subjective than objective by the annotators, and therefore, human-labeled data can be split on the .5 threshold. On the other hand, our model was trained on minimizing the mean squared error between predictions and true labels, and not on classification. Furthermore, the data in the training set were unbalanced towards objective labels (Figure 6). For this reason, one might want to consider a threshold different from .5 for taking a prediction of the model as subjective. We found the optimal threshold value for the model output by considering the model's predictions and optimizing the value of the F1 score (for the minority category "subjective") as a function of the threshold value. The result of this optimization can be seen in Figure 7. It shows that taking a classification threshold of θ=.6245, leads to the optimal classification behavior with an impressive accuracy of 92% and F1=.80. This is a slight improvement over the classification that would be obtained by a plain .5 threshold, which still leads to a very good classification performance with an accuracy of 86% and F1=.74. The overall performance of our model as a classifier can be appreciated in the diagonal confusion matrix (Figure 8).

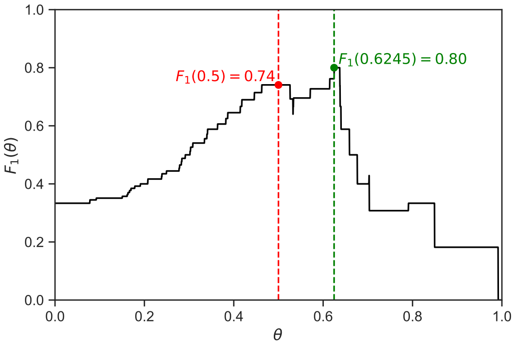


Figure 7: Threshold value for the binary classifier as a function of F1 score.

4.4 Evaluation on the benchmark subjectivity dataset

As mentioned above, the most used dataset for evaluating subjectivity labels is the SUBJ dataset introduced in Pang and Lee (2004). This dataset contains 10,000 short texts. Of these, 5,000 – automatically labeled as subjective – are movie review snippets (e.g., *"bold, imaginative, and impossible to resist"*) from www.rottentomatoes.com. The remaining 5,000 – automatically labeled as objective – are sentences from plot summaries taken from the Internet Movie Database (IMDb). This dataset is generally taken as a gold standard for subjectivity. However, although there is a clear correlation between subjectivity and the source of the text (review snippets being subjective vs. plot sentences being objective), one can find many examples in the dataset where this assumption fails. For instance, the IMDb sentence: *"What better place for a writer to pick up a girl?"* is labeled as objective in the SUBJ dataset, but the objectivity of this sentence is rather questionable. In all fairness, such a sentence might indeed have been objective in the context of the plot summary in which it appeared but, without such context, as it appears in the dataset, it is less clearly objective. Such examples of not-so-objective IMDb sentences abound in the SUBJ dataset. The opposite, however, is less common, with the snippets taken from www.rottentomatoes.com appearing consistently subjective, at least on visual inspection.

Comparing a variety of traditional (i.e., non-

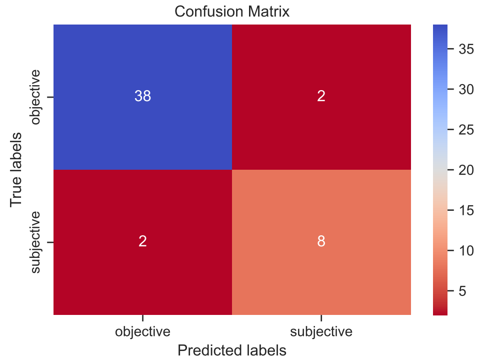


Figure 8: Confusion matrix of our model on the test set.

deep-learning) methods, Wang and Manning (2012) report maximum accuracy of 93.6% in tagging this corpus. In a recent review, Kasnesis et al. (2021) raise this maximum accuracy to 98.3% using a method based on the ELECTRA-large transformer. Our text regression model was trained on a completely different type of texts. Such texts were also used for setting the classification threshold. Despite the mislabeling present in the SUBJ dataset, it is still interesting to evaluate how our model performs on the test set of the SUBJ dataset. The density plot of the predicted subjectivity scores is presented in Figure 9.

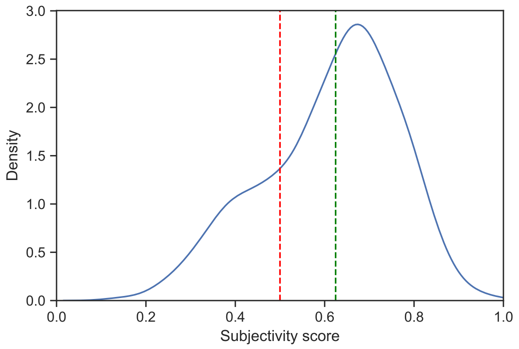


Figure 9: Density plot of our model's predictions on the test set of the SUBJ dataset.

As mentioned above, it appears that many of the "objective" sentences in the SUBJ dataset are in fact more than a bit subjective. The opposite (i.e., rather objective sentences labeled as "subjective") is less common in the dataset. If this intuition is true, and our model captures the actual subjectivity of the texts in the corpus, we should expect to see that the predictions of our model are visibly skewed towards the subjective, instead of being balanced as it is assumed in the dataset design. The kernel density estimate plot of our model's predictions (Figure 9) confirms this intuition: there are substantially more subjectively labeled sentences than one would have expected in a balanced labeled corpus. The green dashed line on the density plot denotes the optimized classification threshold, and the red dashed line plots the suboptimal .5 classification threshold. The slight shoulder on the left side of the density plot is a trace of bimodality. This bimodality arises because, on average, the sentences from movie plots are indeed more objective than the movie review snippets, but this is far from a clear-cut distinction in terms of objectivity.

If we use the classification threshold that we established on our own testing set, without further optimization, we obtain an accuracy of 78.2%, and an impressive F1=.79 on the SUBJ test set, just slightly below what we obtained for our own testing set. This is remarkable, considering that the SUBJ dataset is substantially different from the dataset that we trained our model on. Even if we had chosen to keep the suboptimal classification threshold at .5, we would still obtain an accuracy of 69.8% and a very good F1=.76. Examining the confusion matrix for this dataset with the optimized threshold value (Figure 10), we find that the accuracy especially suffers from cases that were labeled as "objective" in the corpus, but our model in fact considers them subjective. However, if we bear in mind the mislabeling present in SUBJ dataset that we discussed above, these might in fact not be errors, but sometimes cases where our model is actually outperforming the supposed gold standard. For instance, the aforementioned sentence *"What better place for a writer to pick up a girl?"*, which is labelled as objective in the SUBJ dataset, but appears subjective to us, is given a subjectivity score of .66 by our model. Thus, this sentence is evaluated as unclear but slightly subjective by our model. To us this appears to be a better assessment of this sentence's subjectivity than the gold standard label of "objective". To give another example, a sentence *"Moving cross country isn't even a problem for her"* is tagged as objective in the SUBJ dataset. Without the context, this sentence seems to represent an opinion/judgement, which is in essence subjective. Our model's prediction for subjectivity of this sentence is .78, which, in our opinion, is a more accurate estimate.

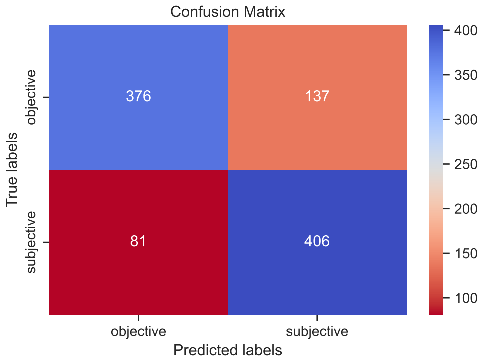


Figure 10: Confusion matrix of our model on the test set of the SUBJ dataset.

4.5 Performance of state-of-the-art classifiers on our data

At an accuracy of 78.2% and an F1 of .79, our system performs relatively well on the SUBJ dataset benchmark. Nevertheless, this performance is well below the top performance (with an accuracy of 98.3%) reported by Kasnesis et al. (2021) for the same dataset. We suspect, however, that the outstanding performance of subjectivity classifiers trained on this dataset is in fact misleading. As we have seen, many of the examples (certainly more than 2%) in this gold standard are actually mislabeled with respect to subjectivity itself. This suggests that such top performing systems, rather than learning to distinguish subjective from objective passages, are in fact learning to distinguish the language used in movie review snippets from the language used in movie plots. The fact that this distinction indeed correlates with subjectivity explains why our system, trained on data explicitly labeled for subjectivity, is still able to perform well on this dataset.

To investigate this further, we trained a two-way classifier (based on a distilBERT-base-uncased transformer) on the 8,100 training passages of the SUBJ dataset, using an additional 900 passages as a validation set. Our system performed slightly below the best reported performances, with an accuracy of 93.5% and F1=.93 on the 1,000 test passages from the SUBJ test set. We did not spend much time optimizing this system because we believe that improving the system's performance would not lead to results much different from those we report below. Once this model was trained, we tested the model on the 50 test sentences from our human-labeled dataset.

The two-way classifier seemed to perform relatively well on the 50 test sentences from our dataset, with an accuracy of 75.5%. However, examining the performance in more detail revealed that on our dataset (which reflects human subjectivity ratings), the model obtained a dismal F1=.25 in classifying subjective sentences. The very low F1 is explained by the confusion matrix below (Figure 11): the model shows more false positives and misses than it shows hits in labeling a sentence as subjective. This confirms our suspicion that the outstanding performance of this model on the SUBJ dataset reflects not the fact that the model is a good classifier of subjectivity, but rather the fact that this model instead learned how to distinguish the language in movie review snippets from that used in movie plot descriptions. Given this finding, it is to be expected that even the top state-of-the-art models reaching accuracies above 98% on the SUBJ dataset, would not succeed in distinguishing what is really subjective from what is really objective.

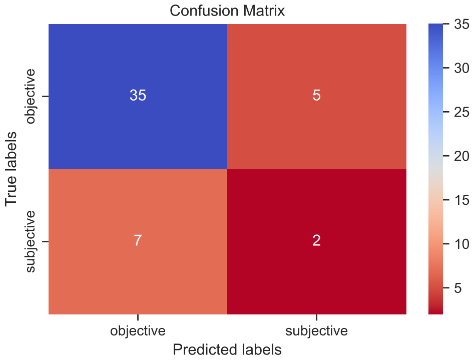


Figure 11: Confusion matrix of the state-of-the-art classifier trained on SUBJ dataset on our human-labeled test set.

5 Conclusion

In this paper, we approached the problem of subjectivity analysis as a regression task and tested the efficiency of transformer language representation models, such as RoBERTa, in annotating subjectivity using a paradigm of semi-supervised approach based on a small subset of human-labeled data. Our model showed a very high correlation with the average human rater and significantly outperformed a widely used rule-based *pattern* subjectivity regressor. The model also performed well as a binary classifier, both on our news dataset and on the benchmark subjectivity dataset exemplifying different discourse types. In contrast, we found that the state-of-the-art classifiers with best performance on the benchmark dataset and trained on that dataset show catastrophic performance on our human-labeled dataset, which is not very different from the baseline. This means that our model generalizes across domains much better than the current best systems. Moreover, this brings to light the issues of the subjectivity dataset that is considered the gold standard for subjectivity detection task. Rather than labelling for actual subjectivity, the gold standard dataset uses the origin of the texts (movie review snippets vs movie plot descriptions) as a proxy for subjectivity. Although the origin of the text undoubtedly correlates with subjectivity, these distinctions are not the same. As a result, state-of-the-art subjectivity classifiers trained on this dataset might be learning how to distinguish the language of movie review snippets from that of movie plot descriptions, rather than classifying subjectivity, as perceived by native speakers. Future work could further analyze how the performance of state-of-the-art classifiers trained on the benchmark subjectivity dataset compares to human-labeled subjectivity ratings in order to shed light on what exactly these systems are learning. Our work highlights the importance of using human annotations in such complex tasks as subjectivity detection. Future work can also be done in further comparing the performance of systems that are trained on the datasets labeled following explicit theoretical instructions to those trained on naïve human judgements about subjectivity. In addition, future studies on automatic subjectivity detection systems could investigate the origins of the differences in subjectivity perception across native speakers.

References

Francesco Antici, Luca Bolognini, Matteo Antonio Inajetovic, Bogdan Ivasiuk, Andrea Galassi, and Federico Ruggeri. 2021. SubjectivITA: An italian corpus for subjectivity detection in newspapers. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 12th International Conference of the CLEF Association, CLEF 2021, Virtual Event, September 21–24, 2021, Proceedings 12*, pages 40–52. Springer.

Steven Bird, Edward Loper, and Ewan Klein. 2009. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media Inc., Sebastopol, CA.

Iti Chaturvedi, Erik Cambria, Roy E. Welsch, and Francisco Herrera. 2018. Distinguishing between facts and opinions for sentiment analysis: Survey and challenges. *Information Fusion*, 44:65–77.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *ArXiv*, abs/2003.10555.

Tom De Smedt and Walter Daelemans. 2012. Pattern for python. *Journal of Machine Learning Research*, 13(66):2063–2067.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Christoph Hube and Besnik Fetahu. 2019. Neural based statement classification for biased language. In *WSDM '19: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19, pages 195–203, New York, NY, USA. Association for Computing Machinery.

Hairong Huo and Mizuho Iwaihara. 2020. Utilizing bert pretrained models with various fine-tune methods for subjectivity detection. In *4th International Joint Conference, APWeb-WAIM 2020, Tianjin, China, September 18-20, 2020, Proceedings, Part II*, pages 270–284. Springer.

Jakob Jünger and Till Keyling. 2019. Facepager. An application for automated data retrieval on the web.

Panagiotis Kasnesis, Lazaros Toumanidis, and Charalampos Z Patrikakis. 2021. Combating fake news with transformers: A comparative analysis of stance detection and subjectivity analysis. *Information*, 12(10):409.

Ronald W Langacker. 1990. *Subjectification*. Walter de Gruyter, Berlin/New York.

Chenghua Lin, Yulan He, and Richard Everson. 2011. Sentence subjectivity detection with weakly-supervised learning. In *Proceedings of 5th international joint conference on natural language processing*, pages 1153–1161.

Bing Liu. 2010. Sentiment analysis and subjectivity. In Nitin Indurkhya and Fred J. Damerau, editors, *Handbook of natural language processing*, 2 edition, pages 627–666. CRC Press.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv:1907.11692*.

Steven Loria. 2018. Textblob documentation.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 271–278, Barcelona, Spain.

Kartikey Pant, Tanvi Dadu, and Radhika Mamidi. 2020. Towards detection of subjective bias using contextualized word embeddings. In *Companion Proceedings of the Web Conference 2020*, pages 75–76, New York, NY, USA. Association for Computing Machinery.

Pavel Přibáň and Josef Steinberger. 2022. Czech dataset for cross-lingual subjectivity classification. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1381–1391, Marseille, France. European Language Resources Association.

Thilina Rajapakse. 2019. Simple transformers.

Leonard Richardson. 2007. Beautiful soup documentation.

Ellen Riloff and Janyce Wiebe. 2003. Learning extraction patterns for subjective expressions. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 105–112.

Ellen Riloff, Janyce Wiebe, and Theresa Wilson. 2003. Learning subjective nouns using extraction pattern bootstrapping. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, pages 25–32.

Santwana Sagnika, Bhabani Shankar Prasad Mishra, and Saroj K Meher. 2021. An attention-based cnn-lstm model for subjectivity detection in opinion-mining. *Neural Computing and Applications*, 33:17425–17438.

Elizabeth Closs Traugott. 1995. Subjectification in grammaticalization. In D. Stein and S. Wright, editors, *Subjectivity and subjectivisation: Linguistic perspectives*, volume 1, pages 31–54. Cambridge University Press.

Raphael Vallat. 2018. Pingouin: statistics in python. *Journal of Open Source Software*, 3(31):1026.

Sida I Wang and Christopher D Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 90–94.

Janyce Wiebe and Ellen Riloff. 2005. Creating subjective and objective sentence classifiers from unannotated texts. In *In Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Text Processing (CICLing-05)*, pages 486–497. Springer.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2-3):165–210.

Han Zhao, Zhengdong Lu, and Pascal Poupart. 2015. Self-adaptive hierarchical sentence model. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pages 4069–4076. AAAI Press.

A Appendix

Keywords for “Covid-19” news: "pandemic", "epidemic", "covid", "vaccin", "vaxx", "lockdown", "coronavirus", "omicron", "quarantine".

Keywords for “crime” news: "[ˆa-z]kill", "jail", "arrest", "crime", "murder", "kidnap", "[ˆa-z]rape", "[ˆa-z]rapi[ˆd]", "criminal", "terrorist", "shooting", "homicide", "robbery", "sentenced", "felony", "fraud".

Keywords for exclusion of news items about soap opera and natural disaster: “soaps”, “spoiler”, “storm”, “avalanche”, “volcano”, “lightning”, “tornado”, “flood”.

A Fine Line Between Irony and Sincerity: Identifying Bias in Transformer Models for Irony Detection

Aaron Maladry, Els Lefever, Cynthia Van Hee and **Veronique Hoste**
LT3 - Ghent University
`firstname.lastname@ugent.be`

Abstract

In this paper we investigate potential bias in fine-tuned transformer models for irony detection. Bias is defined in this research as spurious associations between word n-grams and class labels that can cause the system to rely too much on superficial cues and miss the essence of the irony. For this purpose, we looked for correlations between class labels and words that are prone to trigger irony, such as positive adjectives, intensifiers and topical nouns. Additionally, we investigate our irony model's predictions before and after manipulating the data set through irony trigger replacements. We further support these insights with state-of-the-art explainability techniques (Layer Integrated Gradients, Discretized Integrated Gradients and Layer-wise Relevance Propagation). Both approaches confirm the hypothesis that transformer models generally encode correlations between positive sentiments and ironic texts, with even higher correlations between vividly expressed sentiment and irony. Based on these insights, we implemented a number of modification strategies to enhance the robustness of our irony classifier.

1 Introduction

Irony is a complex form of figurative language with which people convey the opposite meaning of what they say. A typical example of verbal irony is the explicit expression of positive sentiment towards a negative situation or event. In some ironic statements, both the positive sentiment and the negative sentiment are expressed explicitly, like in the following example: "So nice of my stupid neighbor to start mowing the lawn in the morning". However, more subtle ironic statements make this paradox less obvious, if the speaker leaves out the explicit negative sentiment ("stupid"). When doing so, they assume the receiver of their message already knows the connotative sentiment linked to this situation. This assumption of connotative common-sense knowledge, along with the contradicting nature of the expression, makes automatic irony detection a notoriously hard task. Since detecting irony can also be difficult for humans, we often use rhetorical devices such as exaggerations, metaphors and intonation (in spoken language) or tone to hint at the underlying irony.

Whereas traditional feature-based approaches have long been the go-to methodology, most state-of-the-art systems have switched to bi-directional transformers (Devlin et al., 2018). These transformer systems have taken the lead for most benchmarks in Natural Language Processing (NLP), thanks to their word and sentence representations. This leap did, however, come at the cost of model explainability and insights into feature importance, which are more easily accessible for traditional machine learning algorithms, such as Logistic Regression and Decision Tree.

In this paper, we explore several explainability techniques and identify patterns in computational modeling of a fine-tuned transformer model for irony detection. In addition to existing metrics, we first search for potential biases in the data based on correlations between the irony label and lemmas in the train data. After investigating the impact of these bias words on the performance of our system (by replacing and removing them and checking the performance before and after), we combine this methodology with existing SOTA attribution techniques to verify whether they reveal similar patterns.

2 Related Research

As of late, transformer models have become an integral part of most state-of-the-art systems for irony detection. This can be done either through direct fine-tuning (Ángel González et al., 2020) or by using the contextual embeddings from a transformer model as input for a different neural classifier, like a Convolutional Neural Network (CNN) (Ahuja and

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 315–324
July 14, 2023 ©2023 Association for Computational Linguistics

Sharma, 2022). In some cases, this transformer representation input was enriched with additional features, as Cignarella et al. (2020) did by adding syntactic n-gram features. For Dutch specifically, transformers have already been fine-tuned and (quite favorably) compared to feature-based models with a focus on modeling the implicit sentiments in ironic statements (Maladry et al., 2022a).

The primary downside of these neural representation-based systems is that they are notoriously hard to interpret in a reliable way (Ghorbani et al., 2019). Nevertheless, plenty of attempts were made to explain such models. Popular techniques to gain insights into transformer models can be classified in two groups: (1) feature perturbation and (2) attention-based approaches. Most perturbation approaches, such as SHAP (Lundberg and Lee, 2017) and LIME (Ribeiro et al., 2016), are model-agnostic techniques that compare how changes in the input (feature representation) affect the output (prediction). While the aforementioned systems ignore the underlying model architecture, Integrated Gradients (IG) (Sundararajan et al., 2017) keeps track of the gradient changes caused by gradually lowering the feature values so as to find the features that arouse important gradient changes. As this makes it a neural-specific perturbation approach, it is positioned closer to the system architecture. One variation, called Discretized Integrated Gradients (DIG), grounds the perturbations in representations of sub-words that exist in the vocabulary of a transformer's tokenizer (Sanyal and Ren, 2021). By using real anchor-words with (mostly) lower feature values as interpolation points, this technique creates a non-linear path towards the padding-token (where all feature values are 0).

The model-specific alternative (2) to perturbation approaches instead relies on the attention weights and estimates the feature importance by tracking the activations in the neural network for a single input. Though popular, the use of attention weights as an explanation has been the topic of heavy discussion, with researchers opposing (Jain and Wallace, 2019) and supporting (Wiegreffe and Pinter, 2019) their reliability. The most popular explanation only accounts for the attention values in a single (usually the final) layer (Xu et al., 2015). More refined versions of this approach estimate the feature importances by combining the attention weights of different layers, either through averaging or with more sophisticated methods such as attention roll-out (Abnar and Zuidema, 2020) and layer-wise relevance propagation (LRP) (Montavon et al., 2017). While the use of attention is especially popular in computer vision, the more advanced methods have been successfully implemented and validated for transformer models for NLP by Chefer et al. (2021), who showed that their improved implementation of LRP gives better class-specific explanations compared to roll-out because the latter tends to attach too much importance to irrelevant tokens. Further mentions of LRP in this paper follow the implementation of Chefer et al. (2021).

To our knowledge, model explainability has not been researched thoroughly for irony detection. For Dutch, a manual analysis of wrong predictions already reveals some limitations of SOTA systems, including transformers (Maladry et al., 2022b). One of these limitations is a strong reliance on formulaic expressions and the hypothesis that (intense) positive sentiment is prone to function as an irony trigger. Still, this remains largely based on intuition and does not include an extensive investigation with existing attribution techniques for explainability.

3 Investigation of Potential Bias Words

3.1 Data description

For our experiments, we focus on a model that is fine-tuned on a data set for Dutch irony detection (Van Hee et al., 2016). This balanced data set contains 4453 train samples and 1113 held-out test samples. The ironic tweets in this corpus were collected using the Twitter API with irony hashtags, such as #ironie, #sarcasme and #not as search terms. These irony hashtags were then removed for training and testing. The non-ironic tweets were collected from the same users that wrote the ironic tweets and were then manually annotated to make sure that they are not ironic. All tweets in this corpus were labeled using a fine-grained annotation scheme, which also includes subcategories of irony, such as *irony by clash*, *situational irony* and *other verbal irony*. However, we will not elaborate on the distributions of these labels, as we only use the binary irony labels for this research.

bias word	EN	corr.	category
goed	good	0.106	adjective
school	school	0.0938	topical noun
weer	again	0.0906	intensifier
fijn	nice	0.0883	adjective

Table 1: A short list of the most correlated lemmatized unigrams in the data set with translation, Matthews' correlation to the binary irony label and the manually assigned category. The complete table with all n-grams and correlations is publicly available at `https://github.com/aMala3/DataBiasForIrony`.

3.2 Identifying Bias Words

Our first goal is to detect "bias words". In this research, "bias" is defined as spurious associations between word n-grams and class labels in our data set (Van Hee et al., 2016) that can cause the system to rely too much on superficial cues and miss the essence of the irony. For example, if many ironic tweets happen to contain a positive sentiment word, this could cause the system to use intense positive sentiment as a proxy for irony, as is suggested in Maladry et al. (2022b). To verify this hypothesis, we first split the sentences into lemmatized n-grams and use the presence of each n-gram as a binary feature. Subsequently, we calculated the correlation between the binary n-gram features and the irony label across the train set using Matthews' Correlation (also known as *phi coefficient*), a form of Pearson Correlation adapted for two binary values (being presence of the lemmatized n-gram on the one hand, and the irony label on the other hand).

We found that most correlated unigrams (potential spurious biases) can be classified into syntactic and semantic categories (as illustrated in Table 1). For tokens with a positive correlation, we identified the following categories: adjectives with a positive sentiment, intensifiers (including interjections and exclamations used to intensify the overall expression), and topical nouns. This final category includes nouns like "school", "exam", "train" or "bus" that have a strong semantic connection to a specific topic, such as *education* and *public transport*, respectively. On the other side of the spectrum, some of the tokens with a strong negative correlation include hyperlinks (to images, websites, etc.) and laughing, smiling or heart emojis and emoticons such as <3, :) and ;).

For larger n-grams, i.e. bigrams and trigrams, we found that they often include an important unigram and one or more common collocations, like "zo goed" (EN: so good), "wat fijn" (EN: how nice). In other cases, they can be a part of a complete formulaic expression like "goed begin van de dag" (EN: good start to the day). Since this means that there is a significant overlap between the different n-grams, we focus on the unigrams for this study.

system	lab.	prec.	rec.	F1	N
Full	0	0.74	0.74	0.74	2245
	1	0.74	0.74	0.74	2208
Adj.	0	0.69	0.55	0.61	164
	1	0.84	0.91	0.87	431
Int.	0	0.66	0.60	0.63	472
	1	0.77	0.82	0.79	778

Table 2: System performance of the fine-tuned RobBERT model in 10-fold cross-validation on the train set. We compare the performance on the complete train set (Full) to the performance on the subsets where all tweets contain one of the identified adjectives (Adj.) or intensifiers (Int.).

3.3 Quantitative Analysis on Subsets

To investigate whether the correlation between those potential bias words and the irony label is not groundlessly used as an approximation of irony, we evaluate the performance of a fine-tuned transformer on tweets that specifically contain these words.[1] We focus on the first two categories (adjectives and intensifiers) as they can be more easily isolated. For evaluation, we fine-tuned the pretrained Dutch RobBERT (Delobelle et al., 2020) model in a 10-fold cross-validation setting. All systems for this paper were fine-tuned for 2 epochs (200 warm-up steps) with an AdamW optimizer, learning rate of 5e-5, weight decay of 0.01, evaluating every 200 steps on a batch size of 8 on a NVIDIA Tesla V100 GPU with 10% of the train data (for each fold) held out as validation set.

As shown in Table 2, the system performs better for the subsets containing adjectives (*Adj.*) and intensifiers (*Int.*) than on the full set for the irony label '1', with recall scores of 0.91 and 0.82. The lowest scores in the table are the recall scores of 0.55 and 0.60 on the non-ironic label. The combination of these two findings indicates that the system overgenerates irony predictions and has some room for improvement on non-ironic texts in the subsets for adjectives and intensifiers.

---

[1] We limited this list to the 50 most correlated words and manually verified them.

In a next step, we wanted to investigate to what extent the potential bias words serve as trigger words and are therefore responsible for overgenerating irony predictions. We therefore systematically adapt the input by changing the potential bias words and examine how this affects the performance on these subsets.

4 Modified Samples

4.1 Modification

The first category of potential bias words we investigate contains adjectives that are generally used to explicitly express positive sentiment about a situation with a negative connotation. As this is essential to the irony, the adjectives cannot be simply removed. Instead, we propose replacing sentiment words with a strong correlation to the irony label with a synonym that has a weaker or no correlation. To find synonyms for the adjectives, we generated the 20 nearest neighbors using fastText embeddings (Bojanowski et al., 2016) from a pretrained Dutch model with an embedding size of 300 (Grave et al., 2018). The downside of using such a similarity-based approach is that that the proposed "synonyms" can have the opposite sentiment compared to the original adjective. For the word "gelukkig" (EN: lucky or luckily), for example, one of the suggested synonyms was "helaas" (EN: unfortunately) and for "goed" (EN: good), the most similar word was "slecht" (EN: bad). To ascertain the validity of our irony label, as well as the semantic and structural integrity of our sentences, we manually verified and selected a list of relevant synonyms for each adjective. In this case, we employed two setups: one always opting for the least correlated word in the (automatically generated but manually verified) synonym set and one randomly selecting one of the possible synonyms (that are not ranked in the top 50 most correlated lemmas but can still have some correlation to the label).

The second category contains intensifiers (including interjections and exclamations), which function as a supporting element for the expression of irony. As these are the building blocks for hyperboles and exaggerations, they are very common and help clarify that a message should be interpreted as ironic. However, they are generally not essential to recognize the irony because they only intensify the contrasting sentiment that is already expressed by the other words in the sentence. Therefore, we assume they can be removed without significantly altering the meaning of the sentence or flipping the irony label.

set	lab.	prec.	rec.	F1	prob.
adj.	0	0.69	0.55	0.61	0.252
OG	1	0.84	0.91	0.87	0.748
adj.	0	0.64	0.55	0.59	0.262
corr.	1	0.84	0.88	0.86	0.738
adj.	0	0.66	0.55	0.60	0.257
rand.	1	0.84	0.89	0.86	0.743
int.	0	0.66	0.60	0.63	0.357
OG	1	0.77	0.82	0.79	0.643
int.	0	0.61	0.67	0.64	0.435
rem.	1	0.79	0.74	0.76	0.565

Table 3: System performance of the fine-tuned RobBERT model in 10-fold cross-validation on the train set. We compare the results for the subsets where all tweets contain one of the identified adjectives (adj.) or intensifiers (int.) in original form (OG) to the same subsets in modified form (corr., rand. or rem.). The last column presents the average probability of both labels in the subset.

4.2 Quantitative Analysis on Modified Subsets

As presented in Table 3, replacing the adjectives by a synonymous word only causes minimal changes in system performance across both labels. On average, replacing one of the highly correlated adjectives with the least correlated synonymous (positive) adjective (Adj. corr.) causes the irony probability to drop by 1%. When using a random synonym from the list, the average probability only drops by 0.5% (adj. rand.). This tells us that the system is well able to overcome the lexical feature level and properly generalizes the positive meaning of the adjectives.

Although replacing the adjectives only has a minimal impact, the experiments for intensifiers show different results. As is shown in Table 3, removing intensifiers (Int. rem.) from non-ironic tweets improves the recall on those tweets by 7%. By contrast, the same modification caused the recall for the irony label to diminish with 8%. Along with the average predicted irony probability dropping by 8%, this further supports the hypothesis that the system relies on positive sentiments and considers sentiment intensity a trigger for irony.

As we mentioned before, exaggerations and hyperboles may indicate irony and are therefore also an intuitive cue for humans trying to identify irony.

However, prototypical examples of explicit irony are ironic due to the contrast between the expressed and expected sentiment towards a situation, and do not solely depend on the expression of a positive sentiment or its intensity. If an automatic system is too dependent on this element that supports the expression of irony, it is likely to miss more subtle cases and mistake genuinely (intensive) positive sentiment for irony.

5 Explainability Metrics

To further solidify our understanding of how our system models irony, we also employ explainability metrics that account for the system architecture and the mechanisms that drive its decision-making.

5.1 Explaining the Metrics

In our analysis, we include three metrics to locate trigger words: Discretized Integrated Gradients (DIG)[2], Layer-Integrated Gradients (LIG)[3] and an improved implementation of Layer-wise Relevance Propagation (LRP)[4], each with increasing reliance on the model architecture.[5]

DIG and LIG are two feature perturbation approaches based on Integrated Gradients. This means that they estimate feature importance by using alternative input representations (perturbations) with gradually lowered feature scores and comparing those feature changes to the resulting gradient changes. Unlike the original implementation of Integrated Gradients, which uses linear paths to scale down the feature representations, DIG creates a non-linear interpolation path by sending it through the representations of *real* anchor words. These anchor words are sub-words that exist in the vocabulary of the transformer model's tokenizer. LIG, on the contrary, does not account for real word representations, but instead considers how the activations in the final layers are influenced by the activations in previous layers. Compared to those approaches, the LRP attribution technique relies completely on the activations, weights and biases triggered by the current feature representation and does not consider alternative input representations. This implementation (Chefer et al., 2021) uses Deep Taylor Decomposition to attribute importances in each layer, which are then propagated backward throughout the network to result in total attributions for (only) the predicted label.

5.2 Setup of the Analysis

All applied explainability metrics assess how each sub-word in a text impacts the final prediction. Therefore, the resulting attributions are only valid on a local level (i.e. single text samples) and are not general model features. To overcome this issue, we perform both a manual analysis on the local tweet level and a search for generally relevant tokens.

For our manual evaluation, we investigated a random sample of texts with a focus on the sub-sets discussed in Sections 3 and 4. This manual analysis consisted of two parts: first, we compared the different metrics among each other (on the same sample of 50 tweets for all 3 metrics) and second, we looked for systematic attribution patterns (on an additional sample of 100 tweets). To estimate general feature importances in a more quantitative way, we calculated the average attributions for each subtoken in the complete train corpus[6]. This allows us to verify which sub-words our system generally considers more important and complements our manual analysis.

In Section 5.3, we discuss the first part of our manual analysis (i.e. comparing the different metrics). Subsequently, in Section 5.4, we combine the insights from the second part of the manual analysis with an inspection of average token attributions to discuss the general attributions patterns.

5.3 Comparing Explainability Metrics

To gain intuitive insights in the respective quality of the different explainability metrics, we performed a manual analysis using the three metrics on a set of 50 samples, half of which containing a positive sentiment word (1) and the other half containing an intensifier (2). In the following examples (Figures 1 and 2), we present the tokens with a positive attribution (for the irony label) in green and the negative attributions (for the non-ironic label) in red, with brighter colors presenting stronger attributions. For DIG and LIG, a sentence can contain both positive and negative attributions, but this is not the case for LRP as the attributions are label-specific. For each of the metrics, the visualizations were generated with Captum (https://captum.ai/), a

---

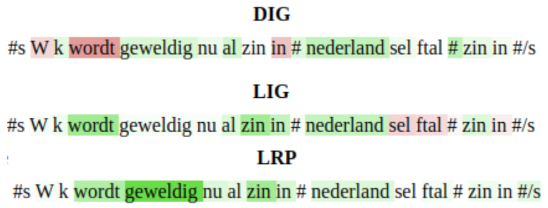
[2] https://github.com/INK-USC/DIG

[3] https://github.com/cdpierse/transformers-interpret

[4] https://github.com/hila-chefer/Transformer-Explainability

[5] For each text, we use the explanation for unseen data, meaning that the attributions were generated with the same 10-fold cross-validation setup we used in Section 3.3.

[6] `https://github.com/aMala3/DataBiasForIrony`



DIG

#s W k wordt geweldig nu al zin in # nederland sel ftal # zin in #/s

LIG

#s W k wordt geweldig nu al zin in # nederland sel ftal # zin in #/s

LRP

#s W k wordt geweldig nu al zin in # nederland sel ftal # zin in #/s

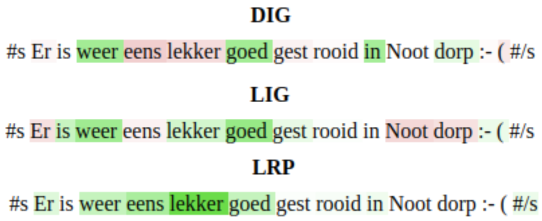
Figure 1: World Cup is going to be great, already looking forward to it #DutchTeam #lookingforwardtoit

specialized library for visualizing explanations for neural networks.

After manual inspection, we found that DIG provides the least intuitive importance attributions, whereas LIG and LRP seem to work better for irony detection. In examples 1 and 2, “wordt” (EN: will be) and “in” (EN: in) gain strong attributions, even though these words only serve a functional purpose. This issue, where high importance is assigned to irrelevant tokens such as non-creative punctuation (e.g., a comma or full stop) and function words (e.g. articles), seems to be the most common for DIG, less common for LIG and the least common for LRP. For DIG, we found that non-zero attributions were assigned to irrelevant punctuation in 18 out of 50 samples (36%) and to irrelevant function words in 26 samples (52%). Meanwhile, for LIG we observed irrelevant punctuation in 7 cases (14%) and irrelevant function words in 17 cases (34%). For LRP, irrelevant punctuation was only overestimated in 3 cases (6%) and function words in 7 cases (14%). Whereas the DIG (and to a lesser extent LIG) attributions for “weer eens lekker goed” (EN: really did a great job again) vary between positive and negative, LRP attributions recognize them as a single span and the primary reason for the irony prediction in this tweet. On the same 50 samples, the DIG attributions gave such opposite attributions within a single word (split into sub-words) or text span in 30 samples (60%), as opposed to the LIG attributions, which only showed this in 18 samples (36%). This issue cannot occur for LRP, because it has label-specific attributions. To conclude, since we showed that LRP is less likely (1) to attribute importance to irrelevant tokens, and (2) to attribute contradicting importances to sub-tokens that belong together, LRP revealed to be the most meaningful metric for our analysis.

5.4 Discussing the Attribution Patterns

Combining our extensive manual analysis on a larger sample (100 additional tweets) with an in-



DIG

#s Er is weer eens lekker goed gest rooid in Noot dorp :- (#/s

LIG

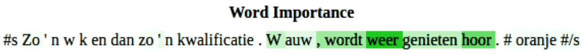
#s Er is weer eens lekker goed gest rooid in Noot dorp :- (#/s

LRP

#s Er is weer eens lekker goed gest rooid in Noot dorp :- (#/s

Figure 2: They really did a great job again sprinkling road salt in Nootdorp :-(

spection of the averaged token attributions allows us to confidently present the following insights. First, we found that the intensifiers, interjections and exclamations indeed receive high attributions, especially when combined with positive sentiment words. Those positive sentiment words, like “goed” (EN: good) and “fijn” (EN: nice) also receive relatively high attributions by themselves. A manual check of the sub-tokens with an average attribution of over 0.75 revealed that 111 of the 254 subtokens (44%) have a positive sentiment. Replacing these sentiment words by synonyms (as was done in Section 4) with a more intense sentiment, such as “geweldig” (EN: great) and “fantastisch” (EN: fantastic), barely increases the attributions. Still, adding a lexical intensifier to a positive adjective results in a larger increase. The highest attributions are often linked to either the intensifier or the positive adjective, without any clear reason to choose one over the other. When several intensifiers and sentiment words co-occur in the same text, the attribution methods stack the attributions on one or a few words, while disregarding the others. As shown in Figure 3, LRP correctly spreads the attributions evenly across a typical formulaic expression.



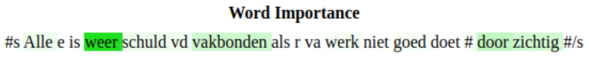
Word Importance

#s Zo ' n w k en dan zo ' n kwalificatie . W auw , wordt weer genieten hoor . # oranje #/s

Figure 3: This kind of World Cup and then this kind of qualification. Wow, will be fun again. #orange

Some topics with high correlations in Section 3 also achieve high LRP attributions on average. This is shown in the topic *politics*, represented by subwords like “conservatieve” (EN: conservative; with an average attribution of 0.95).

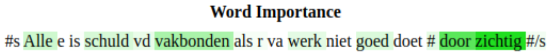
However, this is not the case for *“school”*, related to the topic *education*), and *trein* (EN: train), related to the topic*public transport*). Based on the lemmatized unigram correlations, “school” has the second highest correlation with the irony label, but



Word Importance

#s Alle e is weer schuld vd vakbonden als r va werk niet goed doet # door zichtig #/s

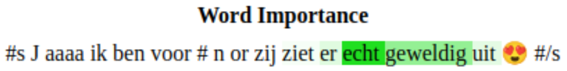
Figure 4: The *unions* are to blame for everything *again* if rva does not do its job properly *#transparent*



Word Importance

#s Alle e is schuld vd vakbonden als r va werk niet goed doet # door zichtig #/s

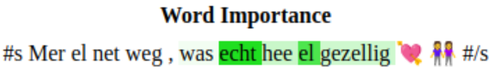
Figure 5: The *unions* are *to blame* for *everything* again if rva does not do its *job properly #transparent*



Word Importance

#s J aaaa ik ben voor # n or zij ziet er echt geweldig uit 😍 #/s

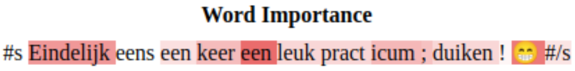
Figure 6: Yaaa, I'm here for #nor she looks *really great*



Word Importance

#s Mer el net weg , was echt hee el gezellig 💘 👭 #/s

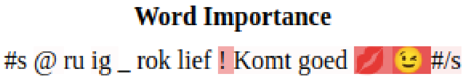
Figure 7: Merel just left, was *really veery nice*



Word Importance

#s Eindelijk eens een keer een leuk pract icum ; duiken ! 😁 #/s

Figure 8: Finally a fun particum; diving! :D



Word Importance

#s @ ru ig _ rok lief ! Komt goed 💋 😉 #/s

Figure 9: #ruig_rok sweet! will be all right

the sub-token "school" only has an average LRP attribution of 0.07. Likely, this is due to the fact that the attributions tend to relate to intensifiers and positive adjectives. We assume this because the manual evaluation revealed that removing intensifiers can occasionally redirect the attributions to topics, as shown in Figure 4 (before removal) and Figure 5 (after removal). Here,"weer" (EN: again) is the intensifier, and "vakbonden" (EN: labor unions) are the topic of the evaluation.

Surprisingly, the system also attributes high scores to nouns that do not fit the expected topical nouns, but that instead have a strong negative connotative sentiment, such as "wereldoorlog" (EN: world war) and "dictatuur" (EN: dictatorship). Alongside the aforementioned topical nouns, this could indicate that the system already models some connection between positive (adjectives) and negative (topics) sentiments within the same sentence. This is also visible on a larger scale, when comparing the average attributions on the positive adjective subset to the same subset where the adjectives are replaced by less correlated synonyms. The average attribution of "tandarts" (EN: dentist) changes from 0.69 to 1 when replacing a highly correlated adjective with a less correlated synonym, as was done in Section 4. Still, this currently seems to be limited to very popular topics for irony and nouns with a strong negative sentiment.

When a text contains intense positive sentiment, but is not intended in an ironic way, it is at risk to be mistaken for irony. As shown in Figures 6 and 7, both the intensifiers, "echt" (EN: really) and "heel" (EN: very), as well as the positive adjectives "geweldig" (EN: amazing) and "gezellig" (EN: pleasant / cozy), achieve high attributions for the irony label. Notably, the positive emojis do not receive any attributions for the irony label, while they carry the same sentiment. In fact, when looking at the correctly classified genuinely positive texts (Figures 8 and 9), this type of positive emoji serves as a trigger for the non-ironic label. Based on the averaged token attributions[7], this generally seems to be the case for a selection of positive emojis. Moreover, the correlation between single tokens and the irony label in Section 3 already identified this as a potentially spurious bias.

Altogether, these results suggest that, while the performance of transformers for irony detection is quite good, the way our system models irony remains rather superficial since it seems to depend on the detection of lexical exaggerations of positive sentiment. Although there are some indications to argue that the model can partially model the contrast that is so essential to irony, the system seems to only pick up the most extreme contrasts and most common topical nouns. For the non-ironic label, the system has also modeled an exaggerated relation between the use of positive emojis and a text being sincere because they were simply more common in those texts. In the end, the patterns in our transformer model are more similar to the simple correlations calculated in Section 3 than expected.

6 Training with Modified Data

As argued in Section 4, the modified tweets can keep the same irony label as the original tweets. This means that they could also prove helpful as additional train data, as they introduce (1) more lexical variety to the data set by using less frequent synonyms and (2) are also examples of irony with a

---

[7] Available at `https://github.com/aMala3/DataBiasForIrony`.

system	prec.	rec.	F1	acc.
Base	0.7242	0.7210	0.7210	0.7227
Aug	**0.7245**	**0.7221**	**0.7222**	**0.7237**

Table 4: System performance of the fine-tuned Roberta models on the held-out test set. Scores are averaged over 5 train runs to overcome the relatively difference between highest and lowest F1-score (2-3%) depending on the seed.

lower sentiment intensity. Therefore, the modified samples could help improve the robustness of our model and allow it to recognize more subtle ironic expressions.

In Table 4, we present the scores on the held-out test set for the two versions of the train set: (1) the original train set (*Base*) with a train size of 4007 and validation size of 446 and (2) the original train data including the modified samples (*Aug*) with a train size of 5794 and validation size of 645 (a total increase of 1986 samples).

While the augmented system may be more robust in practice, fine-tuning RobBERT on (2) the data set with modified examples shows no direct improvement over the original train set when evaluating on the held-out test set. We hypothesize this may be related to the fact that the same biases that are part of the train data are also present in the test data, as it is part of the same data set.

7 Conclusion

In this paper we investigated the origin and effect of bias on automatic irony detection in Dutch. By looking into the potential biases that could emerge from the data, we found that these can be classified into three categories: (1) positive sentiment words (mostly adjectives), (2) intensifiers, interjections and exclamations and (3) topical nouns. To investigate whether our fine-tuned transformer model uses these biases as trigger words, we evaluated the system performance on subsets that specifically contain those bias words (i.e. positive sentiment words and intensifiers) and compared the results to modified samples where the adjectives were replaced by synonyms and the intensifiers were removed. In addition, we also investigated how our system models irony using three state-of-the-art explainability techniques that assign feature attributions to each of the sub-tokens in a text: Discretized Integrated Gradients (DIG), Layer Integrated Gradients (LIG) and Layer-wise Relevance Propagation(LRP). After generating LRP attributions for every text in our train set, we ranked the different sub-tokens according to their average impact on the prediction.

Both of these methods support the hypothesis that our fine-tuned system for irony detection strongly relies on positive sentiment and is particularly triggered by intense positive sentiment. Although some of the common topics of ironic tweets are partially recognized as important and become slightly more important when the intensifiers are removed, the system generally does not pay too much attention to them.

While intense positive sentiments are most commonly used in hyperboles or exaggerations, which is a rhetorical device used to support the expression of irony, they are not the sole solution for irony detection. As shown in our analysis, more subtle irony can go undetected and genuinely positive texts are often wrongly classified as ironic. Therefore, we attempted to use modified samples with less intense sentiments to augment our train data. Although there was no noticeable increase in performance in the held-out test set, further testing on new external data sets is needed to make reliable conclusions.

For future research, we suggest further investigating how these biases could be mitigated to make sure genuine sentiment is not mistaken for irony. This could either be done by further augmenting the data or by adapting the model. A data-driven approach we propose is to create counterfactual samples, where an ironic tweet is made non-ironic and the other way around. This is, however, no simple feat due to the creativity in ironic expressions. Likewise, our approach for creating modified samples can still be improved to result in a fully-automatic framework for data augmentation. As model adaptation, we propose improving the system with additional features that, for example, represent relevant common-sense knowledge. Finally, it would also be interesting to use human annotations of trigger words for irony detection and compare the perspectives of annotators to the model explanation.

Acknowledgements

This work was supported by Ghent University under grant BOF.24Y.2021.0019.01 as well as the interdisciplinary NewsDNA project under grant BOF.GOA.2018.0006.03.

References

Samira Abnar and Willem Zuidema. 2020. Quantifying attention flow in transformers. *arXiv preprint arXiv:2005.00928*.

Ravinder Ahuja and Subhash Chander Sharma. 2022. Transformer-based word embedding with cnn model to detect sarcasm and irony. *Arabian Journal for Science and Engineering*, 47(8):9379–9392.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Hila Chefer, Shir Gur, and Lior Wolf. 2021. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 782–791.

Alessandra Teresa Cignarella, Valerio Basile, Manuela Sanguinetti, Cristina Bosco, Paolo Rosso, and Farah Benamara. 2020. Multilingual irony detection with dependency syntax and neural models. *arXiv preprint arXiv:2011.05706*.

Pieter Delobelle, Thomas Winters, and Bettina Berendt. 2020. RobBERT: a Dutch RoBERTa-based Language Model. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3255–3265, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Amirata Ghorbani, Abubakar Abid, and James Zou. 2019. Interpretation of neural networks is fragile. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3681–3688.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Sarthak Jain and Byron C Wallace. 2019. Attention is not explanation. *arXiv preprint arXiv:1902.10186*.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc.

Aaron Maladry, Els Lefever, Cynthia Van Hee, and Veronique Hoste. 2022a. Irony detection for dutch: a venture into the implicit. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 172–181.

Aaron Maladry, Els Lefever, Cynthia Van Hee, and Véronique Hoste. 2022b. The limitations of irony detection in dutch social media. *Language Resources and Evaluation*.

Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. 2017. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern recognition*, 65:211–222.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144.

Soumya Sanyal and Xiang Ren. 2021. Discretized integrated gradients for explaining language models. *arXiv preprint arXiv:2108.13654*.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *International conference on machine learning*, pages 3319–3328. PMLR.

Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2016. Exploring the realization of irony in twitter data. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 1795—-1799, Paris, France. European Language Resources Association (ELRA).

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. *arXiv preprint arXiv:1908.04626*.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR.

José Ángel González, Lluís-F. Hurtado, and Ferran Pla. 2020. Transformer based contextualization of pre-trained word embeddings for irony detection in twitter. *Information Processing & Management*, 57(4):102262.

8 Limitations

The primary limitation for this work is the difficulty of telling the difference between mistakes made by automatic systems and wrongly assigned importances from the attribution techniques. Additionally, our work currently only relies on a single pre-trained model that was fine-tuned on the currently only available data set for Dutch irony detection. The patterns in computational modeling we described only apply to this particular system and data set and may very well differ when training a different model on data that was collected in a different way, where the data set may rely on different patterns and biases. Finally, despite the good agreement scores (a Cohen's kappa of 0.84) for binary irony classification, this remains a complex task where annotators can be uncertain about the label. In the end, the annotators for any irony or sarcasm detection task can only make assumptions about what the author of a text intended to convey. For our setup, both the annotators and the automated system predict whether a text is ironic without considering the corresponding context. In a realistic setting, most social media texts are reactions to previous comments or external events that can be essential in order to recognize the irony. This means that the model predictions can differ from the annotated label but still be a plausible interpretation.

ChatGPT is fun, but it is not funny!
Humor is still challenging Large Language Models

Sophie Jentzsch[1] and **Kristian Kersting**[2,3,4]
[1]Institute for Software Technology, German Aerospace Center (DLR), Cologne, Germany
[2]Computer Science Department, Technical University Darmstadt, Darmstadt, Germany
[3]Centre for Cognitive Science, TU Darmstadt
[4]Hessian Center for AI (hessian.AI)
sophie.jentzsch@dlr.de, kersting@cs.tu-darmstadt.de

Abstract

Humor is a central aspect of human communication that has not been solved for artificial agents so far. Large language models (LLMs) are increasingly able to capture implicit and contextual information. Especially, OpenAI's ChatGPT recently gained immense public attention. The GPT3-based model almost seems to communicate on a human level and can even tell jokes. Humor is an essential component of human communication. But is ChatGPT really funny?

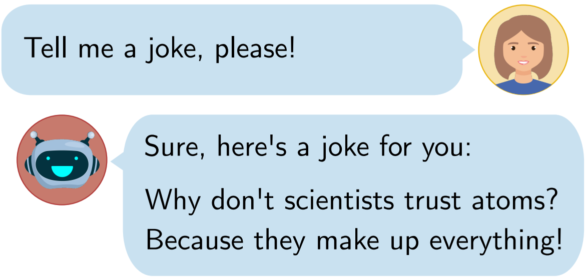
We put ChatGPT's sense of humor to the test. In a series of exploratory experiments around jokes, i.e., generation, explanation, and detection, we seek to understand ChatGPT's capability to grasp and reproduce human humor. Since the model itself is not accessible, we applied prompt-based experiments.

Our empirical evidence indicates that jokes are not hard-coded but mostly also not newly generated by the model. Over 90% of 1008 generated jokes were the same 25 Jokes. The system accurately explains valid jokes but also comes up with fictional explanations for invalid jokes. Joke-typical characteristics can mislead ChatGPT in the classification of jokes. ChatGPT has not solved computational humor yet but it can be a big leap toward "funny" machines.

1 Introduction

For humans, humor plays a central role in forming relationships and can enhance performance and motivation [16]. It is a powerful instrument to affect emotion and guide attention [14]. Thus, a computational sense of humor holds the potential to massively boost human-computer interaction (HCI). Unfortunately, although computational humor is a longstanding research domain [26], the developed machines are far from "funny." This problem is even considered to be AI-complete [22].

Continuous advances and recent breakthroughs in machine learning (ML), however, lead to an ever-increasing panoply of applications [e.g., 15, 3] and likewise open new opportunities for natural language processing (NLP). Transformer-based large language models (LLMs) increasingly capture and reflect implicit information, such as stereotypes [7], moral [6], and humor [5, 25]. Humor is often implicit and carried by subtle details. Thus these novel qualities of LLMs give reason to hope for new advances in artificial humor.



Tell me a joke, please!

Sure, here's a joke for you:

Why don't scientists trust atoms?
Because they make up everything!

Figure 1: Exemplary illustration of a conversation between a human user and an artificial chatbot. The joke is a true response to the presented prompt by ChatGPT.

Most recently, OpenAI's ChatGPT gained immense attention for its unprecedented capabilities. Users can interact with the model via public chat API in a conversation-like course. The system is able to answer a huge variety of questions while taking the previous contextual conversation into consideration. In fact, it can even tell jokes, as displayed in Fig. 1. ChatGPT is fun and almost appears to interact on a human-like level. Yet, when interacting with the model, users may quickly get a glimpse of its limitations. Although ChatGPT generates text in almost error-free English, grammatical glitches and content-related mistakes occur. In the preceding exploration, we noticed that ChatGPT is likely to repeat the exact same jokes frequently. Moreover, the provided jokes were strikingly correct and sophisticated. These observations led to the hypothesis that output jokes are not originally generated by the model. Instead, they seem

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 325–340
July 14, 2023 ©2023 Association for Computational Linguistics

to be reproduced from the training data or maybe even hard coded in a predefined list.

As the system's inner workings are not shared, we conducted a series of structured prompt-based experiments to learn about the system's behavior and allow for inference regarding the generation process of ChatGPT's output. Specifically, we aim to understand to what extent ChatGPT is able to capture human humor by conducting a systematic prompt-based analysis.

The remainder of this paper is structured as follows: First, Sec. 2 discusses related research. The main contribution assembles three experimental conditions: Joke generation, joke explanation, and joke detection. While the overall approach is outlined in Sec. 3, the detailed conduction is explained together with the results of each experiment in Sec. 4.1, Sec. 4.2, and Sec. 4.3, respectively. We close by discussing implications and further research in Sec. 5 and conclude our work in Sec. 6.

Terminology Disclaimer. AI-related terminology tends to make use of analogies to human characteristics, such as *neural networks*, or the term artificial *intelligence* itself. Likewise, we use human-related terms in the discussion around conversational agents, which are supposed to mimic human behavior as accurately as possible, e.g., ChatGPT "*understands*" or "*explains*." Although we believe that these analogies describe the system's behavior and its inner workings very well, they might be misleading. We would like to emphasize that the discussed AI models are not on a human level and, at best, constitute a simulation of the human mind. Whether AI can potentially ever *think* or *understand* in a conscious way is a philosophical question that is out of the scope of this investigation.

2 Related Work

Our work considers the intersection of two lines of research, namely LLMs and computational humor.

Large Language Models. NLP witnessed several leaps in the recent past. Only a few years ago, the introduction of transformer architectures in neural networks [21] enabled the development of contextualized models, such as BERT [9] or GPT [2]. These so-called large language models (LLMs) capture relations in the natural data and even reflect implicitly-included information, which can be both a risk [23] and a chance [17]. Either way, however, it is a prerequisite to solving computational humor.

OpenAI[1] recently released ChatGPT, a GPT3.5-based LLM that is able to interact with users conversationally through a chat-like API. It is fine-tuned with *Reinforcement Learning from Human Feedback (RLHF)* [12] in three essential steps, including large-scale manual labeling and proximal policy optimization [18]. The result of this process is a model with unprecedented capabilities: It interacts in a conversational manner, i.e., it refers to the users' prompt while also taking information from the preceding conversation into account. It is able to summarize information, generate new texts of all shapes, and even write program code. Nevertheless, there are still glitches and limitations, e.g., factual wrong information presented as valid.

Computational Humor. Computational humor is a small but active research area of NLP with the main streams *Humor Detection* or *Classification* [e.g., 8, 4], and *Humor Generation* [e.g., 19]. Although advances in ML and NLP transfer to computational humor, researchers highlight the great complexity of automated humor and the limitations of current endeavors [26, 24]. Humor is one of the few capabilities that seemed to be reserved for human individuals thus far, which is why it is even considered AI-complete [14, 22].

While decades of research in linguistics and psychology offer quite a few logic-based humor theories [e.g., 13, 27], the work in the context of artificial agents is predominantly data-driven. In these approaches, pretrained language models, such as ChatGPT, play a central role [10]. With their innovative capabilities, GPT-based models have the potential to open a new chapter of computational research.

3 Method

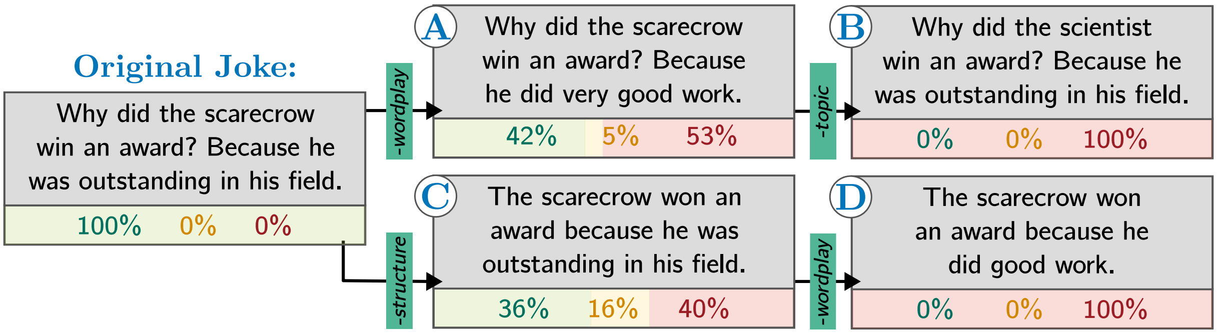
The presented experiments are grouped into three individual tasks, which are introduced in Sec. 3.1, Sec. 3.2, and Sec. 3.3. Implementation details and extended results are made available at GitHub[2]. In all experiments, each prompt was conducted in a new empty conversation to avoid unwanted influence. To conduct a large number of prompts with OpenAI's ChatGPT[3] web service, there were certain obstacles to overcome. Since there was no official API available at the time, prompts were

---

[1] OpenAI, `https://openai.com/`

[2] Project repository: `https://github.com/DLR-SC/JokeGPT-WASSA23`

[3] ChatGPT user API at `chat.openai.com/`, Accessed: January-March 2023 (detailed dates in the Appendix)



Why did the scarecrow win an award? Because he was outstanding in his field.

A: Why did the scarecrow win an award? Because he did very good work.

B: Why did the scientist win an award? Because he was outstanding in his field.

C: The scarecrow won an award because he was outstanding in his field.

D: The scarecrow won an award because he did good work.

Figure 2: Modification of top jokes to create joke detection conditions. Below each condition, the percentages of samples are stated that were classified as joke (green), potentially funny (yellow), and not as a joke (red). In condition *(A) Minus Wordplay*, the comic element, and, therefore, the pun itself, was removed. For condition *(B) Minus Topic*, the joke-specific topic was additionally eliminated, e.g., by removing personifications. Condition *(C) Minus Structure* keeps the validity of the joke intact but changes the typical q-a-structure to a single-sentence-sample. From that, the comic element was additionally removed to create condition *(D) Minus Wordplay.*

entered with the help of a wrapper. The number of permitted prompts per hour was limited to 74. Moreover, ChatGPT was unavailable for longer periods due to exceeded capacity.

In this work, we differentiate between *originally generated* output, i.e., text composed by the model, and *replicated* output, i.e., text that is memorized from training data and played back by the system in the exact wording. *Modified* output is a mix of both, i.e., replicated text that is slightly altered, e.g., mixing the first half of one joke with the second half of another. We classify a joke as *valid* if it is funny from a human perspective. Accordingly, an *invalid* joke might be grammatically correct and even contain joke-like elements but fails to deliver a punch line. Naturally, as humor is subjective, these categories are always debatable. That being said, the distinction is comparably evident for the present examples, as we expound in the following chapters.

3.1 Joke Generation

To test if there is a limited number of reoccurring jokes, we analyze the deviation of output jokes. To this end, we asked ChatGPT to provide a joke a thousand times. We applied a predefined list of ten differently worded prompts, such as *"Can you tell me a joke, please?"* or *"I would love to hear a joke."* The resulting observations are described in Sec. 4.1. We identified 25 repeating top jokes, which form the basis for the two subsequent tasks.

3.2 Joke Explanation

In the joke generation task, it was tested whether ChatGPT is able to generate valid jokes. However, this task does not necessarily reflect the system's ability to *understand* humor, i.e., *why* the joke might be funny from a human perspective. To see to what extent the model captures these complex inner workings of jokes, we asked ChatGPT to explain each of the generated top 25 jokes. The prompt was always *"Can you explain why this joke is funny:* [joke]." The results from this second task are presented in Sec. 4.2.

3.3 Joke Detection

In the first two tasks, we identified certain criteria that (almost) all output joke samples had in common, i.e., structure, topic, and wordplay. These criteria seemed to be central joke characteristics for ChatGPT. To examine how close these cues are connected to ChatGPT's conception of humor, we manually modified the top 25 jokes to eliminate one or multiple of the three criteria, resulting in four additional conditions for these jokes. We asked the system to classify each sample with the prompt "*What kind of sentence is that:* [sample]." ChatGPT's response would then either include a categorization as a joke or not, as explained in Sec. 4.3. The three defined joke characteristics were defined as follows:

Structure: The generated jokes from Sec. 4.1 were in noteworthy similar semantic structure. Despite one sample, all 1008 generated jokes were in the same question-answer format.

Comic element: In jokes, there is usually a stylistic element that creates comic momentum. ChatGPT's generated jokes exclusively contained wordplay puns, e.g., the double meaning of one word.

Topic Joke scenarios tend to be bizarre and not close to reality. Not always, but often they contain personifications of objects, i.e., protagonists can be computers or bananas.

To determine the impact of these characteristics on the classification result, we compared the original top 25 jokes to samples with one or multiple of these characteristics removed. The considered jokes were modified manually as described in Fig. 2 to create alternative samples that were semantically as similar as possible. The comprehensive sets of samples and their classification can be found in the Appendix in Sec D. The sets were created as follows.

In the first modification *A*, the wordplay was removed from the joke (*minus wordplay*). To achieve that, the term(s) that form(s) the center of the pun were replaced by a wording with comparable primary meaning but without double meaning. As a side effect, this step removes the comic element and therefore destroys the pun. The joke would not be perceived as funny from a human perspective. If the jokes contained a joke-like topic, e.g., an award-winning scarecrow, this was removed in a second step (*minus topic*) by replacing it with an everyday entity, e.g., a scientist, to achieve modification *B*. In case the original sample did not contain an unrealistic joke-specific topic, such as "*Why did the man put his money in the freezer?*", it was included in set *B* and not *A*. Thus, samples of the set *A* contained a joke topic but no comic element ($N = 19$), and samples of the set *B* included none of both ($N = 25$).

Eliminating the question-answer format for modification *C*, i.e., *minus structure*, was straightforwardly possible for all 25 original jokes ($N = 25$) by rewriting it in the format "[sentence one] *because* [sentence two]." In this case, the pun remained intact, and the joke was similarly funny. The original topic remained unchanged. Then, the comic element, i.e., the wordplay, was again removed to form set *D* ($N = 25$).

4 Results

With this design at hand, let us now turn to our empirical evidence gathered on joke generation, explanation, and detection.

4.1 Joke Generation

To test how rich the variety of ChatGPT's jokes is, we asked it to tell a joke a thousand times. All responses were grammatically correct. Almost all outputs contained exactly one joke. Only the prompt *do you know any good jokes?* provoked multiple jokes, leading to 1008 responded jokes in total. Besides that, the variation of prompts did have any noticeable effect.

To extract the deviation of jokes in the set of responses, similar samples were grouped. Removing direct duplicates reduced the number of individual samples to 348. Then, we removed opening sentences, such as *"How about this one"* in the example in Fig. 1, and minor formatting differences, such as extra line breaks. This resulted in a list of 128 individual responses. Finally, some samples could again be grouped together, such as in Ex.1.

Example 1. The following samples are no direct duplicates, as the wording is slightly different. However, they represent the same pun and are therefore grouped together.

(1.1) *Why did the bicycle fall over?*
Because it *was two-tired.*

(1.2) *Why didn't the bicycle* ***stand up by itself****?*
Because it was ***two tired****.*

These steps resulted in a final list of 25 top frequent jokes.

Top 25 Jokes. The final list of the top 25 jokes covered 917 of 1008 samples and can be found in the Appendix in Sec. B. Jokes are presented together with their number of occurrences. These are the five most frequent jokes:

T1. *Why did the scarecrow win an award? Because he was outstanding in his field.* (140)

T2. *Why did the tomato turn red?*
Because it saw the salad dressing. (122)

T3. *Why was the math book sad?*
Because it had too many problems. (121)

T4. *Why don't scientists trust atoms?*
Because they make up everything. (119)

T5. *Why did the cookie go to the doctor?*
Because it was feeling crumbly. (79)

The number of occurrences among these examples varies largely. While the top four jokes occurred over a hundred times each, the jokes T13 to T25 occurred less than 20 times. All 25 puns together sum up to about 90% of the gathered puns, but the top four examples alone make more than 50%. This observation contradicts our initial hypothesis: In the case of randomly picking from a predefined list, we would expect the occurrence of samples to be more equally distributed. Nevertheless, the small number of repeating samples indicates a limited versatility in ChatGPT's response pattern.

All of the top 25 samples are existing jokes. They are included in many different text sources, e.g., they can immediately be found in the exact same wording in an ordinary internet search. Therefore, these examples cannot be considered original creations of ChatGPT.

Of 1008 samples, 909 were identical to one of the top 25 jokes. The remaining 99 samples, however, did not necessarily contain new content. About half of them were again modifications of the top jokes, as illustrated by the examples Ex. 2, Ex. 3, and Ex. 4. While some of the modified puns still made sense and mostly just replaced parts of the original joke with semantically similar elements, others lost their conclusiveness. Thus, although the top 25 joke samples rather appear to be replicated than originally generated, there seems to be original content in the remaining samples.

Example 2. Item 2.1 is the famous chicken joke - a so-called anti-joke: It creates expectations with its joke-typical build-up but omits the reliving punch line. Besides that original joke, many variations exist with the chicken in another situation but a similar-sounding anti-pun. Item 2.2 is such a variation and is even more represented in the set of generated samples than in the original joke. Items 2.3, 2.4, and 2.5 are not covered by the top 25 jokes and can be considered modifications by ChatGPT, e.g., by replacing "chicken" in Item 2.2 with a semantically similar concept, i.e., "duck," to create Item 2.5.

(2.1) *Why did the chicken cross the road?*
To get to the other side. (7)

(2.2) *Why did the chicken cross the playground?*
To get to the other slide. (33)

(2.3) *Why did the duck cross the road?*
To get to the other pond. (2)

(2.4) *Why did the chicken wear a tuxedo?*
Because it was a formal occasion. (1)

(2.5) *Why did the duck cross the playground?*
To get to the other slide. (1)

For anti-jokes, it is especially hard to tell whether a sample is valid, as they do not compute in the classical sense. Yet, it is safe to say that the first two items are already existing jokes, and the latter ones are rather rare or even generated by ChatGPT.

Example 3. While it is debatable whether we observe that behavior in Ex. 1, Ex 2. clearly illustrates how ChatGPT mixes up elements from different valid jokes and, by that means, creates new samples. Item 3.1 is one of the top 25 jokes. The Items 3.2 and 3.3 were in the remaining 99 not-grouped jokes.

(3.1) *Why did the man put his money in the freezer? He wanted cold hard cash.* (13)

(3.2) *Why did the man put his watch in the blender? He wanted to make time fly.* (2)

(3.3) *Why did the man put his money in the blender? He wanted to make time fly.* (2)

Item 3.1 and Item 3.2 are valid replicated jokes, and Item 3.3 is an invalid combination of those.

Example 4. One of the most frequent jokes in the output list is the math book pun, Item 4.1. There were numerous valid modifications of that joke with minor grammatical variations, such as Item 4.2. It remains unclear whether all variations are replicated from training data sources or only item 4.1 and the others were modifications by ChatGPT. In Item 4.3, the "math book" is replaced by a "physics book." This change is unnecessary and dulls the punch line, but the joke is still valid, as also physics books are concerned with mathematical problems. Item 4.4, in contrast, rather appears as an insult than a joke. There is no wordplay included. ChatGPT exceeds the joke template and generates an invalid output.

(4.1) *Why was the math book sad?*
Because it had too many problems. (92)

(4.2) *Why did the math book look so mad?*
Because it had too many problems. (9)

(4.3) *Why was the* ***physics*** *book sad?*
Because it had so many problems. (1)

(4.4) *Why was the physics book sad?* ***Because it had so many formulas to memorize and so few friends to share them with.*** *(3)*

The examples Ex. 1 to Ex. 4 illustrate how ChatGPT starts from replicated joke examples and modifies them to original but rather invalid responses. If a user openly asks for a pun, most responses consist of replicated puns, and comparably few examples are modified jokes or original content. However, the modification of existing jokes can be forced to happen, e.g., by specifying the prompt. The following examples, Ex. 5, Ex. 6, and Ex. 7, additionally illustrate how ChatGPT is able to deploy the learned joke structure to create original content. These examples were not part of the initial 1008 samples but were created manually to illustrate that behavior.

Example 5. ChatGPT is able to fit different scenarios in the question-answer joke template.

P: Can you tell me a joke about cats?

(5.1) *Why did the cat put his paw in the air? Because he wanted to catch a snowflake!*

The scenario of a cat catching a snowflake is a realistic picture. The joke structure is similar to the reproduced puns. Nevertheless, the joke is missing its comic element and would most likely not work as a joke in a real conversation.

Example 6. In some cases, ChatGPT's modifications can work quite well.

P: Tell me a joke about physics students!

(6.1) *Why did the physics student break up with their lab partner? There was no chemistry!*

ChatGPT's output is most likely based on another existing joke that comes with the question: *"Why did the physics teacher break up with the biology teacher?"*. Although the punch line in Item 6.1 is not as sharp as in the original joke, it still holds the wordplay and is partly valid as a joke.

Example 7. ChatGPT can adapt to a wide variety of specific contexts, such as the field of ML. However, the further away the content is from the known jokes, the more likely it seems to lack any comic element.

P: Tell me a joke about machine learning!

(7.1) *Why did the machine learning algorithm cross the road? To get to the other side of the training data set!*

In Item 7.1, we observe how highly specific vocabulary is integrated into the previously seen chicken pun format. Repeating the same prompt yielded different outputs, which are all in the same textual structure, but with different content.

(7.2) *Why was the machine learning algorithm always so pessimistic? Because it had a negative bias!*

(7.3) *Why did the neural network go to therapy? Because it was suffering from overfitting!*

The samples are quite reasonable in terms of content, as, for instance, biases in ML systems. On top of that, they even contain basic implementations of wordplays, e.g., the double meaning of *suffering* in the context of disorders and in the context of machine learning overfitting. Nevertheless, the puns lack the last nuances to constitute an actually valid joke.

4.2 Joke Explanation

We asked ChatGPT to explain each of the 25 top jokes, which mostly resulted in a valid explanation. Wordplays and double meanings were identified and pointed out correctly, such as in Ex. 8.

Example 8. ChatGPT correctly names the double meaning of the term "problems" as the comic element of the pun and explains both meanings.

P: Can you explain why this joke is funny: Why was the math book sad? Because it had too many problems.

(8.1) *This joke is funny because it's a play on words. The word "problems" is being used in two different ways: as a mathematical term (referring to mathematical exercises) and as an idiomatic expression (referring to difficulties or challenges in life). [...]*

For 23 of 25 valid jokes, the explanations were accurate and reasonable. ChatGPT did not only explain direct but also acoustic double meanings correctly, such as "too tired" and "two-tired" or "peeling" and "feeling." A special case is the explanation of the chicken-anti-joke and its modifications that were already discussed in Ex. 2.

Example 9. The chicken joke is a famous anti-joke. This common knowledge is reflected in the model and the joke, therefore, explained correctly.

P: Can you explain why this joke is funny: Why did the chicken cross the road? To get to the other side.

(9.1) *The humor in this joke lies in the unexpected and straightforward punchline. When someone hears the setup of the joke, they are likely expecting a clever or surprising answer to why the chicken crossed the road. However, the punchline is just a simple and obvious explanation - the chicken crossed the road to get to the other side.*
The joke is an example of an "anti-joke," a type of joke that intentionally avoids a typical punchline and instead delivers a mundane or literal answer. In this case, the answer is so obvious that it's unexpected and therefore amusing to many people.

Example 10. However, for the similarly popular variations of the chicken joke, ChatGPT fails to recognize this connection. It only focuses on the content of the given sample and creates an abstruse explanation:

P: Can you explain why this joke is funny: Why did the chicken cross the playground? To get to the other slide.

(10.1) *The joke is a play on words, using the double meaning of the word "slide". In a playground, there are often two types of slides - a playground slide that children can slide down and a baseball slide, which is a sliding technique used in baseball.* [...]

This example illustrates how ChatGPT tries to apply an approved scheme of reasoning to a situation where it is not suitable.

Example 11. ChatGPT does not only come up with a convincing explanation for valid jokes but as well for invalid ones. This sample was one of the few outputs among the generated 1008 jokes from Sec. 4.1 that were considered invalid. It only occurred once.

P *Can you explain why this joke is funny:*
Why did the cookie go to the gym?
To get a-cookie-dized.

(11.1) *The joke is a play on words. "To get "a-cookie-dized" is a pun on the phrase "to get categorized", with "cookie" being substituted for "categorized." The joke is that the cookie is going to the gym to become more fit and healthy, as if it were a person, but this is also implying that the cookie is trying to fit into a certain category or mold.*

The joke itself is hard to understand, and the explanation even harder. ChatGPT's explanation does not reveal any humorous elements. The highlighted wordplay makes no sense from a human perspective. The system seems unable to contradict that this is not a funny joke or to admit that it cannot identify any pun. Instead, it creates a fictional but convincing-sounding explanation. Further examples can be found in the Appendix in Sec. C.

4.3 Joke Detection

The results of Sec. 4.2 indicate that ChatGPT will most likely not object if a presented sample is not valid, leading to the question of whether the system is even *able* to differentiate jokes from not-funny but joke-like samples. Therefore, we built five different conditions (see Fig. 2) of each joke, including the valid jokes themselves, and asked the system to classify each sample.

ChatGPT's responses were diverse and contained specifics regarding sentence structure or stylistic elements. In terms of joke classification, however, three categories of responses stood out:

- **[P] Joke:** The sample is immediately and explicitly recognized as a joke or a pun.
- **[H] Potentially funny:** The humorous potential of the sentence is mentioned secondary.
- **[N] No Joke:** The sample is neither classified as a joke or pun, nor is its comic potential mentioned in the response.

Fig. 2 illustrates the modification steps and the overall results of this task. A more detailed description of the categories, as well as all considered modifications and their classification, are given in the Appendix in Sec. D.

All original 25 jokes were clearly classified as a joke. This is not much surprising since each of the presented samples was output by ChatGPT as an exemplary joke in an earlier task. However, it serves as an affirmation of the validity of this task and of ChatGPT's confidence in reasoning. Two of the modification sets, namely modification *A* and modification *C*, show mixed classifications. These are the two conditions where one of the three characteristics was removed, and the other two remained unchanged. In both cases, the classifications of jokes are relatively equally divided into *jokes* and *no jokes*, with a slight tendency to the latter. Only a few samples were categorized as *potentially humorous*. For the remaining modification sets, i.e., set *B* and set *D*, each with two characteristics removed, all included samples were classified as *no joke*. None of the 25 samples per set was classified as *joke* or *potentially humorous*.

Thus, one single joke characteristic alone is not sufficient for ChatGPT to classify a sample as a joke. This applies to both a joke-typical structure and a joke-typical topic. In the case of two joke characteristics, the classification results were mixed, and all samples with three joke characteristics were classified as a joke.

5 Discussion

We aimed to understand ChatGPT's ability to capture and reflect humor. The results from three prompt-based tasks show implications regarding the system's inner workings.

Joke Generation. More than 90% of the generated samples were the same 25 jokes. This recurrence supports the initial impression that jokes are not originally generated. Presumably, the most frequent instances are explicitly learned and memorized from the model training, e.g., in the RLHF step that substantially contributes to ChatGPT's revolutionary capabilities. If and to what extent a generated output is reproduced from training data is a non-trivial question. If we get the opportunity to access further training details, we will test that subsequent hypothesis.

Nevertheless, we also observed jokes that were modified or generated by ChatGPT. This and the uneven distribution of output samples do not support the initial hypothesis of hard-coded jokes. ChatGPT generalizes characteristics of the learned top jokes, e.g., semantic format and wordplay puns, and can squeeze new topics into the known pattern. Although these are valid joke characteristics, it is quite a one-sided conception of jokes and even more of humor in general. Thus, ChatGPT understands this specific kind of joke quite well but fails to reflect a larger spectrum of humor.

Joke Explanation. The model is able to grasp and explain stylistic elements, such as personifications and wordplays, impressively well. Yet, there are obvious limitations: ChatGPT struggles to explain sequences that do not fit into the learned patterns. Further, it will not indicate when something is *not* funny or that it lacks a valid explanation. Instead, it comes up with a fictional but convincing-sounding explanation, which is a known issue with ChatGPT.

Joke Detection. We identified three main characteristics that generated jokes had in common, i.e., structure, wordplay, and topic. The presence of a single joke-characteristic, e.g., the question-answer template, is not sufficient for a sample to be wrongly classified as a joke. The fact that ChatGPT was not misled by such surface characteristics shows that there is indeed a certain understanding of humorous elements of jokes. With more joke characteristics, a sample is more likely to be classified as a joke.

Although ChatGPT's jokes are not newly generated, this does not necessarily take away from the system's capabilities. Even we humans do not invent new jokes on the fly but mostly tell previously heard and memorized puns. However, whether an artificial agent is able to *understand* what it learned is an exceptionally tough question and partly rather philosophical than technical.

In the present experiments, all prompts were posted in an empty, refreshed chat to avoid uncontrolled priming. But, clearly, context plays an important role in the perception of humor. ChatGPT is able to capture contextual information and adjust its responses accordingly to the preceding course of conversation. This is an intriguing capacity, which we would like to include in future investigations.

6 Conclusion

We test ChatGPT's ability to recognize and reflect human humor. The model is able to correctly identify, reproduce, and explain puns that fit into the learned pattern, but it fails to meet puns of other kinds, resulting in a limited reflection of humor. Also, it cannot yet confidently create intentionally funny original content.

The observations of this study illustrate how ChatGPT rather learned a specific joke pattern instead of being able to be actually funny. Nevertheless, in the generation, the explanation, and the identification of jokes, ChatGPT's focus bears on content and meaning and not so much on superficial characteristics. These qualities can be exploited to boost computational humor applications. In comparison to previous LLMs, this can be considered a huge leap toward a general understanding of humor.

We plan to conduct similar tasks on newly released GPT4 models [11] and on equivalent open source models, such as LLaMa [20] and GPT-NeoX [1], to compare their capabilities regarding joke generation and understanding.

Limitations

The present study comes with two major limitations. First, humor is highly subjective, and a valid and reliable evaluation is hard. Things can be perceived as funny for very different reasons - even for being particularly not funny, such as anti-jokes. Thus, when ChatGPT generates an odd joke about ML, one could even argue that ChatGPT has a sense of humor that is just different from ours. Also, humor is diverse in reality. The present investigation focuses on one specific form of humor, namely standalone jokes. There are more manifestations to consider, which would require a much more complex experimental setup.

Second, we cannot confidently trace back the outcome of the system or map it to specific input data. This is challenging for large data-driven models in general, but especially in this case, where we neither have access to the model itself nor to any training data or to the exemplary samples from RLHF. This prompt-based investigation creates a good intuition for the opportunities and limitations of ChatGPT. However, our observations and conclusions are solely based on system outputs. Further insights are needed to truly understand those relations.

Ethics Statement

ChatGPT has achieved massive public attention and societal impact, as people use the tool for all different kinds of tasks. This impact comes with a huge responsibility and risks, such as discriminating biases or spreading misinformation.

However, the system fails to meet the requirements of open science, as data, training details, and model characteristics are kept private. We, therefore, consider our work an important contribution to understanding ChatGPT's capabilities and objectively highlight its potential and limitations.

Acknowledgements

The authors would like to thank the anonymous reviewers for the helpful comments. This work benefited from the Hessian Ministry of Higher Education, and the Research and the Arts (HMWK) cluster projects "The Adaptive Mind" and "The Third Wave of AI".

References

[1] Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, et al. 2022. Gpt-neox-20b: An open-source autoregressive language model. *Challenges & Perspectives in Creating Large Language Models*, page 95.

[2] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

[3] Giancarlo Frosio. 2023. The artificial creatives: The rise of combinatorial creativity from dall-e to gpt-3. *Handbook of Artificial Intelligence at Work: Interconnections and Policy Implications (Edward Elgar, Forthcoming).*

[4] Xu Guo, Han Yu, Boyang Li, Hao Wang, Pengwei Xing, Siwei Feng, Zaiqing Nie, and Chunyan Miao. 2022. Federated learning for personalized humor recognition. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(4):1–18.

[5] Md Kamrul Hasan, Sangwu Lee, Wasifur Rahman, Amir Zadeh, Rada Mihalcea, Louis-Philippe Morency, and Ehsan Hoque. 2021. Humor knowledge enriched transformer for understanding multimodal humor. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12972–12980.

[6] Sophie Jentzsch, Patrick Schramowski, Constantin Rothkopf, and Kristian Kersting. 2019. Semantics derived automatically from language corpora contain human-like moral choices. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 37–44.

[7] Sophie Jentzsch and Cigdem Turan. 2022. Gender bias in bert-measuring and analysing biases through sentiment rating in a realistic downstream classification task. In *Proceedings of the 4th Workshop on Gender Bias in Natural Language Processing (GeBNLP)*, pages 184–199.

[8] Yuta Kayatani, Zekun Yang, Mayu Otani, Noa Garcia, Chenhui Chu, Yuta Nakashima, and Haruo Takemura. 2021. The laughing machine: Predicting humor in video. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2073–2082.

[9] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

[10] JA Meaney, Steven Wilson, Luis Chiruzzo, Adam Lopez, and Walid Magdy. 2021. Semeval 2021 task 7: Hahackathon, detecting and rating humor and offense. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 105–119.

[11] OpenAI. 2023. Gpt-4 technical report.

[12] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*.

[13] Victor Raskin. 1979. Semantic mechanisms of humor. In *Annual Meeting of the Berkeley Linguistics Society*, volume 5, pages 325–335.

[14] G Ritchie, R Manurung, H Pain, and D O'Mara. 2006. Computational humor. *IEEE intelligent systems*, 21(2):59–69.

[15] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2021. High-resolution image synthesis with latent diffusion models.

[16] Brandon M Savage, Heidi L Lujan, Raghavendar R Thipparthi, and Stephen E DiCarlo. 2017. Humor, laughter, learning, and health! a brief review. *Advances in physiology education*.

[17] Patrick Schramowski, Cigdem Turan, Nico Andersen, Constantin A Rothkopf, and Kristian Kersting. 2022. Large pre-trained language models contain human-like biases of what is right and wrong to do. *Nature Machine Intelligence*, 4(3):258–268.

[18] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv e-prints*, pages arXiv–1707.

[19] Oliviero Stock and Carlo Strapparava. 2005. Hahacronym: A computational humor system. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 113–116.

[20] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

[22] Tony Veale. 2021. *Your Wit is My Command: Building AIs with a Sense of Humor*. Mit Press.

[23] Jonas Wagner and Sina Zarrieß. 2022. Do gender neutral affixes naturally reduce gender bias in static word embeddings? In *Proceedings of the 18th Conference on Natural Language Processing (KONVENS 2022)*, pages 88–97.

[24] Xingbo Wang, Yao Ming, Tongshuang Wu, Haipeng Zeng, Yong Wang, and Huamin Qu. 2021. Dehumor: Visual analytics for decomposing humor. *IEEE Transactions on Visualization and Computer Graphics*, 28(12):4609–4623.

[25] Orion Weller and Kevin Seppi. 2019. Humor detection: A transformer gets the last laugh. pages 3621–3625.

[26] Thomas Winters. 2021. Computers learning humor is no joke. *Harvard Data Science Review*, 3(2).

[27] Dolf Zillmann and Jennings Bryant. 1980. Misattribution theory of tendentious humor. *Journal of Experimental Social Psychology*, 16(2):146–160.

A Access Dates

Models such as ChatGPT are constantly approved and changed. Thus, observations made on one day are not necessarily similarly valid on another day. Therefore, we list the dates of experimental access as precisely as possible in the following. All dates are in 2023.

Joke Generation: 22. - 31. January
Joke Explanation: 03. - 13. February
Joke Detection: 23. February - 01. March

B Joke Generation - Top 25 Jokes

The majority of generated samples were the same 25 puns, which are presented in the following as T1 - T25 together with each number of occurrence:

T1. *Why did the scarecrow win an award?*
Because he was outstanding in his field. (140)

T2. *Why did the tomato turn red?*
Because it saw the salad dressing. (122)

T3. *Why was the math book sad?*
Because it had too many problems. (121)

T4. *Why don't scientists trust atoms?*
Because they make up everything. (119)

T5. *Why did the cookie go to the doctor?*
Because it was feeling crumbly. (79)

T6. *Why couldn't the bicycle stand up by itself?*
Because it was two-tired. (52)

T7. *Why did the frog call his insurance company?*
He had a jump in his car. (36)

T8. *Why did the chicken cross the playground?*
To get to the other slide. (33)

T9. *Why was the computer cold?*
Because it left its Windows open. (23)

T10. *Why did the hipster burn his tongue?*
He drank his coffee before it was cool. (21)

T11. *Why don't oysters give to charity?*
Because they're shellfish. (21)

T12. *Why did the computer go to the doctor?*
Because it had a virus. (20)

T13. *Why did the banana go to the doctor?*
Because it wasn't peeling well. (19)

T14. *Why did the coffee file a police report?*
Because it got mugged. (18)

T15. *Why did the golfer bring two pairs of pants?*
In case he got a hole in one. (13)

T16. *Why did the man put his money in the freezer?*
He wanted cold hard cash. (13)

T17. *Why don't seagulls fly over the bay?*
Because then they'd be bagels. (13)

T18. *Why did the chicken go to the seance?*
To talk to the other side. (11)

T19. *Why was the belt sent to jail?*
Because it held up a pair of pants. (11)

T20. *Why did the chicken cross the road?*
To get to the other side. (7)

T21. *Why did the computer go to the doctor?*
Because it had a byte. (6)

T22. *Why did the cow go to outer space?*
To see the moooon. (6)

T23. *Why did the man put his money in the blender?*
He wanted to make liquid assets. (6)

T24. *Why don't skeletons fight each other?*
They don't have the guts. (5)

T25. *What do you call an alligator in a vest?*
An investigator. (5)

C Joke Explanation

In this task, ChatGPT was asked to explain each of the top 25 generated jokes. The prompt for each joke is "*Can you explain why this joke is funny: [joke].*" This prompt is not noted for each joke in the following, as it is redundant. ChatGPT's responses are complete. Only the formatting of texts might be changed in favor of the paper structure.

The majority, i.e., 23 of 25 generated explanations, were valid. Only the explanations E8, which is discussed in Ex. 9.1 in Sec. 4.2, and E16 are not accurate. A selection of exemplary explanations is presented in this section. The explanation numbers match the number in the top joke list in Sec. B. The Explanations E3, E8, and E20 are discussed in the main paper. The comprehensive list of explanations is provided in the project repository[4].

E2. Why did the tomato turn red? Because it saw the salad dressing. *The joke is based on a pun or wordplay. When a tomato ripens, it turns from green to red. The question "Why did the tomato turn red?" sets up an expectation that there will be a logical reason, such as exposure to sunlight or reaching maturity. However, the punchline "Because it saw the salad dressing" is unexpected*

---

[4] *will be added after the anonymity period ends.*

and humorous because it is a play on words. The word "dressing" has a double meaning here; it can refer to the salad dressing (a condiment typically used on salads) or the act of dressing up or looking nice. So the joke is implying that the tomato saw the dressing (meaning the nice-looking salad) and got embarrassed, blushing and turning red.

Overall, the joke is a lighthearted play on words that relies on the listener's understanding of the multiple meanings of the word "dressing" to create a humorous surprise.

E3. Why was the math book sad? Because it had too many problems. *This joke is a play on words, specifically on the word "problems." In mathematics, a "problem" typically refers to a question or exercise that requires a solution, whereas in everyday language, "problems" can refer to difficulties or challenges. In the joke, the math book is said to be "sad" because it has too many "problems."*

While the word "problems" in this context could refer to the mathematical exercises in the book, the joke also suggests that the book is sad because it has too many difficulties or challenges to deal with. The double meaning of "problems" is what makes the joke humorous. Additionally, the idea of a book having emotions like sadness is personification, which is another common element of jokes. The unexpected pairing of a book and emotions can also contribute to the humor of the joke.

E6. Why couldn't the bicycle stand up by itself? Because it was two-tired. *The joke is a play on words, specifically on the word "tired." In the context of the joke, "two-tired" sounds like "too tired," which means being exhausted or lacking energy. However, the actual meaning of "two-tired" in the joke is that the bicycle has two tires instead of a stand or kickstand to support it.*

So, the punchline is unexpected because it uses a homophone (a word that sounds the same but has a different meaning) to create a double meaning in the sentence. This incongruity between what we expect to hear and what is actually said is what makes the joke humorous.

E8. Why did the chicken cross the playground? To get to the other slide. *The joke is a play on words, using the double meaning of the word "slide". In a playground, there are often two types of slides - a playground slide that children can slide down and a baseball slide, which is a sliding technique used in baseball. The joke sets up the expectation that the chicken is crossing the playground to get to the other side, which is a common setup for a joke. However, the punchline surprises the listener by using the word "slide" in a different way, revealing that the chicken actually wants to use the playground slide. The unexpected twist on a familiar joke structure, along with the clever wordplay, makes the joke humorous.*

E16. Why did the man put his money in the freezer? He wanted cold hard cash. *This joke is funny because it plays with the multiple meanings of the phrase "cold hard cash."*

Literally, "cold hard cash" refers to physical currency that is cold to the touch and difficult to bend or manipulate, as opposed to checks or digital payments. However, in this joke, the phrase takes on a figurative meaning as well, suggesting that the man wanted his money to be "cold" or safe from theft, and "hard" or difficult for anyone to access.

By placing his money in the freezer, the man achieves both meanings of "cold hard cash" in a humorous way. The pun on the phrase is unexpected and clever, making the joke amusing.

E20. Why did the chicken cross the road? To get to the other side. *The humor in this joke lies in the unexpected and straightforward punchline. When someone hears the setup of the joke, they are likely expecting a clever or surprising answer to why the chicken crossed the road. However, the punchline is just a simple and obvious explanation - the chicken crossed the road to get to the other side.*

The joke is an example of an "anti-joke," a type of joke that intentionally avoids a typical punchline and instead delivers a mundane or literal answer. In this case, the answer is so obvious that it's unexpected and therefore amusing to many people.

E22. Why did the cow go to outer space? To see the moooon. *This joke is a play on words and relies on the similarity in sound between "moo" (the sound a cow makes) and "moon" (the natural satellite that orbits the Earth). The punchline, "to see the moooon," is humorous because it combines the idea of a cow going to outer space with a pun that emphasizes the word "moon" in a way that references the cow's vocalization.*

Additionally, the joke is also amusing because it presents a nonsensical scenario of a cow traveling to outer space, which is unexpected and somewhat absurd. The humor comes from the surprise and *playfulness of the wordplay and the unexpectedness of the punchline.*

D Joke Detection

In the joke detection task, ChatGPT was openly asked to categorize different samples. The samples were in one of five conditions: The original joke, (A) with removed wordplay, (B) with a removed topic (which entails removed wordplay), (C) removed structure, and (D) removed structure & wordplay. In set C, the joke is still valid, as only the sentence structure but no component of the comic element is modified. All other modifications result in "unfunny" variations of the original joke. Thus, from a human perspective, the original jokes, as well as samples from set C, would probably be categorized as jokes, and samples from sets A, B, and D not.

All modifications that are considered in this investigation are listed in the remainder of this chapter, together with their classification. The original jokes are classified as a joke without exception and can be found in Sec. B. An overview of sample sets and results is displayed in Tab. 1.

The input prompt for each sample was "*What kind of sentence is that:* [sample]." ChatGPT's responses were diverse and could contain individual explanations of sentence structure or stylistic elements. In terms of joke classification, however, responses could be grouped into three categories: Joke or pun, potentially humorous, and no joke. These classes are defined as follows.

[P] Joke/ Pun ChatGPT immediately classifies the sample as a joke or pun with a statement like *The sentence "[...]" is a joke or a play on words. It is a type of humor known as a "pun."*. The response might contain information about the semantic structure, like *It is a question-and-answer format, where the question sets up the joke and the answer delivers the punchline*. In some cases, an explanation of the comic workings is also delivered, as in *The sentence is a joke or a pun, which is a form of wordplay that exploits multiple meanings of a term, or of similar-sounding words, for humorous effect. In this case, the word "[...]" is used in two ways: first, [...]; and second, [...], which creates a play on words*. Either way, the response clearly states the sample to be a joke or a pun, or both.

[H] Potentially Humorous In some responses, ChatGPT only mentions the humorous potential of the input statement secondary. It does not immediately label the sample to be a joke or pun, e.g., *The sentence "[...]" is a declarative sentence. It makes a statement or expresses a fact. In this case, the sentence describes an action taken by a chicken, which is unusual and may be intended to be **humorous or absurd***. In some cases, the focus is on stylistic elements, such as in *The sentence is a personification, where an inanimate object is given human-like qualities and actions. [...], which are actions typically associated with humans*.

[N] No Joke ChatGPT does neither classify the sample as a joke or pun nor mention the comic potential of the input sentence. In this context, "no joke" does not mean that the sample is explicitly stated to be *not* a joke, but no similar quality is mentioned at all. Instead, the response focuses on grammatical structures or similar. Exemplary responses for this category are *The sentence "..." is a declarative sentence. It presents a statement or a fact and ends with a period* or *The sentence "[...]" is a dialogue or a conversation between two people where one person is asking a question and the other person is answering it.*

D.1 Modification A - Minus Wordplay

To create these samples, the wordplay was removed from each original joke by minimally altering the wording. This modification also eliminates the comic element of jokes, which is they would most likely not be perceived as funny by humans.

[N] *Why did the scarecrow win an award?*
Because he did very good work.

[N] *Why did the tomato turn red?*
Because it had a lot sun recently.

[P] *Why was the math book sad?*
Because it was a rainy day.

[P] *Why don't scientists trust atoms?*
Because they tend to lie.

[P] *Why did the cookie go to the doctor?*
Because it was feeling unwell.

[P] *Why couldn't the bicycle stand up by itself?*
Because it didn't have racks.

[H] *Why did the frog call his insurance company?*
He had a scratch in his car.

[N] *Why was the computer cold?*
Because the heater was broken.

Condition	N	Included Characteristics			Classification		
		wordplay	structure	topic	[P]	[H]	[N]
Original	25	✓	✓	✓	25	0	0
Mod A	19	✗	✓	✓	8	1	10
Mod B	25	✗	✓	✗	0	0	25
Mod C	25	✓	✗	✓	9	4	12
Mod D	25	✗	✗	✓	0	0	25

Table 1: Classification results of the joke detection task: *Condition* is either the original sample, i.e., one of the top 25 jokes, or one of four modifications. *Included characteristics* illustrates which of the three identified characteristics are included. N is the number of samples per condition. The *classification* is either [P] *joke*, [H] *potentially funny*, or [N] *no joke*.

[N] *Why did the hipster burn his tongue?*
He drank hot coffee.

[N] *Why don't oysters give to charity?*
Because they have no money.

[N] *Why did the computer go to the doctor?*
Because it was sick.

[P] *Why did the banana go to the doctor?*
Because it was sick.

[P] *Why did the coffee file a police report?*
Because it got robbed.

[N] *Why did the golfer bring two pairs of pants?*
In case one gets damaged.

[N] *Why don't seagulls fly over the bay?*
Because they are mostly living in ports.

[N] *Why was the belt sent to jail?*
Because it stole in a store.

[P] *Why did the computer go to the doctor?*
Because it had a headache.

[P] *Why did the chicken go to outer space?*
To see the moon.

[N] *Why don't skeletons fight each other?*
They are dead already.

D.2 Modification B - Minus Topic

To create the samples of this condition, the joke-typical and potentially humorous topic was removed from the jokes. For instance, in the case of personification, i.e., if the protagonist of the original joke was an object, such as a tomato or a computer, it was replaced by a natural, human protagonist, such as a teacher or a scientist. The typical question-answer format remained unchanged. However, just as in Modification A, the actual pun gets lost.

[N] *Why did the scientist win an award?*
Because he did very good work.

[N] *Why did the man turn red?*
Because he saw the neighbour dressing.

[N] *Why was the child sad?*
Because it had many problems.

[N] *Why don't scientists trust journalists?*
Because they make up everything.

[N] *Why did the woman go to the doctor? Because she was feeling crumbly.*

[N] *Why couldn't the man stand up by himself?*
Because he was drunk.

[N] *Why did the driver call his insurance company?*
He had a jump in his car.

[N] *Why did the kid cross the playground?*
To get to the other slide.

[N] *Why was the student cold?*
Because it was winter.

[N] *Why did the coworker burn his tongue?*
He drank his coffee too hot.

[N] *Why don't millionaires give to charity?*
Because they're selfish.

[N] *Why did the man go to the doctor?*
Because he had a virus.

[N] *Why did the teacher go to the doctor?*
Because he wasn't feeling well

[N] *Why did the driver file a police report?*
Because it got robbed.

[N] *Why did the athlete bring two pairs of pants?*
In case one gets dirty.

[N] *Why did the man put his money in the freezer?*
To hide it from intruders.

[N] *Why don't pigeon fly over the bay?*
Because they mostly live in cities.

[N] *Why did the daughter go to the seance?*
To talk to her mother.

[N] *Why was the cashier sent to jail?*
Because she held up a dress.

[N] *Why did the man cross the road?*
To get to the other side.

[N] *Why did the man go to the doctor?*
Because it had a bite.

[N] *Why did the astronaut go to outer space?*
To see the moon.

[N] *Why did the man put bananas in the blender?*
He wanted to make a smoothie.

[N] *Why don't schoolboys fight each other?*
They don't have the guts.

[N] *What do you call a man in a vest? A vest wearer.*

D.3 Modification C - Minus Structure

In these samples, only the sentence structure of the original jokes was altered in order to eliminate the prominent question-answer structure. Thus, the actual pun remained intact, and the joke topic was unchanged.

[P] *The scarecrow won an award because he was outstanding in his field.*

[N] *The tomato turned red because it saw the salad dressing.*

[N] *The math book was sad because it had too many problems.*

[P] *Scientists don't trust atoms because they make up everything.*

[N] *The cookie went to the doctor because it was feeling crumbly.*

[P] *The bicycle couldn't stand up by itself because it was two-tired.*

[P] *The frog called his insurance company because he had a jump in his car.*

[N] *The chicken crossed the playground to get to the other slide.*

[N] *The computer was cold because it left its Windows open.*

[N] *The hipster burned his tongue because he drank his coffee before it was cool.*

[P] *Oysters don't give to charity because they're shellfish.*

[N] *The computer went to the doctor because it had a virus.*

[N] *The banana went to the doctor because it wasn't peeling well.*

[H] *The coffee filed a police report because it got mugged.*

[N] *The golfer brings two pairs of pants in case he got a hole in one.*

[N] *The man put his money in the freezer because he wanted cold hard cash.*

[H] *Seagulls don't fly over the bay because then they'd be bagels.*

[H] *The chicken went to the seance to talk to the other side.*

[H] *The belt was sent to jail because it held up a pair of pants.*

[N] *The chicken crossed the road to get to the other side.*

[P] *The computer went to the doctor because it had a byte.*

[N] *The cow went to outer space to see the moooon.*

[P] *The man put his money in the blender because he wanted to make liquid assets.*

[P] *Skeletons don't fight each other because they don't have the guts.*

[P] *An alligator in a vest is called an investigator.*

D.4 Modification D - Minus Wordplay 2

Samples of this condition are again modifications from sample set D. Additionally to altering the sentence structure, the wordplay was removed, such as in Set A. This resulted in potentially unfunny, one-line statements.

[N] *The scientist won an award because she did good work.*

[N] *The man turned red because he saw his neighbour dressing.*

[N] *The child was sad because it was raining outside.*

[N] *Scientists don't trust journalists because they tend to lie.*

[N] *The teacher went to the doctor because he was ill.*

[N] *The man couldn't stand up by himself because he was drunk.*

[N] *The driver called his insurance company because he had a scratch in his car.*

[N] *The child crossed the playground to get to the other slide.*

[N] *The student was cold because the heater was broken.*

[N] *The coworker burned his tongue because he drank hot coffee.*

[N] *Millionaires don't give to charity because they are selfish.*

[N] *The woman went to the doctor because she was sick.*

[N] *The chef went to the doctor because he was sick.*

[N] *The driver filed a police report because she got robbed.*

[N] *The athlete brings two pairs of pants in case one gets damaged.*

[N] *The man put his money in the freezer to hide it from thieves.*

[N] *Pigeons don't fly over the bay because they are mostly living in cities.*

[N] *The daughter went to the seance to talk to the other side.*

[N] *The cashier was sent to jail because she held up a dress.*

[N] *The man crossed the road to get to the other side.*

[N] *The man went to the doctor because he had a bite.*

[N] *The astronaut went to outer space to see the moon.*

[N] *The man put his bananas in the blender because he wanted to make a smoothie.*

[N] *Schoolboys don't fight each other because they don't have the guts.*

[N] *A man in a vest is called a vest wearer.*

How to Control Sentiment in Text Generation: A Survey of the State-of-the-Art in Sentiment-Control Techniques

Michela Lorandi and **Anya Belz**
ADAPT Research Centre, Dublin City University
`{michela.lorandi, anya.belz}@adaptcentre.ie`

Abstract

Recent advances in the development of large Pretrained Language Models, such as GPT, BERT and Bloom, have achieved remarkable performance on a wide range of different NLP tasks. However, when used for text generation tasks, these models still have limitations when it comes to controlling the content and style of the generated text, often producing content that is incorrect, irrelevant, or inappropriate in the context of a given task. In this survey paper, we explore methods for controllable text generation with a focus on sentiment control. We systematically collect papers from the ACL Anthology, create a categorisation scheme based on different control techniques and controlled attributes, and use the scheme to categorise and compare methods. The result is a detailed and comprehensive overview of state-of-the-art techniques for sentiment-controlled text generation categorised on the basis of how the control is implemented and what attributes are controlled and providing a clear idea of their relative strengths and weaknesses. [1]

1 Introduction

In recent years, there has been a surge of interest in developing algorithms and models for Controllable Text Generation (CTG). This research field aims to enable users to generate text with specific attributes, controlling e.g. the text's sentiment, topic, or level of formality. In this survey paper, we focus on state-of-the-art CTG techniques that control sentiment. We provide a comprehensive overview of the existing literature and categorise approaches based on their implementation of control, and which specific attributes they control.

Our main contributions are as follows:

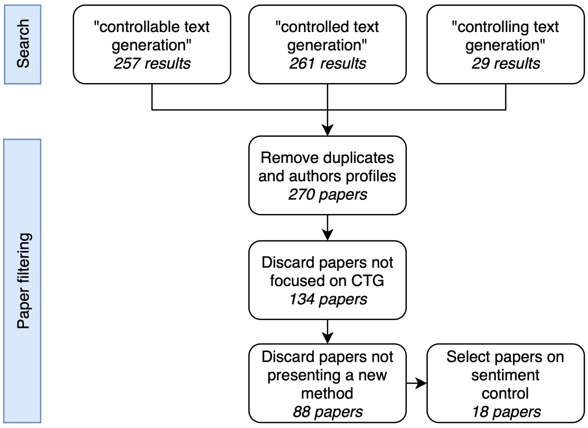

Search
"controllable text generation" 257 results
"controlled text generation" 261 results
"controlling text generation" 29 results
Paper filtering
Remove duplicates and authors profiles 270 papers
Discard papers not focused on CTG 134 papers
Discard papers not presenting a new method 88 papers
Select papers on sentiment control 18 papers

Figure 1: Paper selection process from ACL Anthology to which we added 1 paper (see in text).

- We propose a categorisation scheme for Sentiment-Controlled Text Generation specifying control attributes and how control is implemented.
- We analyse state-of-the-art techniques for Sentiment Control, and categorise each technique using the proposed categorisation scheme.
- We compare the selected papers in terms of performance, efficiency and generality.

The paper is structured as follows. Section 2 summarises two previous related survey papers, while Section 3 outlines the scope of the present survey, the method we used for systematic paper selection, and some high-level statistics for the selected papers. The proposed categorisation scheme is described in Section 4, consisting of (i) different types of controlled attributes (Section 4.1), and (ii) different types of control implementation techniques (Section 4.2). Section 5 describes the CTG techniques from the surveyed papers in terms of the categorisation scheme, including which attributes are controlled. Section 6 compares the different techniques in terms of their generality, performance, and efficiency. We conclude with suggested future directions (Section 7), some discussion (Section 8) and final remarks (Section 9).

---

[1] The categorised list of papers can be found in our GitHub repository `https://github.com/DCU-NLG/sentimentCTG-survey`

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 341–353
July 14, 2023 ©2023 Association for Computational Linguistics

2 Related Research

Prabhumoye et al. (2020) propose a schema of the language generation pipeline based on five components that control the generation process: external input, sequential input, generator operations, output, and training objective. They argue that control of specific attributes requires modification of these five components, and present an overview of existing control techniques in terms of which component different techniques use to exert the control. The work focuses on how the proposed schema can be applied to enable control of text generation with a particular focus on autoregressive models.

As part of a general introduction and overview of techniques in pretrained language model (PLM) based CTG and evaluation methods, Zhang et al. (2022) propose a set of control conditions (semantic, structural, lexical), and broadly group together methods for CTG into finetuning, retraining/refactoring, and postprocessing. The work addresses only Transformer-based PLMs, and distinguishes seven subtypes of methods, based on how the control signal works with the PLM.

In this survey, we consider all types of methods that have been used for sentiment-controlled Text Generation, not just Transformer-based PLMs, and we conduct a systematic paper selection process. We provide a categorisation scheme based on Control Attribute Types and Control Implementation Techniques that we use to characterise and compare the selected methods. Finally, we provide comparisons in terms of performance and efficiency.

3 Survey Scope and Paper Selection

This paper aims to fill a gap in the current literature by surveying recent models applied to Controllable Text Generation (CTG) with a specific focus on sentiment control. Furthermore, we propose a categorisation of the selected papers based on controlled attributes, and how the control is implemented.

We conducted an otherwise unrestricted search on the ACL Anthology using the keywords "controllable text generation," "controlled text generation" and "controlling text generation," as shown in Figure 1. After removing duplicates, authors' profiles, and non-paper resources, we obtained 270 papers. From this original pool, we discard papers that are not strictly related to CTG, such as papers that mention CTG but do not explore the task. Next we only retain papers which present a new model or control method, discarding those that e.g. only present a new dataset or perform a comparative study. After filtering, we are left with 88 papers from which for the present survey we select all papers implementing sentiment control, thus giving us 18 papers. We add one more relevant paper (Dathathri et al., 2019) not present in the ACL Anthology that was cited in our pool of papers. Table 1 lists the resulting 19 papers in the left-most column.

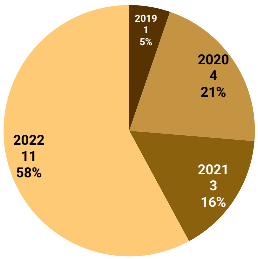


Figure 2: Distribution of selected papers across years.[2]

The 19 papers span the period 2019 (1 paper), 2020 (4 papers), 2021 (3 papers), and 2022 (11 papers), illustrating the rapidly growing interest in this topic, as shown in Figure 2 . Papers report work using Complete Training techniques (3 papers), Model Fine-Tuning (3), Disentanglement (1), Modification of Token Distribution (6), and Hybrid techniques (6). In 10 of the papers, multiple attributes are controlled simultaneously, whereas in 9, single attributes are controlled one at a time. 14 papers are designed for free text generation (rather than a specific task), 2 methods are for Story Generation, 2 for Conversational Agents, and one for Topic to Essay Generation. We return to properties of techniques in more detail in Section 6 and Table 1.

4 Categorisation Scheme

We collect all selected papers and annotate them based on different aspects, such as control attributes addressed and architecture used to solve the control problem with a specific focus on how the control is implemented and embedded in the proposed architecture. Using the collected information, we cluster control attributes and models to create a categorisation scheme for Sentiment-Controlled Text Generation in which we specify types of controlled attributes and types of control implementation. The created categorisation scheme will be used to cate-

---

[2] The proposed scheme is specific for Sentiment-Controlled Text Generation, but we are currently working on a general scheme for CTG.

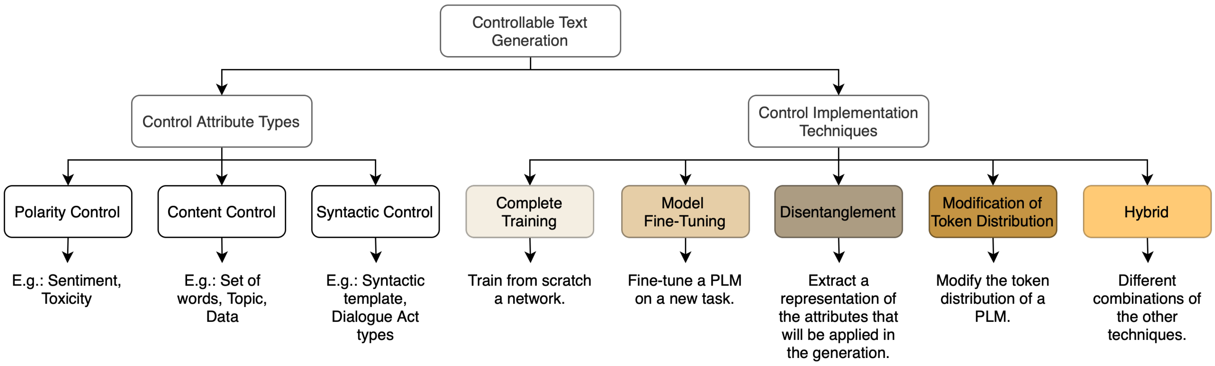

Controllable Text Generation
Control Attribute Types
Control Implementation Techniques
Polarity Control
Content Control
Syntactic Control
Complete Training
Model Fine-Tuning
Disentanglement
Modification of Token Distribution
Hybrid
E.g.: Sentiment, Toxicity
E.g.: Set of words, Topic, Data
E.g.: Syntactic template, Dialogue Act types
Train from scratch a network.
Fine-tune a PLM on a new task.
Extract a representation of the attributes that will be applied in the generation.
Modify the token distribution of a PLM.
Different combinations of the other techniques.

Figure 3: Categorisation of Sentiment-Controlled Text Generation methods[3] considering Control Attributes Types (left) and Control Implementation Techniques (right).[2]

gorise and describe the studied papers.

4.1 Types of controlled attributes

Sentiment-Controlled Text Generation methods control different attributes, such as sentiment, set of words, and topics. In order to help gain insights into different control attributes currently in use and how they are controlled and combined in different methods, we distinguish three Control Attribute Types (Figure 3 left): Polarity Control, Content Control, and Syntactic Control.

Polarity Control covers attributes that control whether otherwise similar content is expressed with positive vs. negative judgment, toxic vs. neutral attitude, different political stance, or one of several competing perspectives e.g. in sport. For example, the negative sentence *The office is poorly maintained and dirty at all times* and the positive sentence *The office is well maintained and clean at all times* have a different polarity, but are otherwise similar in content, i.e. both are about cleaning and maintenance of an office. In the 19 papers in our survey, we encountered two attribute of this type: *Sentiment* and *Toxicity*.

Content Control attributes control the core content of a text. For example, the set of words *{burger, good, fries}* may be used to enforce presence of these words in the output sentence, e.g.: *The **burger** was very **good** and the **fries** are excellent.* In our 19 papers we encountered 4 attriibutes of this type: *Topic*, *Action*, *Character*, *Need*.

Syntactic Control attributes control the structure, syntax, and grammar of the output sentence. For example, we can give the model a syntactic template *(S (S) (,) (CC) (S))* to generate the final text *the film is a visual treat, but almost unsurpassed.* (Yang et al., 2021). In our 19 papers, there was just one attribute of this type: *Tense*.

A system can in principle simultaneously control multiple attributes of the same or different types, thus enabling more fine-grained control. Table 1 lists the seven control attribute types encountered in the 19 papers in our survey, and which paper implements control over which attribute, in column 5.

4.2 Types of control implementation

We categorise controllable text generation techniques based on how they implement the control, as shown in Figure 3, right half, distinguishing four types: Complete Training, Model Fine-Tuning, Disentanglement, Modification of Token Distribution, and Hybrid.

Complete Training techniques train new models from scratch in order to obtain specialised models specifically trained for a Controllable Text Generation task. **Model Fine-Tuning** techniques use a pre-trained model which is fine-tuned to adapt it to the new task including control attributes. **Disentanglement** techniques extract a representation of the control attributes which is applied to steer text generation toward the specified attributes. **Modification of Token Distribution** techniques modify the token distribution of a pre-trained model in order to steer text generation. **Hybrid** techniques include two or more types of the above control implementation techniques.

Table 1 shows the control implementation type(s) addressed by each of the 19 papers in our survey in column 4.

---

[3] The colours have been tested with Adobe Color Accessibility Tools Color Blind Safe.

Work	**Model**	**Task**	**Control**	**Control Attributes**							**Sentiment Datasets**
				S	T	To	A	C	Te	N	
Complete Training											
Qiao et al. (2020)	CVAE	TEG	Multiple	✓	✓						ZHIHU (Feng et al., 2018)
Betti et al. (2020)	GAN	FT	Single	✓	✓						Movie Reviews (Socher et al., 2013), Customer Reviews (Hu and Liu, 2004)
Xie et al. (2022)	Enc-Dec	SG	Multiple	✓				✓		✓	Story Commonsense
Model Fine-Tuning											
Qian et al. (2022)	GPT-2	FT	Multiple	✓	✓	✓					PPLM Prompts
Gu et al. (2022b)	BERT	FT	Multiple	✓	✓	✓					PPLM Prompts
Fang et al. (2022)	MA	FT	Multiple	✓	✓				✓		Yelp
Disentanglement											
Yu et al. (2021)	GPT-2	FT	Multiple	✓	✓						PPLM Prompts
Modification of Token Distribution											
Dathathri et al. (2019)	MA	FT	Multiple	✓	✓						PPLM Prompts
Madotto et al. (2020)	MA	CA	Single	✓	✓						(Adiwardana et al., 2020) prompts
Goswamy et al. (2020)	GPT-2	FT	Multiple	✓	✓						-
Kumar et al. (2022)	MA	FT	Single	✓		✓					PPLM Prompts
Gu et al. (2022a)	MA	FT	Single	✓	✓	✓					PPLM Prompts
Landsman et al. (2022)	MA	FT	Single	✓							OpenWebText Corpus Prompts
Hybrid											
Wang et al. (2022)	BART	SG	Multiple	✓			✓	✓			ROCStory (Rashkin et al., 2018)
Tian et al. (2022)	Enc-Dec	CA	Single	✓							weibo.com (Shang et al., 2015)
Liu et al. (2021)	GPT-2	FT	Single	✓		✓					OpenWebText Corpus Prompts
Zhang and Song (2022)	GPT-2	FT	Single	✓		✓					OpenWebText Corpus Prompts
Krause et al. (2021)	MA	FT	Single	✓	✓	✓					Bookcorpus (Zhu et al., 2015)
Liu et al. (2022)	GPT-2	FT	Multiple	✓	✓						IMDb, OpeNER (Agerri et al., 2013), SenTube (Uryupina et al., 2014)

Table 1: Overview of methods for Sentiment Control in Text Generation. Models: MA=Model Agnostic. Tasks: SG=Story Generation, TEG=Topic to Essay Generation, FT=Free Text, CA=Conversational Agent. Control Attributes: S=Sentiment, T=Topic, To=Toxicity, A=Action, C=Character, Te=Tense, N=Need.

5 Characterisation of CTG Techniques

Table 1 provides an overview of the 19 papers we survey (listed in column 1), in terms of the control implementation technique used (headings inserted into the rows), the type of model used (column 2), the NLP task implemented by the model (column 3), the attributes controlled by the technique (column 5), whether a single or multiple attributes are controlled at a time (column 4), and finally the datasets used in training (column 6).

In the remainder of this section, we summarise each of the 19 papers in our survey in more detail, grouped together in terms of the control implementation technique used.

5.1 Complete Training

Betti et al. (2020) propose a text GAN composed of one generator and two discriminators. The generator is a Relational Memory with self-attention (Santoro et al., 2018) with the objective to generate text consistent with the specified control attribute. The syntax discriminator distinguishes between real and generated sentences, while the semantic discriminator assesses whether the generated sentence expresses the control attribute, e.g. positive sentiment. To solve the well-known problem of differentiation in GANs applied to text, the Gumbel-softmax trick (Jang et al., 2016) is applied. This approach enables control only for one attribute at a time and it has been evaluated on sentiment and topic control.

In order to enable multi-attribute control, Qiao et al. (2020) propose a Sentiment-Controllable topic-to-essay generator that deploys a Conditional Variational Auto-Encoder in adversarial training. The model simultaneously controls the topics of the essay and the sentiment of each sentence composing the essay. The topic control is achieved using a Topic Graph Attention, which includes a topic knowledge graph in the generation process. Sentiment control is achieved by injecting the sentiment representation both in the encoder and the decoder.

In a different direction, Xie et al. (2022) propose a psychology-guided story generation method that controls storytelling as the protagonist's psychological state changes. This technique enables multi-attribute control considering the protagonist of the story (Character), their chain of emotions (Emotion), and chain of needs (Need) representing the evolution of the psychological state of the protagonist. The model is an encoder-decoder architecture with the addition of psychology controllers designed to integrate the local and global psychological state into the story context representation.

5.2 Model Fine-Tuning

Model Fine-Tuning can be achieved in many ways. One way is to focus on prefix tuning, i.e. fine-tuning a model to extract continuous attribute-specific vectors that will be prepended to the activations of the pre-trained model to steer text generation. E.g., Qian et al. (2022) fine-tune GPT-2 (Radford et al., 2019) to obtain prefixes, but they use the contrast between prefixes, for example, positive vs negative sentiment, to encourage the desired attribute and discourage the opposite attribute. In this method, only the prefixes are trained and GPT-2 weights are kept frozen.

Similarly, Gu et al. (2022b) fine-tune BERT (Devlin et al., 2019) to obtain prefixes. The idea is to have an Autoencoder structure, i.e. the encoder-decoder reconstructs the input sentence, to map attribute-relevant sentences to latent representations of attributes. At inference time, the model searches the attribute representation in the attribute space and uses it as a prefix for the decoder. In the case of multiple attributes the intersection of attributes is taken as the prefix, instead of contrastive prefixes (Qian et al., 2022). In this setting, the decoder is fixed, while the encoder is fine-tuned to get the attribute representations.

Fang et al. (2022) further explore the usage of Variational Autoencoders to learn a latent representation of control attributes. The idea is to use contrastive learning to separate the latent space into several parts, thus obtaining learnable vectors associated with a control attribute. At inference time, all the vectors associated with the desired attribute are extracted and combined with a Dirichlet distribution to produce a latent variable, which is fed to the decoder.

All three methods allow the control of multiple attributes (sentiment and topic) at the same time. The last supports control of the tense of the sentences together with the other attributes.

5.3 Disentanglement

Yu et al. (2021) learn an alignment function to transform the control attribute into an aligned attribute representation. The Bayes rule is used to separate attributes encouraging the alignment function to encode different attributes to different representations. The aligned representation is given to a pre-trained LM (PLM) to steer the generation toward the given control attributes. This method enables control of multiple attributes at the same time (sentiment and topic).

5.4 Modification of Token Distribution

Dathathri et al. (2019) propose a Plug and Play Language Model (PPLM) which uses external attribute classifiers to guide text generation without requiring any training of the PLM. The PLM is used to obtain the next token distribution, which is fed to external classifiers, called Attribute Models, to assess whether the token correctly expresses the desired attributes. The internal latent representations of the LM are updated with a backward pass using the gradients of the attribute models to increase the likelihood of the desired attributes. Finally, the next token distribution is recomputed taking into account the updated latent representations. This model allows control of multiple attributes at a time, such as sentiment and topic.

Inspired by this work, Madotto et al. (2020) propose a variation of PPLMs in which the backward pass is executed n times depending on the desired intensity of the control attribute. Furthermore, they add Residual Adapters (Houlsby et al., 2019) on top of each transformer layer to steer the PLM output distribution without changing its parameters.

Goswamy et al. (2020) propose a different variation of PPLMs based on GPT-2, in which a modified loss is considered to take into account the intensity of the controlled sentiment. Furthermore, instead of considering only positive/negative sentiment, control over 8 emotion categories is enabled.

Starting from PPLMs, Gu et al. (2022a) observe that using a controller alone leads to the trade-off problem, i.e. the controller used to modify the token distribution only focuses on how to make the prefix related to the desired attribute without taking into account the original distribution of the LM. In this way, the controller takes over the LM's control for the next token distribution. In order to alleviate

Model	Control Impl	Attribute Relevance %			Fluency ↓	Diversity ↑			
		Overall	Pos	Pos Prob	Ppl	Dist-1	Dist-2	Dist-3	Avg
Yu et al. (2021)	D	-	-	64.49	36.62	0.48	**0.85**	*0.91*	**0.75**
Qian et al. (2022)	MFT	-	83.3	-	-	-	-	-	-
Gu et al. (2022b)	MFT	**86.7**	-	-	**28.4**	-	-	-	0.49
Dathathri et al. (2019)	MTD	78.8	-	-	46.6	0.36	0.77	*0.91*	0.68
Kumar et al. (2022)	MTD	-	**96**	-	28.9	**0.53**	0.77	0.74	0.68
Gu et al. (2022a)	MTD	-	-	**66.58**	48.52	0.40	0.80	*0.91*	0.70

Table 2: Comparison of techniques evaluated using the PPLM prompts. Different models are used to compute Attribute Relevance and Perplexity, making techniques comparison difficult. MTD=Modification of Token Distribution, MFT=Model Fine-Tuning, D=Disentanglement, Pos=Positive, Pos Prob=Positive probability, Ppl=Perplexity.

this problem, they propose a weighted decoding method that adds a regulator module that permits fine-grained adjustment of a bias signal from the controller. At every step, the regulator detects differences between the PLM distribution and the target attribute and it determines whether to suppress or amplify the bias signal. This method is model agnostic and has been evaluated with sentiment, topic, and toxicity attributes.

The last two methods propose sampling procedures that can be applied to any LM. Landsman et al. (2022) propose to modify beam search by reweighing the token candidate likelihoods to control different attributes. Diverse beam search (Vijayakumar et al., 2016) is used to decode k candidates, which are then scored using an attribute model. The obtained scores are used to reweigh the original likelihoods to produce a reweighed candidate distribution that considers both fluency and attribute characteristics. The resulting distribution is used to sample the next token.

Lastly, Kumar et al. (2022) propose a sampling method combining LM log-likelihoods with arbitrary constraints in a single energy function generating samples in a non-autoregressive manner. The idea is to use a PLM without changing its distribution but sampling from it considering different constraints, i.e. control attributes. The constraints are discriminative classifiers trained from scratch or fine-tuned. This method allows multi-attribute control (sentiment and toxicity).

5.5 Hybrid

Hybrid techniques combine two or more Control Implementation techniques. One possibility is to combine Complete Training and Fine-Tuning, for example, designing a model composed of different modules in which some modules are trained from scratch and some are fine-tuned models. In this context, Tian et al. (2022) propose a conversation model that generates empathetic responses and guides the mood of the conversation in a positive direction while acknowledging the user's emotion. The idea is to extract the sentiment from the conversation context using a fine-tuned sentiment evaluator and use both the context and the extracted sentiment to steer the generation of the next response by generating a responding strategy that will be used by the Conditional Conversation model to generate the final response. The proposed method enables only single-attribute control (of sentiment).

Another way to enable controllability using a hybrid technique is to combine Fine-Tuning and Modification of Token Distribution. Wang et al. (2022) propose a technique to control Story Generation by fine-tuning an encoder that learns the representation of new special tokens identifying the control attributes, thus allowing the model to properly include this information in the generation process. The next token distribution is obtained by combining the decoder distribution and the attention distribution, which allows the model to copy important information from the specified control attributes. The model allows fine-grained control taking into account the characters of the story with their actions and emotions.

In contrast to Wang et al. (2022) who learn the representation of special tokens during fine-tuning, Liu et al. (2021) propose to modify an LM's token distribution including two fine-tuned versions of the PLM: an expert, focused on the desired attribute, and an anti-expert, focused on the opposite of the desired attribute. The next token distribution is obtained by subtracting the anti-expert distribution from the expert one and combining the result with the distribution of the frozen PLM to maintain fluency. This method enables the control only of one control attribute at a time and it has been tested on sentiment and toxicity attributes.

Similarly, Krause et al. (2021) propose to con-

Target Sentiment	Model	Control Impl	Positive AR % ↑			Fluency ↓	Diversity ↑		
			Pos	Neutr	Neg	Ppl	Dist-1	Dist-2	Dist-3
Positive	Landsman et al. (2022)	MTD	-	**98.87**	**74.37**	51.4	*0.56*	**0.84**	**0.85**
	Zhang and Song (2022)	H	-	94.98	64.96	48.71	0.14	0.50	0.76
	Liu et al. (2021)	H	-	94.46	36.42	**45.83**	*0.56*	0.83	0.83
Negative	Landsman et al. (2022)	MTD	**28.42**	**1.99**	-	53.29	0.57	**0.85**	**0.85**
	Zhang and Song (2022)	H	31.24	6.36	-	45.60	0.12	0.48	0.77
	Liu et al. (2021)	H	35.99	3.77	-	45.91	**0.60**	0.84	0.83

Table 3: Comparison of techniques evaluated using the OpenWebText prompts. Different models are used to compute Perplexity, making techniques comparison difficult. AR=Attribute Relevance, Ppl=Perplexity, Pos=Positive prompts, Neutr=Neutral prompts, Neg=Negative prompts, MFD= modification of token Distribution, H=Hybrid.

trast the desired control attribute and its opposite. Instead of fine-tuning specialised LMs for each attribute, GPT-2 is fine-tuned with control codes to obtain a Class-Conditional LM (CCLM). At each time step, the generation is guided by computing classification probabilities for all possible next tokens via the Bayes rule by normalizing two class-conditional distributions: conditioned on the desired attribute and conditioned on the undesired attribute. Like the previous method, it allows the control of one attribute at a time and has been evaluated using sentiment, topic, and toxicity attributes.

Liu et al. (2022) also use a CCLM which is fine-tuned using an external discriminator to generate texts with the desired attributes, supporting multi-attribute control. The token distribution is modified based on a contrastive generator that learns effective representations by bringing together positive samples, i.e. samples with desired attributes, and separating negative samples, i.e. samples without desired attributes. The obtained distribution is combined with the distribution of a PLM to maintain the fluency of the generated text. The generated text is fed to the external discriminator to assess whether it contains the desired attributes or not. The model has been tested on the joint control of sentiment and topic.

Zhang and Song (2022) explore the contrast between desired and undesired attributes proposing a fine-tuned LM incorporating the attribute knowledge of a discriminator, similarly to Liu et al. (2022), to optimize continuous virtual tokens called control-prompts. The learned control-prompts are used as prefixes to steer a fixed conditional LM to generate attribute-specific texts. The LM is fine-tuned using (i) likelihood training, encouraging the LM to generate tokens with higher probability as scored by the discriminator assessing the desired attribute, and (ii) unlikelihood training, keeping the generated tokens away from lower-probability candidates.

6 Comparison of Different Techniques

In this Section, we compare the methods from the last section in terms of performance, efficiency, and generality.

6.1 Performance

In the performance comparison below, three quality criteria from the CTG field are used, namely attribute relevance, fluency, and diversity. **Attribute relevance** (AR) (Yu et al., 2021) assesses the proportion of texts correctly generated with the desired sentiment, i.e. the accuracy of the sentiment attribute measured using an external classifier. Details of the external classifier depends on the evaluation procedure, for more details refer to Appendix A. In some cases, instead of reporting the accuracy, the probability of the text being positive is reported (Pos Prob). **Fluency** is calculated as the perplexity of an external LM (Pichel Campos et al., 2018), while **diversity** is measured as the proportion of unique n-grams obtained using the Distinct metric (Dist-n in Table 2) (Li et al., 2016). Since both AR and fluency are calculated using an external component, it is difficult to obtain a fair comparison due to the usage of different models.

We consider the techniques that have been evaluated using the prompts used in the evaluation of PPLM (Dathathri et al., 2019) and the prompts extracted from OpenWebText (Gokaslan and Cohen, 2019), as detailed below.

In Table 2, we compare 6 methods that have been evaluated using the PPLM prompts, i.e. 15 prefixes used to start text generation. Perplexity is calculated using three different models; as regards attribute relevance, all the methods train or fine-tune a different classifier (for details regarding models see Appendix A.1). Performance results are not directly comparable for these two criteria due to the usage of different models.

Diversity is the only metric for which we can have a fair comparison and here we can see that the technique proposed by Yu et al. (2021) manages to have good variety in the generated texts, while also maintaining good fluency.

Lastly, we compare three techniques evaluated using the OpenWebText (OWT) prompts, composed of neutral, positive, or negative prompts (Table 3). Two different models have been used to compute perplexity, while for the computation of attribute relevance, all the techniques use the same classifier. More details are in Appendix A.2.

In Table 3, we can see that the technique proposed by Landsman et al. (2022) obtains the highest AR and Diversity in both positive and negative target sentiment. This suggests that the proposed technique is able to generate text in the correct sentiment using diverse tokens. The same classifier is used to compute AR for every technique, allowing a fair comparison between them. On the other hand, different LMs are used to calculate Fluency, so it is more difficult to decide whether the differences are due to the model used during evaluation or due to the proposed technique.

6.2 Generality and parameter efficiency

In terms of generality, some of the techniques we have discussed are highly specialised and require many modifications to adapt them to include more or new control attributes. For example, the technique proposed by Xie et al. (2022) is specifically designed to control emotions and needs representing the psychological state of the story's protagonist. Other techniques require the training or fine-tuning of specific models for each control attribute (Liu et al., 2021).

In terms of efficiency, we see some techniques that require the storage and usage of multiple LMs (Liu et al., 2021). On the other hand, many techniques are model agnostic, so they can be applied to any PLM allowing reuse of existing models (Landsman et al., 2022 and Dathathri et al., 2019). In Table 4, we compare the studied techniques in terms of the number of trainable parameters. In Model Agnostic techniques, we consider the number of parameters considering the models used in the reference paper. Unfortunately, it is not possible for all techniques to correctly identify the number of parameters. In general, the modification of token Distribution techniques have fewer trainable parameters than others. The techniques proposed by Gu et al. (2022a) and Landsman et al. (2022) have 0 trainable parameters because they are sampling procedures using a PLM without any training or fine-tuning. More details regarding model parameters in Appendix B.

Model	# trainable parameters
Complete Training	
Qiao et al. (2020)	68M
Betti et al. (2020)	1 generator + 2 discriminators*
Xie et al. (2022)	280M + state trackers, planners*
Model Fine-Tuning	
Qian et al. (2022)	491.520K/attribute
Gu et al. (2022b)	110M
Fang et al. (2022)	117M
Disentanglement	
Yu et al. (2021)	2M
modification of token Distribution	
Dathathri et al. (2019)	∼1K/attribute
Madotto et al. (2020)	5.175M
Goswamy et al. (2020)	∼1K/attribute
Kumar et al. (2022)	774M
Gu et al. (2022a)	0
Landsman et al. (2022)	0
Hybrid	
Wang et al. (2022)	407M
Tian et al. (2022)	337M + Enc + Strategy gen*
Liu et al. (2021)	1548M/attribute
Zhang and Song (2022)	117M
Krause et al. (2021)	345M
Liu et al. (2022)	External Discriminator*

Table 4: Comparison of studied techniques in terms of the number of trainable parameters. More details in Appendix B. *=total number of trainable parameters unclear.

7 Future direction and work

In this section, we summarise the future direction and work described in the analysed papers. Overall, we can identify two suggested directions: model generalisation and fine-grained control.

Model generalisation. Different works suggest to explore the generalisation of the proposed models to explore their capabilities across domains. This can be achieved by introducing the usage of more controlled attributes, such as writer's style and dialog acts (Betti et al., 2020, Yu et al., 2021 and Liu et al., 2022), and the usage of more tasks, such as poetry generation, machine translation, and intelligent education agents (Xie et al., 2022 and Fang et al., 2022).

Fine-grained control. Some works also suggest to explore the capabilities of the proposed methods to support fine-grained control. For example, we can extend the methods to include control attributes in the table-to-text scenario (Zhang and Song, 2022) or explore correlation between different attributes combination to enable fine-grained control (Gu et al., 2022b).

8 Discussion

In this Section, we discuss issues and trends observed in the studied techniques, which suggest possible future directions for the field.

Lack of a standard evaluation procedure. We observe that it is difficult to directly compare the performance of models evaluated on the same dataset using the same metrics, due to the usage of different methods for the metrics' evaluation. In fact, considering the six methods evaluated with PPLM prompts (Section 6.1), we observe that each method has been evaluated using a different classifier to calculate the AR metric. The usage of different classifiers affects the final result of the metric, thus requiring that every work recomputes all the evaluations to have a fair comparison with previous work. Furthermore, we observe that the papers use different datasets for the evaluation, making the comparison between papers even more difficult.

Lack of combination of different control attribute types. We observe that it is mainly topic and toxicity that are explored in combination with sentiment. While topic is a Content Control attribute, other content control attributes, such as data or set of words, are not explored in combination with sentiment, suggesting that a possible future direction is to investigate the combination of such control attributes. Furthermore, we see that Syntactic Control is not explored in combination with sentiment. Moreover, there are not many combinations of different attributes in the *same* category. For example, sentiment and toxicity (Polarity Control) are used together to enable multiple control in just two papers (Qian et al., 2022; Gu et al., 2022b)), but not many other attributes are widely explored.

9 Conclusion

We have reported a systematic survey of Sentiment-Control Text Generation techniques spanning the years 2019–2022. We proposed a categorisation scheme to analyse the studied papers based on the control attributes used and on how the control is implemented. We compared the papers based on their performance, generality and efficiency. While analysing the selected papers, we observed some issues and trends, such as the lack of a standard evaluation procedure and the lack of combinations between different control attribute types.

Acknowledgements

This work was conducted with the financial support of the Science Foundation Ireland Centre for Research Training in Digitally-Enhanced Reality (d-real) under Grant No. 18/CRT/6224 and the Science Foundation Ireland under Grant Agreement No. 13/RC/2106_P2 at the ADAPT SFI Research Centre at Dublin City University. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

References

Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. 2020. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.

Rodrigo Agerri, Montse Cuadros, Sean Gaines, and German Rigau. 2013. Opener: Open polarity enhanced named entity recognition. *Procesamiento del Lenguaje Natural*, (51):215–218.

Federico Betti, Giorgia Ramponi, and Massimo Piccardi. 2020. Controlled text generation with adversarial learning. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 29–34, Dublin, Ireland. Association for Computational Linguistics.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and play language models: A simple approach to controlled text generation. *arXiv preprint arXiv:1912.02164*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Xianghong Fang, Jian Li, Lifeng Shang, Xin Jiang, Qun Liu, and Dit-Yan Yeung. 2022. Controlled text generation using dictionary prior in variational autoencoders. In *Findings of the Association for Computa-*

tional Linguistics: ACL 2022, pages 97–111, Dublin, Ireland. Association for Computational Linguistics.

Xiaocheng Feng, Ming Liu, Jiahao Liu, Bing Qin, Yibo Sun, and Ting Liu. 2018. Topic-to-essay generation with neural networks. In *IJCAI*, pages 4078–4084.

Aaron Gokaslan and Vanya Cohen. 2019. Openwebtext corpus.

Tushar Goswamy, Ishika Singh, Ahsan Barkati, and Ashutosh Modi. 2020. Adapting a language model for controlled affective text generation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2787–2801, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yuxuan Gu, Xiaocheng Feng, Sicheng Ma, Jiaming Wu, Heng Gong, and Bing Qin. 2022a. Improving controllable text generation with position-aware weighted decoding. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3449–3467, Dublin, Ireland. Association for Computational Linguistics.

Yuxuan Gu, Xiaocheng Feng, Sicheng Ma, Lingyuan Zhang, Heng Gong, and Bing Qin. 2022b. A distributional lens for multi-aspect controllable text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1023–1043, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Mark Heitmann, Christian Siebert, Jochen Hartmann, and Christina Schamp. 2020. *More than a feeling: Benchmarks for sentiment analysis accuracy*. Ssrn.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. 2021. GeDi: Generative discriminator guided sequence generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4929–4952, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sachin Kumar, Biswajit Paria, and Yulia Tsvetkov. 2022. Gradient-based constrained sampling from language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2251–2277, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

David Landsman, Jerry Zikun Chen, and Hussain Zaidi. 2022. BeamR: Beam reweighing with attribute discriminators for controllable text generation. In *Findings of the Association for Computational Linguistics: AACL-IJCNLP 2022*, pages 422–437, Online only. Association for Computational Linguistics.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and William B Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. 2021. DExperts: Decoding-time controlled text generation with experts and anti-experts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6691–6706, Online. Association for Computational Linguistics.

Guisheng Liu, Yi Li, Yanqing Guo, Xiangyang Luo, and Bo Wang. 2022. Multi-attribute controlled text generation with contrastive-generator and external-discriminator. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5904–5913, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

Andrea Madotto, Etsuko Ishii, Zhaojiang Lin, Sumanth Dathathri, and Pascale Fung. 2020. Plug-and-play conversational models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2422–2433, Online. Association for Computational Linguistics.

Jose Ramom Pichel Campos, Pablo Gamallo, and Iñaki Alegria. 2018. Measuring language distance among historical varieties using perplexity. application to European Portuguese. In *Proceedings of the Fifth*

Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018), pages 145–155, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Shrimai Prabhumoye, Alan W Black, and Ruslan Salakhutdinov. 2020. Exploring controllable text generation techniques. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1–14, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Jing Qian, Li Dong, Yelong Shen, Furu Wei, and Weizhu Chen. 2022. Controllable natural language generation with contrastive prefixes. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2912–2924, Dublin, Ireland. Association for Computational Linguistics.

Lin Qiao, Jianhao Yan, Fandong Meng, Zhendong Yang, and Jie Zhou. 2020. A sentiment-controllable topic-to-essay generator with topic knowledge graph. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3336–3344, Online. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Hannah Rashkin, Antoine Bosselut, Maarten Sap, Kevin Knight, and Yejin Choi. 2018. Modeling naive psychology of characters in simple commonsense stories. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2289–2299.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Adam Santoro, Ryan Faulkner, David Raposo, Jack Rae, Mike Chrzanowski, Theophane Weber, Daan Wierstra, Oriol Vinyals, Razvan Pascanu, and Timothy Lillicrap. 2018. Relational recurrent neural networks. *Advances in neural information processing systems*, 31.

Lifeng Shang, Zhengdong Lu, and Hang Li. 2015. Neural responding machine for short-text conversation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1577–1586.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Zhiliang Tian, Yinliang Wang, Yiping Song, Chi Zhang, Dongkyu Lee, Yingxiu Zhao, Dongsheng Li, and Nevin L. Zhang. 2022. Empathetic and emotionally positive conversation systems with an emotion-specific query-response memory. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6364–6376, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Olga Uryupina, Barbara Plank, Aliaksei Severyn, Agata Rotondi, and Alessandro Moschitti. 2014. Sentube: A corpus for sentiment analysis on youtube social media. In *LREC*, pages 4244–4249.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2016. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*.

Xinpeng Wang, Han Jiang, Zhihua Wei, and Shanlin Zhou. 2022. CHAE: Fine-grained controllable story generation with characters, actions and emotions. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6426–6435, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Yuqiang Xie, Yue Hu, Yunpeng Li, Guanqun Bi, Luxi Xing, and Wei Peng. 2022. Psychology-guided controllable story generation. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6480–6492, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Erguang Yang, Mingtong Liu, Deyi Xiong, Yujie Zhang, Yao Meng, Changjian Hu, Jinan Xu, and Yufeng Chen. 2021. Syntactically-informed unsupervised paraphrasing with non-parallel data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2594–2604, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Dian Yu, Zhou Yu, and Kenji Sagae. 2021. Attribute alignment: Controlling text generation from pre-trained language models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2251–2268, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Hanqing Zhang and Dawei Song. 2022. DisCup: Discriminator cooperative unlikelihood prompt-tuning for controllable text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3392–3406, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Hanqing Zhang, Haolin Song, Shaoyu Li, Ming Zhou, and Dawei Song. 2022. A survey of controllable text generation using transformer-based pre-trained language models. *arXiv preprint arXiv:2201.05337*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

A Performance Comparison of Different Techniques

In this Section, we provide further details about the automatic evaluation reported in Section 6.1. Furthermore, we report all the models that have been used in the different techniques to calculate the evaluation metrics explained.

A.1 Comparison using PPLM prompts

Perplexity has been calculated using three different models. Dathathri et al. (2019) and Gu et al. (2022a) used GPT (Radford et al., 2018), Qian et al. (2022), Gu et al. (2022b), and Yu et al. (2021) used GPT-2 Large, Kumar et al. (2022) used while GPT-2 XL. Regarding attribute relevance, all the methods trained or fine-tuned a different classifier. Dathathri et al. (2019) trained a classifier on SST-5 (Socher et al., 2013), while Gu et al. (2022a) trained a classifier on IMDb movie reviews (Maas et al., 2011). Qian et al. (2022) and Gu et al. (2022b) fine-tuned RoBERTa (Liu et al., 2019) and DeBERTa (He et al., 2020), respectively, on the Yelp Review dataset (Zhang et al., 2015). Finally, Yu et al. (2021) fine-tune BERT with IMDb movie reviews dataset, while Kumar et al. (2022) fine-tuned SieBERT (Heitmann et al., 2020) on 15 different polarity datasets.

A.2 Comparison using OWT prompts

The techniques in Table 3 have been evaluated using the OpenWebText prompts, which are prompts randomly selected from OpenWebText dataset. For each selected prompt 25 completions are generated using a base LM. Based on the average sentiment of these completions, each prompt is labeled as neutral, positive, or negative resulting in 3 datasets of prompts containing respectively 5K, 2,5K, and 2.5K prompts

In order to compute perplexity, Zhang and Song (2022) used GPT-2 Large, and Landsman et al., 2022 and Liu et al., 2021 used GPT-2 XL. While for the computation of attribute relevance, all the techniques used the same classifier, i.e. a DistilBERT (Sanh et al., 2019) sentiment classifier provided by Huggingface and fine-tuned on SST-2 (Socher et al., 2013).

B Parameters Comparison of Different Techniques

In Table 5, we show the number of parameters in each studied technique. We report all the components of the techniques with the respective parameters (Column 2), the number of trainable parameters (Column 3), and the total number of parameters (Column 4).

In some of the techniques, we can see a huge difference between the number of trainable parameters and the total number of parameters, for example, (Zhang and Song, 2022) and (Landsman et al., 2022).

Model	**Model components**	**# trainable parameters**	**# parameters**
Complete Training			
Qiao et al. (2020)	Encoder (biGRU) + Recognition network (MLP) + Prior network (MLP) + Sentence decoder (GRU) + Discriminator (CNN)	68M	68M
Betti et al. (2020)	Generator (Relational Memory with self-attention) + Syntax Discriminator (Conv net) + Semantic Discriminator (Conv net)	1 generator + 2 discriminators*	1 generator + 2 discriminators*
Xie et al. (2022)	Encoder (BART) 140M + State trackers, planners (BiGRU) + Decoder (BART) 140M	280 + state trackers, planners*	280 + state trackers, planners*
Model Fine-Tuning			
Qian et al. (2022)	PLM (GPT-2 medium) 345M + 491.520K/attribute	491.520K/attribute	345.491M
Gu et al. (2022b)	Encoder (BERT-base-uncased) 110M + Decoder (GPT-2 medium) 345M	110M	455M
Fang et al. (2022)	Encoder (BERT-base-uncased) 110M + Decoder (GPT-2) 117M + Deep Dual function network 1K	117M	227M
Disentanglement			
Yu et al. (2021)	PLM (GPT-2 medium) 345M + Attribute Alignment function (MLP) 2M	2M	347M
modification of token Distribution			
Dathathri et al. (2019)	PLM (GPT-2 medium) 345M + PPLM Discriminator ∼1K/attribute	∼1K/attribute	345M
Madotto et al. (2020)	PLM (DialoGPT medium) 345M parameters + Residual Adapters 5.175M parameters + Discriminator ∼1K/attribute	5.175M	350.175M
Goswamy et al. (2020)	PLM (GPT-2 medium) 345M + PPLM Discriminator 1K/attribute	1K/attribute	345M
Kumar et al. (2022)	PLM (GPT-2 Large) 774M + Discriminative Classifier (GPT-2 Large) 774M	774M	1548M
Gu et al. (2022a)	PPLM 345M + Trainable Regulator (TF-IDF)	0	345M
Landsman et al. (2022)	PLM (DExperts expert) 774M	0	774M
Hybrid			
Wang et al. (2022)	PLM (BART-large-cnn) 407M	407M	407M
Tian et al. (2022)	Encoder + Emotion detector (BERT) 110M + Responding Strategy Generator + GPT 117M + BERT 110M	337M + Encoder + Strategy generator*	337M + Encoder + Strategy generator*
Liu et al. (2021)	PLM (GPT-2 Large) 774M + Expert (GPT-2 Large) 774M and anti-Expert (GPT-2 Large) 774M /attribute	1548M/attribute	1548M/attribute + 774M
Zhang and Song (2022)	CLM (GPT-2 large) 774M + Attribute Discriminator (GPT-2 small) 117M	117M	891M
Krause et al. (2021)	PLM (GPT-2 medium) 345M	345M	345M
Liu et al. (2022)	PLM (GPT-2 medium) 345M + External Discriminator (biGRU)	External Discriminator*	345M + External Discriminator*

Table 5: Comparison of studied techniques in terms of the number of parameters. In Model Agnostic techniques, we consider the number of parameters considering the models used in the reference paper. * the total number of trainable parameters is unclear.

Transformer-based Prediction of Emotional Reactions to Online Social Network Posts

Irene Benedetto[1,2] **Moreno La Quatra**[3]

Luca Cagliero[1] **Luca Vassio**[1] **Martino Trevisan**[4]

[1] Politecnico di Torino, {name.surname}@polito.it
[2] MAIZE SRL, {name.surname}@maize.io
[3] Kore University of Enna, {name.surname}@unikore.it
[4]University of Trieste, {name.surname}@dia.units.it

Abstract

Emotional reactions to Online Social Network posts have recently gained importance in the study of the online ecosystem. Prior to post publication, the number of received reactions can be predicted based on either the textual content of the post or the related metadata. However, existing approaches suffer from both the lack of semantic-aware language understanding models and the limited explainability of the prediction models. To overcome these issues, we present a new transformer-based method to predict the number of emotional reactions of different types to social posts. It leverages the attention mechanism to capture arbitrary semantic textual relations neglected by prior works. Furthermore, it also provides end-users with textual explanations of the predictions. The results achieved on a large collection of Facebook posts confirm the applicability of the presented methodology.

1 Introduction

Most Online Social Network (OSN) platforms allow users to annotate posts with personal reactions. Reactions to social posts not only indicate the user sentiment (e.g., *like, dislike*) but also reflect emotions or feelings (e.g., *sadness, love, care*). Moreover, the quantity of reactions is also a direct measure of the popularity of a post and, indirectly, can indicate the audience's enthusiasm for a specific topic. Such annotations are particularly relevant to marketers, advertisers, and policymakers because they can be exploited to profile OSN users' behaviors and personalize the offer of related services/products. At large, understanding how users react to different types of content on OSNs is of paramount importance to studying human behavior and the online ecosystem.

Predicting the emotional reactions triggered by an OSN post before its publication is particularly appealing as it enables ad hoc content revision and prioritization. The information available ante-publication encompasses the textual content of the post and a set of related metadata (e.g., publisher, publication date, presence of links or associated images). Previous approaches to emotional reaction prediction (Giachanou et al., 2018) rely on traditional occurrence-based text statistics on text, e.g., TF-IDF (Manning et al., 2008). Thus, they ignore the semantics behind the text. Furthermore, the prediction models are used as closed boxes and do not provide any explanations of the predicted reaction.

This paper proposes a new approach to emotional reaction predictions based on Transformers (Vaswani et al., 2017) and Shapley-based explanations. The proposed architecture encodes both the textual content of the post and the related metadata to obtain attention-based predictions of the number of reactions per type. On top of a multi-task regressor, we use an established explainable AI model, namely SHAP (Lundberg and Lee, 2017), to provide end-users with explanations on the most influential textual features.

The main contributions of this paper can be summarized as follows:

- The paper presents a new approach to predict emotional reactions to OSN posts ante-publication, i.e., disregarding post comments or replies.
- It formulates the emotional reaction prediction task as a multi-task regression problem, where the target variables are the number of reactions per type received by the post.
- It proposes a Transformer-based architecture combining both textual content and metadata. The adoption of Transformer-based approaches allows us to capture arbitrary semantic textual relations neglected by prior works.
- It provides end-users with textual explanations

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 354–364
July 14, 2023 ©2023 Association for Computational Linguistics

of the generated predictions based on an established Shapley-based model.

- We extensively validate the proposed approach on a collection of real posts from the Facebook OSN. The results confirm the superior performance of transformer-based models and the prevailing role of textual content compared to metadata.

2 Related work

Despite the availability of various sources of data, accurately predicting content popularity on OSNs is still a challenging task due to their dynamic nature and the presence of various factors. Several techniques and models have been proposed to address this challenge. Typically the forecast objectives are popularity metrics, such as the number of likes a post will get. The features used for the prediction are obtained within the OSN, such as the content of the post, the author of the post, the previous posts, and antecedent metrics on the performance of the post, or are gathered from outside of the OSN, e.g., presence in newspapers and TV shows.

Popularity (likes, views, comments, etc.) is the main sign of success on OSNs, and, as a consequence, it is the focus of most pieces of research. Many works predict content popularity considering content intrinsic characteristics and social interaction features. In particular, Natural Language Processing (NLP) techniques are leveraged for helping predict future popularity. The factors that impact the popularity of posts on Facebook are identified in (Sabate et al., 2014), using an empirical analysis involving multiple linear regressions, with the most important factors being the number of followers and the presence of images. Similarly, (Ferrara et al., 2014) highlights the characteristics related to the dynamics of content consumption in Instagram, while (Gayberi and Oguducu, 2019) and (Carta et al., 2020) predict the popularity of a post by combining user and post features. The authors of (Rizos et al., 2016) predicts different Reddit news popularity indicators using the comment tree and the user graph, while (Li et al., 2013) considers the early views for prediction, focusing on features related to the intrinsic attractiveness of a video and the influence from a propagation structure. Fewer works in the literature deal with emotional reaction prediction. Among them, (Giachanou et al., 2018) simplifies the problem to a classification task. Conversely, (Krebs et al., 2017) predicts Facebook reaction quantity, using NLP on the text of the post and on comments and answers to the post after publication. Notice that content popularity can be also forecast from other information outside of the social network (exogenous), e.g., presence in newspapers, TV shows, etc. An example is the work by (Bertone et al., 2021) that forecasts Instagram and Facebook influencer popularity by extracting external data from Google Trends and then applying financial stock-market tools such as Bollinger Bands.

Another body of literature focuses on the prediction of the temporal dynamics of post popularity. The authors of (Vassio et al., 2021, 2022) study how influencer posts attract likes and reactions and the factors for content popularity evolution. Similarly, (Ahmed et al., 2013) identifies temporal evolution patterns and uses those to predict the future popularity of the content using data from Youtube, Digg and Vimeo, applying K-means clustering and simple linear forecasting technique. Finally, (Ramachandran et al., 2018) propose a model that reproduces the popularity attraction on Twitter, observing that hourly interactions decrease geometrically with time.

Other works focus on predicting other dimensions of popularity, and not the intensity itself. The authors of (Hu et al., 2017) predict popularity of posts by subdividing their evolution into three key moments (“burst”, “peak”, and “fade”) using a Support Vector Regression technique. In a similar direction, (Yu et al., 2020) predicts when the popularity of Twitter hashtags reaches its peak, using an LSTM Deep Learning model with topological network information, social information, and Hashtag strings.

Differently from previous works, in this paper we focus on predicting not only a single metric, but multiple reaction types associated with specific sentiments. We keep the regression objective to make precise and fine-grained predictions, without simplifying it to a classification problem. Moreover, we only use data available at publication time, hence we do not need any other information like early popularity or comment content. Indeed, our approach is only based on the content and metadata of the post itself, and on the features of the creator’s profile. We leverage modern Transformers to capture more subtle, semantic patterns in the text and perform better predictions.

3 Methodology

3.1 Problem formulation

Let u be a user of an Online Social Network (OSN) creating a post p at time t_p. p is characterized by a textual content c_p and a set of metadata (e.g., number of followers of the post creator at time t_p, the post publication date and time, and post author, and the presence of images, videos, or links associated with the post). The other OSN users can annotate p with 9 types of reactions in **R** (i.e., *like, love, wow, care, haha, wow, angry, comment, share*). More details on the annotations and metadata information considered in this study are given in Section 4.1.

Let $r \in \mathbf{R}$ be an arbitrary reaction type. We keep track of the number of reactions of that particular type produced by OSN users over time. Specifically, let $r(p, t)$ be the cumulative function of the number of reactions of type r received by post p at time t.

Given a post p we model the task of emotional reaction prediction as a *multi-task regression problem*, where the targets are the numbers $r(p, t)$ of reactions at time $t > t_p$ for every reaction type $r \in \mathbf{R}$.

Notice that:

- The prediction model *exclusively* considers ante-publication information. Hence, post comments and replies are ignored.
- Whenever not otherwise specified, we set the *prediction horizon* to *one day ahead*, i.e., the time elapsed after the post publication time t_p.
- The *popularity of the post creator* (e.g., the number of friends/followers) likely influences the absolute number of post reactions and it is considered in the post metadata. Furthermore, to properly handle imbalances in the target variable distributions, we address the prediction of the logarithmic function $\tilde{r}_{p,t} = \log_{10} \frac{r_{p,t}}{n_{o_p,t}}$ instead of $r_{p,t}$, where $n_{p,t}$ is the number of followers of the post owner o_p at time t. The use of the logarithm allows us to both aid in achieving training convergence and to compress the variability, while not losing information when the input values are too small.
- The presence of specific terms/expressions in the content of the post may trigger specific reactions (e.g., *Sounds great!* likely triggers a *wow* reaction). We consider this aspect by attending specific pieces of text by means of a transformer architecture (Conneau et al., 2019).

3.2 Regression Model

To address this multi-task regression problem, we propose the architecture depicted in Figure 1, hereafter denoted by Transformer+Metadata. It leverages both text and metadata data to attain precise emotional reaction forecasts. It consists of:

- A Transformer model, namely XLM-RoBERTa (Conneau et al., 2019), to represent each token in the input text.
- A subsequent average pooling operation, to derive the representation of the textual content of the post.
- A fusion stage between the text encoding and the metadata vector[1].
- A fully connected multi-task regression layer with logits to generate the output predictions.

We compare Transformer+Metadata with the following baseline methods:

- Metadata-Only: A classical regression model relying on Linear regression, Adaboost regressor, Random forest regressor, and Multi-Layer perceptron. The models rely only on post metadata, i.e., they disregard the textual content of the post.
- TF-IDF+Metadata: An extended version of Metadata-Only considering also the content of the post encoded using an established occurrence-based text representation, i.e., the term frequency-inverse document frequency (TF-IDF) (Manning et al., 2008). Textual and metadata features are concatenated to feed the regression model. Unlike Transformer+Metadata, TF-IDF+Metadata does not rely on Transformers.
- TF-IDF: a simplified version of TF-IDF+Metadata based on textual features solely.

---

[1] On the training dataset the vector is normalized based on mean and variance.

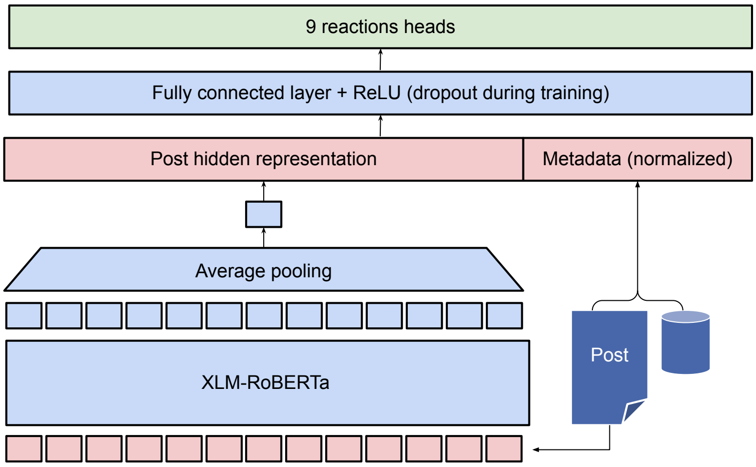

9 reactions heads
Fully connected layer + ReLU (dropout during training)
Post hidden representation
Metadata (normalized)
Average pooling
XLM-RoBERTa
Post

Figure 1: The proposed architecture.

- Transformer-Only: An architecture based on the RoBERTa transformer for text encoding. It ignores all metadata information. It corresponds to the left-hand side of the architecture proposed in Figure 1.
- Moving Average: an auto-regressive model based on moving average proposed by Facebook Inc. This naive approach forecasts a post's reactions using a moving-average over the last 10 posts of the same author. These forecasts are provided directly by CrowdTangle within the posts' metadata.

Our Transformer-based regressors (Transformer+Metadata and Transformer-Only) rely on XLM-RoBERTa (Conneau et al., 2019) for the following reasons:

- It achieved state-of-the-art performance on various natural language processing tasks including sentiment analysis, named entity recognition, and machine translation.
- It has been trained on a large-scale dataset including 100 different languages. Thus, it allows us to apply the proposed approach on multilingual OSN posts (e.g., foreign expressions words included in English posts).

3.3 Model Explanation

To provide explanations of the forecasts, we study the influence of textual features on the predicted reaction values. With the goal of explaining the influence of textual tokens on the prediction model performance, we apply the explainer on top of the Transformer-Only model. Specifically, for each reaction type we highlight the tokens in the input text that mostly influence the prediction of the value of a particular reaction type using the text explainer available in the SHAP library (Lundberg and Lee, 2017). SHAP relies on the concept of Shapley Value, which is established for game theory. It quantifies the value to each player in a cooperative game based on the contribution to the total payoff of the group.

Let $\mathbf{T}_p$ be the set of tokens contained in the content of post p. Given a regression model $\mathcal{R}$ and a target number of reactions $r(p,t)$ for post p, the explainer computes the SHAP Value $\phi(t_p)$ associated with each token $t_k \in \mathbf{T}_p$. SHAP quantifies the influences of token t_k on the $r(p,t)$'s prediction.

4 Experimental setup

4.1 Dataset

To evaluate the proposed techniques, we use a dataset of posts from the Facebook social network. To obtain the posts, we rely on the CrowdTangle platform and its API[2]. CrowdTangle is a content discovery and social analytics tool owned by Meta, which is open to researchers and analysts worldwide upon having a partnership agreement.

We consider posts created by the Facebook profiles of UK newspapers and TV/radio stations of national and local importance. The list of profiles is created and maintained directly by the CrowdTangle team and, in total, includes 1 165 profiles. The list includes the most popular UK broadcast-

---

[2] `https://github.com/CrowdTangle/API` Latest access: April 2023

ers (e.g., the BBC) or newspapers (e.g., the Daily Mail), while others represent local media companies. Notice that the same company can be present with multiple accounts. For instance, the BBC owns many Facebook profiles dedicated to different topics (politics, economics, etc.) and regions (Wales, Scotland, etc.). This study considers posts created over more than three years, from the beginning of 2019 to mid-2022. Over that period, the profiles created a total of $\approx 6\,M$ posts, thus $\approx 40\,k$ per week, on average.

In this work, we consider only influencers whose number of followers is greater than $100\,k$, called Mega and Macro in (Zarei et al., 2020). This decision was based on our observation that influencers with a smaller audience tend to elicit a limited number of reactions. We processed a total of 709,142 posts, which were split into train, test, and development sets in an 80/10/10 ratio.

Each post bears a textual caption with a maximum length of 500 characters and, optionally, can include a picture, a video or a link. In case the post includes a link to an external webpage, CrowdTangle reports the title of the linked webpage as extracted from the HTML and the first sentence of the webpage body – e.g., an article's content. In our experiments, we concatenate this extra text to the post caption for later processing. Moreover, CrowdTangle provides various metadata on posts, such as the creation time and, important to our analysis, a historical view of the reactions received. On Facebook, users can *comment* on a post – i.e., reply with a short text and interact with other commenters. Users also have the possibility to *react* to posts – i.e., expressing their feeling through a predefined set of 7 *emoji*. Namely, they can express *Like*, *Love*, *Care*, *Laughter*, *Surprise*, *Sadness* and *Anger*. The post's metadata include the temporal evolution of the number of comments and reactions (separately by type) received by a post. This information is provided at several time steps, whose granularity becomes coarser as time passes. During the first 24 hours after the post's creation, the metadata indicate the number of comments and reactions every fifteen minutes. After one day from post creation, numbers are provided with a daily granularity. For our analysis, we always consider the number of reactions and comments a post received after 1 day since its creation, i.e., $t = 24$ hours in $r(p, t)$.

Table 1 reports general statistics on the dataset, showing the median, mean and standard deviation of the number of reactions per post. We observe that the prevalent reaction is *Like*, as, in median, posts in our dataset receive 59 *likes*. The distribution presents a heavy tail, as already previously shown in the literature (Vassio et al., 2022), since the mean number of *p*er post is 333 and the standard deviation is rather large (1 768). Other reactions are rather frequent and in the second and third position we find *Comment* and *Share*, respectively, while others are certainly rarer, such as *Care*, *Angry* or *Sad*. Indeed, *Care* is the rarest reaction and is 0 for more than half of posts.

Table 1: Median, mean and standard deviation of target values, for each reaction.

	Median	Mean	Standard Deviation
Angry	1.00	41.04	257.05
Care	0.00	9.95	261.63
Haha	4.00	51.95	414.03
Like	59.00	333.80	1 768.86
Love	2.00	43.94	510.39
Sad	1.00	60.53	809.89
Wow	2.00	20.20	161.46
Comment	46.00	194.45	602.06
Share	13.00	117.56	993.57

4.2 Metrics

Given a post p by an influencer o_p, we have the predicted $\widehat{r}(p, t)$ and actual $r(p, t)$ reactions. To measure the performance of the different prediction models, we adopt the well-known Median Absolute Percentage Error (MedAPE). It is defined as follows:

$$\text{MedAPE} = median \frac{|r(p,t) - \widehat{r}(p,t)|}{r(p,t)} \times 100$$

The MedAPE measures, in percentage, to what extent the prediction deviates from the real value, which in our case is the number of reactions – here, we do not apply any normalization. Notice that the model's output and target values are still the logarithms of the reactions per follower and expected reactions per follower to reduce skewness. However, we opt to measure the performance on the original reaction number, as its prediction represents the objective of our work.

To understand how Transformer-based models operate, we leverage Shapley values. They represent the contribution of each token to the predicted value. The $\text{mean}(|\text{SHAPvalue}|)$ for a specific regression and reaction is defined as:

$$\mathrm{mean}(|\mathrm{SHAPvalue}|) = \frac{1}{N}\sum_{p=1}^{N}\phi_{i,p,t}$$

where $\phi_{i,p,t}$ is the function that computes the Shapley value of token i in post p for reaction t. We sum up the values for each token across all posts and divide them by the total number of posts to obtain the average token importance. The $\mathrm{mean}(|\mathrm{SHAPvalue}|)$ provides insight into the degree of importance of each token in the prediction.

4.3 Model evaluation and hyper-parameter setting

To evaluate Metadata-Only, TF-IDF+Metadata, and TF-IDF we perform hyper-parameters tuning using a 5-fold cross validation. To evaluate Transformer+Metadata and Transformer-Only we train the `xlm-roberta-base`[3] model version using a batch size of 16 and a learning rate of $5 \cdot 10^{-5}$. Models undergo a maximum of 5 epochs of training, with an early stopping criterion applied. Both models also have a weight decay of 0.01 and a warmup ratio of 0.06.

4.4 Hardware

Experiments were run on a machine equipped with Intel® Core™ i9-10980XE CPU, 2 × Nvidia® RTX A6000 GPU, 128 GB of RAM running Ubuntu 22.04 LTS. We provide detailed information about the models used for the evaluation and the fine-tuning procedure in the official project repository[4].

5 Results

In this section, we show and discuss the results of our experiments. We evaluate the proposed regression models on the dataset described in Section 4.1 in terms of Median Absolute Percentage Errors. Then, we delve into the Transformer+Metadata model and investigate the most relevant words to the regression task using the SHAP algorithm.

5.1 Analysis of Regression Performance

In this section, we show and discuss the performance of the different regression models. In Table 2, we report Median Absolute Percentage Error for all models, separately by reaction type. The table allows us to compare Transformer+Metadata with the other simpler models and quantify the benefits of this approach. For the sake of brevity, for TF-IDF+Metadata, Metadata-Only, and TF-IDF we report only the outcomes of the model that performs best on the validation set (i.e., MLP regressor).[5] Each cell in the Table indicates the MedAPE for a given combination of model and reaction. Notice that the value is reported in percentage. Since the reported error is relative, the number in a cell represents the percentage error with respect to the target magnitude.

Watching Table 2, we first observe the problem we address challenges all the proposed models, as in all cases the MedAPE is above 50%. *Care* is an exception, as the MedAPE assumes low values for Transformer-based models. This is expected, as this reaction often assumes value 0 as reported in Table 1. As such, the models tend to output 0 as the predicted value, leading to a correct prediction in most cases. In many cases, the simplest models provide very erroneous predictions, which we report in Table 2 with $> 100\%$ to indicate that the prediction model was not able to provide any kind of meaningful output.

Comparing the different regression models, we note that Transformer-based ones perform best thanks to the higher capability to capture semantic information from text. Overall, Transformer-Only and Transformer+Metadata have the best prediction accuracy for most reaction types. If neglect *Care* (which is most of the times 0), the best performance is achieved with *Wow*, where Transformer+Metadata achieves a MedAPE of 45.26%. Transformer+Metadata provides the best performance for all reactions except for *Share*, for which Transformer-Only performs best, with MedAPE of 66.55%. Overall, we observe that the impact of metadata in transformer-based models is relatively limited. Indeed, the improvements of Transformer+Metadata with respect to Transformer-Only are in most cases less than a percentage point. In this direction, notice the asterisks (*) in the table, that indicate whether the best-performing model offers a statistically significant improvement with respect to the given cell. The results show that Trasformer-Based models improve *significantly*

---

[3] `https://huggingface.co/xlm-roberta-base` Latest access: April 2023

[4] `https://anonymous.4open.science/r/social-reactions-predictions-7CB0/README.md`

[5] We selected the MLP regression after performing grid search on validation data with several configuration parameters that include support vector machines, *AdaBoost* regressor, multilayer perceptron regressor, and linear regression

Table 2: Comparison between regression performance on test data for each reaction type, in terms of MedAPE. The asterisk (*) indicates that the best-performing model (reported in bold) offers a statistically significant improvement (t-test with p-value=0.05) with respect to the given cell.

Model	Angry	Care	Comment	Haha	Like	Love	Sad	Share	Wow
Moving Average	>100% *	>100% *	97.20% *	>100% *	>100% *	>100% *	>100% *	86.20% *	>100% *
Metadata-Only	>100% *	>100% *	>100% *	99.98% *	98.03% *	97.12% *	>100% *	99.99% *	>100% *
TF-IDF	78.52% *	30.19% *	95.21% *	97.14% *	>100% *	89.66% *	>100% *	90.24% *	82.10% *
TF-IDF+Metadata	69.45% *	37.98% *	91.21% *	76.95% *	76.38% *	74.35% *	67.27% *	79.62% *	72.14% *
Transformer-Only	56.47%	0.10%	78.30%	62.42%	51.69% *	63.44% *	66.86% *	**66.55%**	50.22% *
Transformer+Metadata	**56.30%**	**0.05%**	**78.15%**	**62.47%**	**50.67%**	**62.07%**	**61.87%**	66.56%	**45.26%**

prediction performance, but in four cases Transformer+Metadata and Transformer-Only performance is not statistically different. Conversely, for traditional machine learning models, jointly considering text and metadata information significantly improves the performance – compare for example TF-IDF+Metadata with TF-IDF. This is likely due to the more limited informative content of the occurrence-based text representation.

5.2 Examples of Model Explanation

We now focus on Transformer-Only and consider the trained model that we obtain. The model is based on a Transformer architecture that bases its predictions on the post text. As such, we now investigate how Transformer-Only operates in making its decision and try to *explain* its choices with the help of the SHAP technique and library. The plots in Figure 2 respectively show the most influential tokens (at most 20) for three reactions, namely *Angry*, *Love*, and *Care*, as obtained using SHAP. Here, our main goal is to gain insights into the behavior of a non-interpretable model, such as the Transformer-Only, and identify the underlying factors driving OSN users' reactions. For each textual token, the plots report the $\mathrm{mean}(|\mathrm{SHAPvalue}|)$ (Lundberg and Lee, 2017) computed over a random sample of $N = 100$ posts in the test set. This value indicates the average feature influence on the output predictions. More specifically, the Shapley values are computed using the *auto* configuration, which automatically recommends the best Explainer for our model.

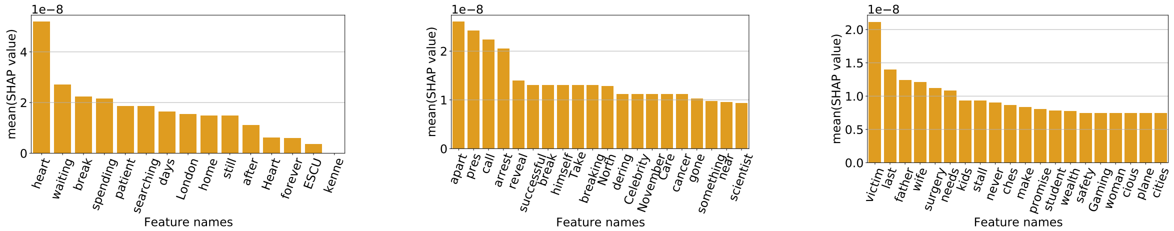
The Explainer is capable of detecting particular words and phrases having a significant impact on the types of reactions a post receives. This information could be useful for individuals and organizations who are looking to maximize engagement with their social media content. By incorporating more positive and uplifting language into their posts, authors may be able to increase the number of positive reactions (*Love* and *Ahah*) they receive, while reducing the number of negative ones (*Sad* and *Angry*).

Based on the SHAP values, we can gain insights into the types of language and content that are more likely to elicit certain reactions from OSNs users. The words and phrases associated with the *Care* reaction (i.e., Figure 2a) suggest that posts that convey a sense of patience and perseverance may be more likely to receive caring reactions. Similarly, the words and phrases associated with the *love* reaction (i.e., Figure 2b) suggest that posts that contain positive news, success stories, and celebrity updates may be more likely to receive love reactions. On the other hand, the words and phrases associated with the *angry* reaction (i.e., Figure 2c) suggest that posts that contain words related to victimhood, surgery, and family issues may be more likely to receive angry reactions. A more detailed analysis of the explanations can be found in Appendix A. It is important to note that these insights are based on the specific data and model used for the analysis, and may not necessarily apply to all contexts and users.

6 Limitations

Our methodology provides a novel approach to predicting emotional reactions to social posts. While our proposed methodology has shown promising results, further research is necessary to address some limitations and expand the scope of our findings.

Firstly, our model's reliance on only textual content and metadata may limit prediction accuracy, as other factors such as user demographics or multimedia content may also impact emotional reactions. Future work could explore the inclusion of additional features. Secondly, this study focuses on predicting emotional reactions to posts, rather than the reasons behind them. While our method provides textual explanations, they may not fully capture the complexity of user emotional responses.



(a) *Care* (b) *Love* (c) *Angry*

Figure 2: The feature importance of strongly correlated tokens computed as the mean of the absolute value of Shapley values for each reaction.

Lastly, our approach was evaluated on Meta posts, but may not generalize well to other social networks or domains. Further evaluations on different datasets are needed to confirm the applicability of our methodology across different contexts.

7 Ethical considerations

While our study minimizes privacy concerns by solely analyzing creator post content, we acknowledge the ethical implications of our model's predictions on individuals and communities.

Overall, the models proposed in the present study raise ethical concerns around the potential impact of our model's predictions on content creator behavior and the distortion of social media sentiment. Our model's predictions may incentivize content creators to prioritize emotional reactions over accuracy or factual content. To address this concern, responsible and ethical use of our methodology, alongside recognition of the limitations in capturing the complexity of user behavior, is necessary.

Finally, we observe that the dataset used in this work has been collected in a lawful and ethical environment. We used the CrowdTangle platform offered by Facebook, which contains only public posts of public profiles. The metadata of posts do not contain any sensitive information and only include the aggregate number of reactions they received.

8 Conclusions and Future Work

The paper explored the use of transformer-based models to predict the number of reactions ante-publication to a social post. It focused on predicting the reaction counts for different types of positive and negative reactions by exploring the role of text and metadata information. Compared to prior approaches based on traditional text representations (e.g., `TF-IDF`), it achieved significant performance improvements thanks to the higher capability to capture the semantics behind the text. Transformers perform best compared to simpler methods even in the absence of metadata. The paper also leveraged a Shapley-based explainer to identify the tokens that mostly influence the prediction outcomes. The explanations meet the expectation, especially for very positive and very negative reaction types.

Acknowledgements

References

Mohamed Ahmed, Stella Spagna, Felipe Huici, and Saverio Niccolini. 2013. A peek into the future: Predicting the evolution of popularity in user generated content. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, page 607–616, New York, NY, USA. Association for Computing Machinery.

Fabio Bertone, Luca Vassio, and Martino Trevisan. 2021. The stock exchange of influencers: A financial approach for studying fanbase variation trends. In *Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '21, page 431–435, New York, NY, USA. Association for Computing Machinery.

Salvatore Carta, Alessandro Sebastian Podda, Diego Reforgiato Recupero, Roberto Saia, and Giovanni Usai. 2020. Popularity Prediction of Instagram Posts. *Information*, 11(9).

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Emilio Ferrara, Roberto Interdonato, and Andrea Tagarelli. 2014. Online Popularity and Topical Interests through the Lens of Instagram. In *Proceedings* *of the 25th ACM Conference on Hypertext and Social Media*, page 24–34.

Mehmetcan Gayberi and Sule Gunduz Oguducu. 2019. Popularity Prediction of Posts in Social Networks Based on User, Post and Image Features. In *Proceedings of the 11th ACM MEDES*, page 9–15.

Anastasia Giachanou, Paolo Rosso, Ida Mele, and Fabio Crestani. 2018. Emotional influence prediction of news posts. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 12.

Ying Hu, Changjun Hu, Shushen Fu, Mingzhe Fang, and Wenwen Xu. 2017. Predicting key events in the popularity evolution of online information. *PLOS ONE*, 12(1):1–21.

Florian Krebs, Bruno Lubascher, Tobias Moers, Pieter Schaap, and Gerasimos Spanakis. 2017. Social emotion mining techniques for facebook posts reaction prediction.

Haitao Li, Xiaoqiang Ma, Feng Wang, Jiangchuan Liu, and Ke Xu. 2013. On Popularity Prediction of Videos Shared in Online Social Networks. In *Proceedings of the 22nd ACM CIKM*, page 169–178.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK.

Arthi Ramachandran, Lucy Wang, and Augustin Chaintreau. 2018. Dynamics and prediction of clicks on news from twitter. In *Proceedings of the 29th on Hypertext and Social Media*, HT '18, page 210–214, New York, NY, USA. Association for Computing Machinery.

Georgios Rizos, Symeon Papadopoulos, and Yiannis Kompatsiaris. 2016. Predicting News Popularity by Mining Online Discussions. In *Proceedings of the 25th International Conference on World Wide Web (WWW)*, page 737–742.

Ferran Sabate, Jasmina Berbegal-Mirabent, Antonio Cañabate, and Philipp R. Lebherz. 2014. Factors influencing popularity of branded content in Facebook fan pages. *European Management Journal*, 32(6):1001–1011.

Luca Vassio, Michele Garetto, Carla Chiasserini, and Emilio Leonardi. 2021. Temporal dynamics of posts and user engagement of influencers on facebook and instagram. In *Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '21, page 129–133, New York, NY, USA. Association for Computing Machinery.

Luca Vassio, Michele Garetto, Emilio Leonardi, and Carla Fabiana Chiasserini. 2022. Mining and modelling temporal dynamics of followers' engagement on online social networks. *Social Network Analysis and Mining*, 31:012012.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Hai Yu, Ying Hu, and Peng Shi. 2020. A prediction method of peak time popularity based on twitter hashtags. *IEEE Access*, 8:61453–61461.

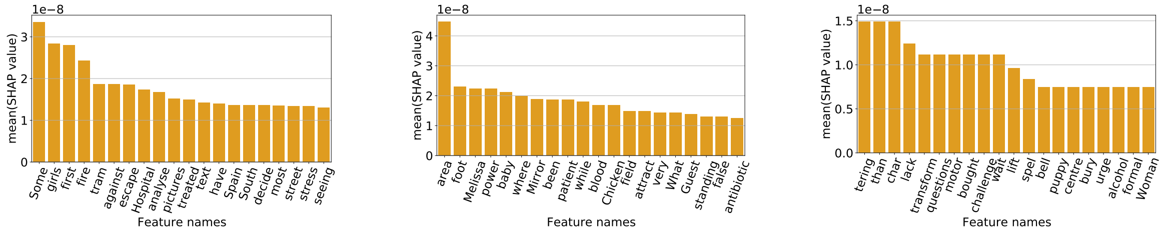
Koosha Zarei, Damilola Ibosiola, Reza Farahbakhsh, Zafar Gilani, Kiran Garimella, Noël Crespi, and Gareth Tyson. 2020. Characterising and detecting sponsored influencer posts on instagram. In *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 327–331.

A Detailed discussion of reaction explainations

In this section, we provide a more detailed explanation of the SHAP values associated with each type of reaction in the proposed Transformer-Only model. We discuss the tokens that are found to be strongly correlated with each reaction prediction and provide some insights into the types of content that may elicit these reactions from social media users. Figure 3 shows a detailed view of the explanation for the reactions that were not discussed in the main body of the paper. The histograms for the other reactions are reported in Figure 2. It is important to note that the $\mathrm{mean}(|\mathrm{SHAPvalue}|)$ presented in this section are not intended to identify the specific words or phrases that elicit each type of reaction. Rather, they are intended to provide insights into the tokens that are strongly correlated with the estimation of each reaction in our regression model.

Care

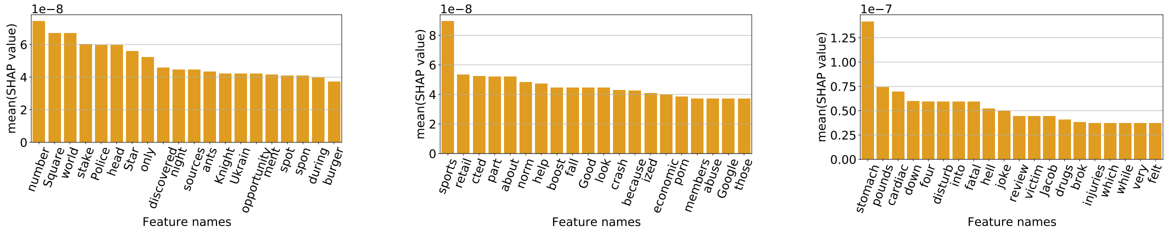
The *care* reaction (Figure 2a) is often associated with words and phrases that convey a sense of empathy and concern for others. Based on the scores provided by the explainability, the words that are most strongly associated with the care reaction include "heart", "waiting", "break", and "spending". These words suggest that users may react with *care* to posts that relate to personal struggles, offer support, or highlight the importance of patience and



(a) *Sad*

(b) *Wow*

(c) *Haha*



(d) *Like*

(e) *Comment*

(f) *Share*

Figure 3: $\mathrm{mean}(|\mathrm{SHAPvalue}|)$ values for most relevant tokens pertaining to different reaction types.

resilience. For example, posts that discuss a difficult experience or offer words of encouragement to those going through a tough time may be more likely to arouse this reaction.

Love

The frequency distribution of token relevance for the *love* reaction is illustrated in Figure 2b. The tokens that are strongly associated with positive emotions and affection tend to receive more *love* reactions from users.

The explainability model has identified certain tokens such as “successful”, “care”, “cancer”, and “scientist” that are useful to estimate the *love* reaction. This may suggest that users tend to react with love towards posts that feature success, care for others, good health, or scientific accomplishments.

Angry

The *angry* reaction (Figure 2c) is often characterized by the use of language that conveys frustration or anger. The explainability models indicate that the tokens most strongly associated with the angry reaction include “victim”, “last”, “father”, and “wife”. OSN users are more likely to react with anger to posts that pertain to unfair treatment or injustice, such as the victimization of vulnerable groups or the loss of a loved one.

Moreover, the words ”surgery”, ”needs”, and £kids” indicate that users may be more prone to reacting with anger to posts that deal with issues of health or well-being, such as inadequate access to medical care or basic necessities.

Sad

The *sad* reaction (shown in Figure 3a) is typically associated with phrases and words that convey negative emotions like grief, disappointment, or sadness. The explainability model’s relevance scores indicate that tokens such as ”fire”, ”hospital”, ”street”, and ”stress” are most strongly connected to the *sad* reaction. This suggests that posts related to traumatic events, medical emergencies, stress, and challenging experiences are more likely to elicit this reaction from users.

Although the words ”girls”, ”tram”, and ”escape” don’t seem directly linked to the *sad* reaction, they may also suggest that users react with sadness to posts related to personal struggles. For instance, stories of oppression or discrimination may evoke feelings of sadness in users.

Wow

The *Wow* reaction (Figure 3b) is triggered by content that suggests surprise or admiration. Tokens like ”power” and ”baby” indicate that users are more likely to react with amazement to displays of strength or cute and impressive babies, and stories that are out of the ordinary. Additionally, tokens like ”area”, ”foot”, and ”field” suggest that users may react with surprise to posts that relate to natural phenomena or impressive athletic feats.

Haha

The *haha* reaction (Figure 3c) is typically used to express surprise or amusement in response to certain words or phrases. Although some associated terms may be coincidental, other tokens show a strong semantic connection to the reaction. For example, words like ”transform”, ”motor”, ”challenge”, and ”puppy” appear to be highly correlated with the *haha* reaction. Such words suggest that users tend to react with humor or surprise to content that involves unexpected or funny instances related to technology, witty takes on everyday experiences or amusing content involving animals.

Like, Comment, and Share

The social media actions of *Like*, *Comment*, and *Share* are prevalent across various platforms. However, according to explainability models, the tokens linked with each of these actions do not necessarily indicate particular content that would elicit them (see Figures 3d, 3e, 3f).

For example, the tokens relevant to the *Like* reaction include ”number”, ”Square”, ”world”, and ”stake”, which do not have an apparent semantic connection with the positive emotions commonly associated with the *Like* reaction. Similarly, the tokens linked with *Comment* and *Share*, such as ”sports”, ”retail”, ”stomach”, and ”pounds”, do not necessarily suggest specific content that would encourage users to engage with a post by commenting or sharing.

Transfer Learning for Code-Mixed Data: Do Pretraining Languages Matter?

Kushal Tatariya[1] **Heather Lent**[2] **Miryam de Lhoneux**[1]
[1] Department of Computer Science, KU Leuven, Belgium
[2] Department of Computer Science, Aalborg University, Denmark
{kushaljayesh.tatariya, miryam.delhoneux}@kuleuven.be
hcle@cs.aau.dk

Abstract

Monolinguals make up a minority of the world's speakers, and yet most language technologies lag behind in handling linguistic behaviours produced by bilingual and multilingual speakers. A commonly observed phenomenon in such communities is code-mixing, which is prevalent on social media, and thus requires attention in NLP research. In this work, we look into the ability of pretrained language models to handle code-mixed data, with a focus on the impact of languages present in pretraining on the downstream performance of the model as measured on the task of sentiment analysis. Ultimately, we find that the pretraining language has little effect on performance when the model sees code-mixed data during downstream finetuning. We also evaluate the models on code-mixed data in a zero-shot setting, after task-specific finetuning on a monolingual dataset. We find that this brings out differences in model performance that can be attributed to the pretraining languages. We present a thorough analysis of these findings that also looks at model performance based on the composition of participating languages in the code-mixed datasets.

1 Introduction

In multilingual societies, contact between multiple languages has resulted in a plethora of linguistic phenomena that have long been the subject of study in linguistics, and more recently in NLP. One such phenomenon is code-switching, or code-mixing[1], in which speakers use material from two or more languages within the same conversation (Thomason, 2001).

Code-mixing typically occurs in informal registers and casual conversations, permitted or constrained by different sociolinguistic factors (Doğruöz et al., 2021). The typical lack of formality surrounding the use of code-mixing contributes to difficulties in data collection, as code-mixing is less likely to occur in official documents by governments and organizations, which have been reliable resources for the creation of many datasets (Sitaram et al., 2019). In contrast, social media has been a particularly fruitful domain for sourcing code-mixed data, useful in a wide variety of downstream tasks (Barman et al., 2014; Banerjee et al., 2016; Chakma and Das, 2016; Vijay et al., 2018; Patra et al., 2018a; Bohra et al., 2018). Among these tasks, sentiment analysis and offensive language detection stand out in particular, as Agarwal et al. (2017) have demonstrated that multilingual speakers are likely to utilize code-mixing to express their emotions, especially when cursing. Thus, improving methodologies for working with intricate code-mixed data is highly relevant to the study of sentiment analysis, and social media at large.

The advent of pretrained language models (PLMs) has tangibly shaped the norms for working with most languages, yet the implications for code-mixed data are much less clear. PLMs have so far largely operated under monolingual assumptions and biases (Ramesh et al., 2023; Talat et al., 2022). Most PLMs, including the massively multilingual ones, are trained on large web corpora, and studies have shown that the quality filters and data selection methodologies for these data sources tend to exclude text with dialectal nuances, such as text with non-standard varieties of English like African American English, or Hispanic-aligned English. (Dodge et al., 2021; Gururangan et al., 2022). Attempts have been made at language modelling for code-mixed data (Gupta, 2019; Nayak and Joshi, 2022), but an interesting question remains about how much the languages used in the pretraining of PLMs interact with each other to impact their performance on code-mixed data. A better understanding of this would enable targeted resource allocation to code-mixed NLP, and also potentially help understand how PLMs process language. PLMs

---

[1] Although distinctions between the two terms are made, we use them interchangeably.

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 365–378
July 14, 2023 ©2023 Association for Computational Linguistics

that have been pretrained on many high- and low-resource languages are now widely available and accessible, which provides a fertile ground for such analyses (Wolf et al., 2020). To shape the focus of this study, we introduce our hypothesis below.

Hypothesis: *PLMs trained exclusively on data from relevant languages would demonstrate better performance than those that contain other extraneous languages and/or are only trained on one language.*

At the same time, the "curse of multilinguality", coined by Conneau et al. (2019), refers to the trade-off between adding more languages to increase cross-lingual capabilities, and the consequences of adding too many which can ultimately lead to loss of performance across the board in monolingual and cross-lingual benchmarks. Massively multilingual models can be susceptible to this, and therefore we presume that models trained on data from relevant language families would be at an advantage. To this end, we test the performance of 7 pretrained language models on the task of sentiment analysis for different code-mixed datasets, which cover 6 languages.

2 Background

2.1 Code-Mixed NLP

In recent years, research in code-mixed NLP has steadily increased, resulting in the release of benchmark datasets like GLUE-CoS (Khanuja et al., 2020) and LinCE (Aguilar et al., 2020), organized shared tasks (Aguilar et al., 2018; Solorio et al., 2020, 2021), and several survey papers (Sitaram et al., 2019; Doğruöz et al., 2021; Winata et al., 2022). Although most code-mixing datasets include at least one high-resource language like English, progress in code-mixed NLP still lags behind as there exist additional challenges not present within the scope of monolingual work. Firstly, detecting or predicting when and where code-mixing will occur is non-trivial for a wide variety of linguistic reasons (Doğruöz et al., 2021). Most language identification approaches operate on the document or sentence level, rather than token level, and thus do not perform well for code-mixed data (Caswell et al., 2020). Moreover, some code-mixed data includes the use of multiple scripts, which can further complicate matters. Therefore, it is not surprising that, as Khanuja et al. (2020) found with mBERT, performance over code-mixed data is typically worse than monolingual counterparts, calling for further studies on the capabilities of PLMs on code-mixed data.

Studies in code-mixed sentiment analysis have demonstrated the strong relationship between a speaker's language choice and the sentiment they wish to convey. For example, Rudra et al. (2016) found that bilingual Hindi-English speakers preferred to express negative sentiments in Hindi. Similarly, Ndubuisi-Obi et al. (2019) found that Naija was used for expressing any kind of sentiment (i.e. high-emotion settings), in lieu of English for matter-of-fact statements. While this makes code-mixing relevant to studies in sentiment analysis, Zaharia et al. (2020) have noted that current methods in this space cannot cope when two languages come together to express one sentiment. Thus, improved methods for code-mixed NLP are also important for sentiment analysis in general, in a world where most people are bilingual or multilingual.

2.2 Transfer Learning

Transfer learning is the capacity of a model to take knowledge acquired from one language or domain and effectively apply it towards another. Thus, without enough data to create PLMs tailored to code-mixed language, transfer learning will undoubtedly play an important role in processing code-mixed text. PLMs have shown promising transfer learning abilities across languages that are similar (Pires et al., 2019; Lin et al., 2019; de Vries et al., 2022). Pires et al. (2019) demonstrated that successful cross-lingual transfer can lead to multilingual representations that are able to incorporate information from multiple languages, and even generalise across observed scripts, ultimately leading to increased performance on code-mixed data. PLMs have also been proven to have zero-shot transfer capabilities (Wu and Dredze, 2020), which can then be further enhanced by fine-tuning on limited instances from the target languages (Lauscher et al., 2020; de Vries et al., 2021). However, other work has shown that transfer learning is not always trivial. In the context of Creole NLP, Lent et al. (2022) found that even pretraining on languages with direct genealogical ties to the target Creoles failed to result in useful PLMs for those languages. Thus, further investigation of the mechanisms of pretraining data on the performance of PLMs is required.

3 Languages and Datasets

The datasets used in this study are mainly comprised of text scraped from Twitter, Facebook and YouTube. Details are summarised in Table 1. All datasets from this work can be found in our github repository[2].

Dataset	**Language**	**Train / Dev**
AfriSenti	pcm	5.1K / 1.2K
NaijaVader	pcm	9.8K / 1.4K
SAIL	hin-eng	10K / 1.2K
IIITH-CodeMix	hin-eng	2.7K / 388
TamilMixSentiment	tam-eng	110K / 1.2
MalayalamMixSentiment	mal-eng	4.2K / 480
DravidianCodeMix	tam-eng	33K / 4.2K
DravidianCodeMix	mal-eng	14K / 1.8K
DravidianCodeMix	kan-eng	5.2K / 656

Table 1: Details about the datasets in the study. The first four datasets have 3 labels - 'positive', 'negative' and 'neutral', and the latter five datasets have 4 labels - 'positive', 'negative', 'mixed_feelings' and 'unknown_state'.

3.1 Code-Switching in India

With the multitude of languages being spoken in India, and the plethora of bilingual and multilingual speakers, code-switching is a commonly observed phenomenon (Barnali, 2017). With the dominance of English in Indian society, educational institutions and official communications, there are millions of English speakers in India who can also be fluent in at least one other native Indian language. Thus, speakers can frequently switch between English and their other native language for ease of communication. Very commonly observed is Hindi-English code-switching, more popularly known as Hinglish, which refers to mixing of Hindi and English lexicon, phrases and syntax. In the written form, it is normally seen in Latin script. This paper looks at Hinglish, along with the mixing of English with Dravidian languages like Malayalam, Tamil and Kannada.

Hinglish Data For Hinglish we use the datasets curated by Joshi et al. (2016) (hereafter referred to as IIITH-CodeMix) and Patra et al. (2018b) (hereafter referred to as SAIL). The IIITH-CodeMix dataset consists of user comments from popular Indian Facebook pages, with comments not written in the Roman script, or comments completely in English being removed. The SAIL dataset, included in the GLUECoS benchmark, on the other hand, is Twitter data, again with only romanized instances of Hindi.

Dravidian Data For south Indian languages in the Dravidian language family, we use 5 datasets in 3 languages - Tamil, Malayalam and Kannada. The dataset introduced in Chakravarthi et al. (2020b) is referred to as TamilMixSentiment, with Tamil-English data, and (Chakravarthi et al., 2020a) is called MalayalamMixSentiment, containing Malayalam-English data. The remaining 3, in Tamil, Malayalam and Kannada, come from Chakravarthi et al. (2021), following a similar annotation scheme as the previous ones, hereafter referred to as DravidianCodeMix. All five datasets have been created from scraping YouTube comments.

The Dravidian datasets, unlike the others, contain text that is not in the Latin script. For this study, however, we transliterated all the non-Latin characters into Latin script to make fair comparisons between monolingual models that have not been trained on non-Latin script and the multilingual ones that have. Moreover, Moosa et al. (2023) found that transliteration helps improve multilingual model performance and cross-lingual representations. We used the transliteration library for Indic languages created by Madhani et al. (2022), trained on the Aksharantar dataset. Additionally, the original datasets contain 5 labels - 'positive', 'negative', 'mixed_feelings', 'unknown_state' and 'not_target_language'. All examples labeled 'not_target_language' were removed from the datasets since they contained non-Indic scripts that the transliteration model has not seen, and language identification falls outside the scope of this study.

3.2 Code-Switching in Nigeria

Nigerian Pidgin, commonly referred to as Naija, is the unofficial lingua franca in Nigeria (Ekundayo, 2022). It is an English-based Creole, which arose from language contact between English and local Nigerian languages such as Hausa, Yoruba, Igbo, and others. Despite the significant influence of English on the language, Naija is a fully independent language of its own, with aspects of morphology, syntax, and semantics that are detached from English (Agbo and Plag, 2020; Odiegwu, 2022). Code-mixing in Nigeria can often occur between English, Naija, and a given speaker's mother

---

[2] `https://github.com/kushaltatariya/Sentiment-Analysis-for-Code-Mixed-Data`

		Is It Present?								
		Monolingual		Multilingual		Indic		African		Code-mixed
Language	**Script**	BERT	RoBERTa	mBERT	XLM-R	IndicBERT	MuRIL	AfriBERTa	AfroXLMR	HingMBERT
English (eng)	Latin	✓	✓	✓	✓	✓	✓		✓	✓
Naija (pcm)	Latin							✓	✓	
Hinglish	Latin									✓
Hindi (hin)	Latin				✓		✓			
	Devanagari			✓	✓	✓	✓			✓
Malayalam (mal)	Latin						✓			
	Malayalam			✓	✓	✓	✓			✓
Tamil (tam)	Latin				✓		✓			
	Tamil			✓	✓	✓	✓			✓
Kannada (kan)	Latin						✓			
	Kannada			✓	✓	✓	✓			✓

Table 2: Languages present in the pretraining of each PLM.

tongue (Mensah and Ndimele, 2014; Akande and Salami, 2021; Sarah and Oladayo, 2021). However, the boundaries between Naija and code-mixing may not always be straightforward to diagnose, as Naija is amenable to immense variation from one speaker to another (Lent et al., 2022). While most datasets in Naija are not designed for studying code-mixing (with the exception of Ndubuisi-Obi et al. (2019)), we surmise that some code-mixing may be present in Naija text, as a result of Naija's flexibility for speakers' individual linguistic backgrounds. Therefore, we include Naija in our analysis to gain a perspective on how language models perform on code-mixing within a language in its own right. This choice is also in line with previous work, which acknowledges the propensity for code-mixing in Naija and other African Creoles (Adebara et al., 2022).

Naija Data We use two datasets for Naija. The first one was introduced by Oyewusi et al. (2020) (hereafter referred to as NaijaVader) within the VADER Sentiment Analysis framework (Hutto and Gilbert, 2014), containing tweets. The authors did not release official splits of the data, so we created our own train-dev-test splits. The second dataset (hereafter referred to as AfriSenti), is part of Muhammad et al. (2023), a Twitter sentiment analysis benchmark for African languages. They used a location and vocabulary based setup to collect tweets in each respective language.

4 Models

The PLMs compared in this study can be classified into four categories based on their pretraining data: monolingual, multilingual, Indic and African, presented in Table 2. We used the base version of each model for our experiments, without performing any additional pretraining.

Monolingual Models For this study, we focus mainly on standard English monolingual PLMs, namely BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019). The datasets contain code-mixing of various languages with English. Thus, English construes a large part of, and is a common thread in, language data that we analyse. Both these models also have multilingual versions, mentioned below, which serves us well for comparability.

Massively Multilingual Models The multilingual BERT model (mBERT) (Devlin et al., 2018) is a transformer model pretrained on the Wikipedias of 104 languages including some Indic and African languages. XLM-RoBERTa (XLM-R) (Conneau et al., 2020) is the multilingual version of RoBERTa, pretrained on 100 languages from the CommonCrawl corpus. The Hindi included in the pretraining is romanized Hindi, instead of Devanagari Hindi, which is notable for our purposes since we only have romanized Hindi in our Hinglish code-mixed datasets. XLM-R specialises in cross-lingual representations. Both PLMs were chosen based on their competitive performance on low-resource languages.

Indic Language Models Introduced by Doddapaneni et al. (2022), IndicBERT v2 is a PLM incorporating 24 Indian languages, including English. It is a standard BERT model pretrained on IndicCorp v2, introduced in the same paper, with the Masked Language Modelling (MLM) objective function. While there are different flavours of the model available that are trained on an additional Translation Language Modelling (TLM) objective, we use the standard MLM-only model since we found marginal differences in the scores when we tested

Dataset	**IndicBERT**	**MuRIL**	**AfriBERTa**	**Afroxlmr**	**mBERT**	**XLM-R**	**BERT**	**RoBERTa**	**HingMBERT**
AfriSenti	-	-	0.75	**0.78**	0.76	0.77	0.77	0.76	0.77
NaijaVader	-	-	0.72	0.74	0.73	**0.74**	0.74	0.73	0.73
SAIL	0.62	0.62	-	-	0.60	0.64	0.60	0.61	**0.66**
IIITH-CodeMix	0.69	0.73	-	-	0.69	0.71	0.70	0.70	**0.74**
TamilMixSentiment	**0.71**	0.70	-	-	0.70	0.71	0.70	0.71	0.71
DravidianCodeMix (tam)	0.64	0.64	-	-	0.65	**0.66**	0.65	0.65	0.66
MalayalamMixSentiment	0.73	0.73	-	-	0.73	**0.74**	0.71	0.74	0.73
DravidianCodeMix (mal)	0.76	**0.77**	-	-	0.75	0.76	0.76	0.75	0.77
DravidianCodeMix (kan)	**0.71**	0.70	-	-	0.70	0.67	0.66	0.70	0.70

Table 3: Accuracy scores on the validation sets. Bold indicates best result for a dataset. The first two datasets are in Naija, next two in Hinglish, then Tamil-English, Malayalam-English, and the final single dataset is for Kannada-English code-mixing.

both the models on our datasets.

MuRIL (Khanuja et al., 2021) contains 16 Indian languages and English, from the Common Crawl OSCAR corpus, Wikipedia, PMINDIA corpus and the Dakshina Dataset, trained on the MLM and TLM objective functions. The TLM objective leverages both translated and transliterated data, to account for code-mixing.

African Language Models For the Naija datasets, we compare two language models trained on African languages, and the only models in our roster that include Naija in the pretraining.

AfriBERTa (Ogueji et al., 2021) is a transformer-based language model pretrained on 11 low-resourced African languages, with data sourced from the BBC news and the Common Crawl Corpus. It is trained with the standard MLM objective.

AfroXLMR (Alabi et al., 2022) is currently the largest available PLM for African languages. This model results from applying multilingual adaptive finetuning on XLM-R, with language adaption being performed on 17 African languages, and 3 other high resource languages spoken on the continent, including English sourced from the mt5 pretraining corpus, the BBC and other news websites.

Code-mixed Language Model We also include HingMBERT (Nayak and Joshi, 2022), a PLM containing Hinglish data in the pretraining. It is a multilingual BERT model that has been further pretrained on the L3Cube-HingCorpus. In the same work, the HingCorpus consists of code-mixed tweets - both in Latin script and transliterated into Devanagari. While there is a version of the model that has been pretrained on both Latin and Devanagari script, we use HingMBERT pretrained only on the latinized corpus to match our data.

In summary, each of the above PLMs selected for this work included training data for at least one language relevant to the target code-mixed data. Thus, we refine our hypothesis:

Refined Hypothesis: *Indic language models would perform better on the Indic datasets, and the African language models would perform better on the Naija datasets, than the monolingual or multilingual language models. Additionally, the code-mixed language model would perform better on the Hinglish datasets than the other PLMs.*

5 Experiments

We used the Massive Choice Ample Tasks (MaChAmp) (van der Goot et al., 2021) codebase for the experiments. MaChAmp provides an efficient and effective way to finetune PLMs on downstream tasks.

5.1 Finetuning

We finetuned the models on the training data from the code-mixed datasets. For the Indic datasets we finetuned the monolingual, multingual, codemix and Indic language models, while for the Naija datasets we finetuned the monolingual, multilingual, code-mixed and African models. We ran the experiments for 50 epochs, maintaining the same hyperparameters across all the models and datasets, and chose the model with the best performance on the validation set.

Finetuning Results We report the validation scores of each model-dataset combination in Table 3. Contrary to the hypothesis, there is not a very tangible difference observed between the performance of each model on the datasets. Models trained on relevant languages in some cases do have the best performance, like Afroxlmr with AfriSenti, which as seen in Table 2 contains Naija in the pretraining. Similarly with HingMBERT and the Hinglish datasets, and MuRil and IndicBERT with DravidianCodeMix (kan) and TamilMixSentiment, but this difference is very marginal. MuRil, trained on Indic languages, outperforms monolingual BERT on DravidianCodeMix (mal) by just one accuracy point. So does Afroxlmr with AfriSenti, where BERT is just one point behind.

On the other hand, for the datasets NaijaVader, MalayalamMixSentiment and DravidianCodeMix (tam), where the PLMs trained on relevant language families do not outperform the other models, XLM-R comes on top, but again with minimal difference. For NaijaVader, three categories of PLMs have very similar accuracy scores - BERT from the monolingual category, Afroxlmr from the African category and XLM-R from the multilingual category.

5.2 Other Tasks

Results from the above section raise the question whether models perform fairly similarly because the models are able to learn simple spurious correlations to classify sentiment, rather than relying on the PLM's capacity to understand the code-mixed data. To rule out this possibility, we performed similar experiments with Named Entity Recognition (NER), sarcasm detection and universal dependency parsing (UDPoS) datasets. If PLM performance on these tasks yield similar results to the sentiment analysis tasks, we can conclude that our findings thus far are pertinent to the capabilities of PLMs on code-mixed data, generally.

NER For NER, we use the dataset introduced by Singh et al. (2018), which is also part of the GlueCoS benchmark. It is a Hinglish dataset of code-mixed tweets annotated with BIO labels for persons, organisations and locations. The authors did not release official train-dev-test splits for the data, so we created our own, resulting in 50k tokens in the training set, and 7k in the validation. We then finetuned the monolingual, multilingual, code-mixed and Indic models on the training data. We also ran a similar experiment with the monolingual, multilingual and African models on the Naija part of MasakhaNER (Adelani et al., 2021), which showed similar results as discussed for Hinglish below. However, since MasakhaNER is sourced from BBC Pidgin, and owing to the formality of the register is less likely to contain code-switching, we report the results for it in Appendix A.1.

Sarcasm Detection For sarcasm detection, we use the dataset curated by Shah and Maurya (2021), consisting of 144k tweets in Hinglish. They are annotated based on the presence of hashtags, where all tweets with #sarcasm, #sarcastic, #irony, #humor were labelled as positive, and others with general hashtags like #politics, #food, #movie were labelled as negative for sarcasm. We used the splits released by the authors, and finetuned the monolingual, multilingual, code-mixed and Indic models on the training data consisting of 115K examples.

	NER	Sarcasm	UDPoS
IndicBERT	0.77	0.89	-
MuRIL	0.77	0.90	-
AfriBERTa	-	-	0.99
AfroXLMR	-	-	**0.99**
mBERT	0.78	0.89	0.99
XLM-R	0.77	0.90	0.99
BERT	0.76	0.89	0.99
RoBERTa	0.76	0.89	0.99
HingMBERT	**0.78**	**0.90**	-

Table 4: NER span-f1 and accuracy scores for sarcasm detection and UDPoS on validation sets.

UDPoS For UDPoS we use the Naija dataset introduced by Caron et al. (2019), consisting of 140k words. While it is not a social media dataset, it contains transcriptions of spoken Naija from different domains like speeches, free conversations, comments about current affairs, radio programs etc. Spoken data such as the kind included in this dataset contains a similar informality to social media, and thus likely to also contain code-switching. We used the official splits released by the authors and finetuned the monolingual, multilingual and African models on the training data.

Other Results The scores for sequence labelling with NER and UDPoS, and classification with sarcasm detection, presented in Table 4, show similar trends to that of sentiment analysis. All the models perform equally well, with the difference between the best and the worst being 2 percentage points in NER, 1 percentage point in sarcasm detection and less than 1 percentage point in UDPoS.

5.3 Zero-shot

Since there were only slight differences observed between the models when finetuning on code-mixed data, we evaluated the models on the code-mixed data in a zero-shot setting. In this scenario, there was no code-mixed data present in the downstream finetuning of the models, before testing on code-mixed data. We performed the zero-shot experiments with the Hinglish datasets and thus, we used monolingual Hindi and English sentiment analysis datasets for downstream finetuning of the monolingual, multilingual, code-mixed and Indic models. This could potentially bring out differences in model performance, if any, that arise from differences in pretraining data.

For the Hindi data, we used the sentiment analysis dataset created by Akhtar et al. (2016), which is also included in the IndicGLUE benchmark (Kakwani et al., 2020). It contains two individual datasets from two different domains - movie reviews and product reviews. While the movie reviews contain entire reviews that can potentially span one or two paragraphs as individual data points, the product reviews contain one or two sentences. Thus, to match the structure of the code-mixed datasets, we only use the product review dataset for downstream finetuning in Hindi. This dataset is in the Devanagari script, so we first transliterated it into Latin script for comparability.

For the English data, we used a reduced version of the SST-2 dataset (Socher et al., 2013), from the GLUE benchmark (Wang et al., 2018), reduced to match the size of the Hindi dataset to eliminate size as a potential factor in the results. We then evaluated these models on the validation sets from SAIL and IIITH-CodeMix. Moreover, the English and the Hindi datasets only have two sentiment labels - 'negative' and 'positive'. Thus, we removed the instances labelled 'neutral' from the Hinglish validation sets for this scenario.

Zero-shot Results Scores from the zero-shot experiments are in Table 5. Pretraining data here seems to make a drastic difference in the relative performance of the models. For both datasets, HingMBERT outperforms other models by a substantial margin, in both English and Hindi settings. When comparing models that do not contain code-mixed data in the pretraining, in the English setting, RoBERTa performs the best on both the datasets. On the other hand, MuRIL shows a very drastic decline in accuracy, being the worst on both datasets. This is reversed in the Hindi setting, where MuRIL outperforms the others, and RoBERTa is the least accurate by a large margin.

6 Analysis

It can be inferred from the above results that for code-mixed datasets, when finetuning a PLM on the code-mixed language, the languages seen in the pretraining may not substantially impact the performance of the model. We further confirmed this by finetuning the African models on IIITH-CodeMix, and the Indic models on NaijaVader. The results are in Table 6.

	SAIL		IIITH-CodeMix	
	Hindi	**English**	**Hindi**	**English**
IndicBERT	0.62	0.61	0.60	0.56
MuRIL	0.64	0.57	0.74	0.43
mBERT	0.57	0.56	0.64	0.47
XLM-R	0.63	0.62	0.70	0.46
BERT	0.61	0.62	0.63	0.57
RoBERTa	0.61	0.66	0.55	0.73
HingMBERT	**0.72**	**0.69**	**0.78**	**0.77**

Table 5: Zero-shot scores on Hinglish validation sets with Hindi and English task-specific finetuning.

	IIITH-Codemix	NaijaVader
IndicBERT	0.69	0.74
MuRIL	0.73	0.72
AfriBERTa	0.68	0.72
Afroxlmr	0.70	0.74
Best Model	0.74	0.74

Table 6: Accuracy scores of Indic models on a Naija dataset and African models on a Hinglish dataset, along with the best scores for each dataset from Table 3.

IndicBERT on NaijaVader is on par with the best performing model, and the African models do not demonstrate a drastic decline in performance on IIITH-CodeMix as compared to the Indic models. On the other hand, the pretraining languages of a PLM greatly influence performance scores when testing on code-mixed data in a zero-shot setting.

6.1 Language Identification and Composition

To understand these scores further, we looked at the composition of each participating language in the datasets, and compared the predictions of each model to see, whether despite overall accuracy being similar in the finetuning scenario, the models were performing better on one language than the other.

To this end, we ran a language identification (LID) model for code-mixed data on the Hinglish validation sets, using the CodeSwitch (Sarkar, 2020) tool, trained on data from the LinCE benchmark. The LID model takes in a code-mixed sentence, tokenizes it into subwords and outputs a language score for each subword. There were instances where the model assigned different languages for subwords from the same word. In these cases we picked the language assigned to the first subword. We manually verified the accuracy of LID on a sample from the IIITH-CodeMix dataset, and with a 95% accuracy, found it suitable enough for our purposes.

We assigned a majority language to each instance in the dataset, where if the instance had more than 50% words in English, it was categorised as *mostly-English*, and *mostly-Hindi* otherwise. Thus, we looked at the predictions of each model for the *mostly-English* and *mostly-Hindi* sentences to see whether, for example, the Indic or code-mixed PLMs were outperforming on the *mostly-Hindi* sentences, and failing on the *mostly-English*.

6.2 Implications of Language Composition: The Finetuning Scenario

Figure 1 illustrates the results. For IIITH-CodeMix, all models perform similarly on the *mostly-Hindi* examples, with MuRIL and HingMBERT performing slightly better. There are slightly larger differences in performance with the *mostly-English* examples, with the monolingual and code-mixed PLMs performing better than the multilingual and Indic PLMs. For the SAIL dataset, there is also a difference seen in performance on the *mostly-Hindi* examples, where the code-mixed PLM is able to handle them the best, followed closely by multilingual XLM-R. Not surprisingly, the monolingual models trail behind, with almost a 10 percentage point difference between HingMBERT and BERT. The *mostly-English* examples have similar performances across the models, with monolingual RoBERTa slightly ahead. All models perform better on *mostly-English* than on *mostly-Hindi* examples, with the pretraining language of the PLM potentially accounting for how big that difference is. The difference is larger in monolingual models compared to the others.

Another notable observation is that for SAIL, HingMBERT performs almost equally on *mostly-English* and *mostly-Hindi* examples. This could be attributed to the language composition of each dataset, where about 40% of the SAIL dataset is *mostly-English*, while the IIITH-CodeMix dataset only has about 14% *mostly-English*. Thus the distribution of the parent languages is more even in SAIL and heavily skewed towards Hindi in IIITH-CodeMix. Therefore, it can be argued that the code-mixed language model also learns the distribution of the participating languages in the dataset during training, and that reflects on the predictions of the model.

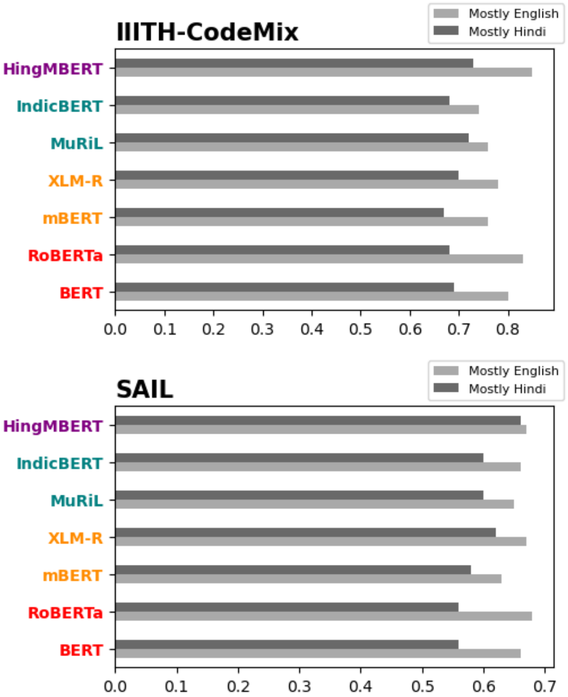


Figure 1: PLM performance relative to LID. The IIITH-CodeMix dev set was 14% *mostly-English* utterances, while the SAIL dev set was 40% *mostly-English* utterances.

We also looked at the distribution of sentiment labels for the *mostly-English* and *mostly-Hindi* examples, and compared model predictions to see if the models showed any bias toward a particular label for a language, but we saw no difference.

Since there are no such LID tools available for the other languages in our roster, we tested the CodeSwitch LID tool on samples from the other datasets as well. We found that the model is able to identify the English words in the samples satisfactorily, if not the other participating languages. So we ran the LID model on all the validation sets from the rest of the Indic and Naija datasets, and conducted similar analyses. The results confirmed the findings from the Hinglish datasets, but since the tool is not very reliable for these languages, we only report the results in Appendix A.2.

6.3 Implications of Language Composition: The Zero-Shot Scenario

The scenario described in the previous section takes a turn when evaluating the models in a zero-shot setting. From the results in Table 5, we find that pretraining has a major impact on the model performance, along with the composition of the parent languages in the dataset. As mentioned before, SAIL has a much more even composition of *mostly-Hindi* and *mostly-English* examples than IIITH-CodeMix.

This reflects in the performance of the models with respect to the finetuning language. While the code-mixed PLM does not show much difference in both scenarios on both datasets, the multilingual models suffer more with English finetuning than Hindi on IIITH-CodeMix, but do not show much difference in SAIL. Interestingly, BERT seems to suffer with English finetuning on IIITH-CodeMix, while RoBERTa has a jump in performance, even though they are both monolingual models pretrained on English data, and IIITH-CodeMix has more Hindi than English text. RoBERTa, in fact, suffers from Hindi finetuning on both the datasets. Conversely, MuRIL always suffers from English finetuning, more on IIITH-CodeMix than SAIL, which can be attributed to parent language composition of the datasets.

When comparing IndicBERT and MuRIL, differences in pretraining also reflect on the scores. MuRIL has seen romanized Hindi, with the TLM objective leveraging transliterated data as well, while the IndicBERT model we used has not. Thus, when finetuning with romanized Hindi, MuRIL has a significant bump in performance, in both cases performing better than IndicBERT. This could also be seen as a drawback for MuRIL when finetuning with English since it performs worse than IndicBERT on both SAIL and IIITH-CodeMix.

7 Summary

We summarise the findings of the paper in this section to answer the main underlying question of this work - do pretraining languages matter? We approach this question for code-mixed data in two transfer learning settings: with in-language finetuning, and zero-shot.

- When finetuning a PLM on a code-mixed dataset, the effects of the pretraining languages of the PLMs do not reflect in the performance scores substantially.
- In the finetuning setting when looking at PLM performance relative to language ID, all the PLMs perform better on the *mostly-English* sentences, than on *mostly-Hindi*, with the pretraining languages of the PLM and the language composition of the dataset potentially accounting for how big that difference is.
- In a zero-shot setting, the pretraining languages of the PLM do matter for performance.
- The language used to finetune a PLM greatly affects performance in the zero-shot setting. MuRIL is the best performing model with Hindi finetuning and RoBERTa has the highest score with English finetuning. The language composition of the dataset also potentially affects how much the score of the best performing model differs from the least performing model.

8 Conclusion

In this study, we found that the pretraining languages do not matter much for performance when downstream finetuning a PLM on code-mixed data. The finetuning process, to an extent, negates the effects of the pretraining languages in the PLMs and generates even performance across the board. On the other hand, the pretraining language of the models and the language composition of the data, both seem to be factors in model performance in a zero-shot setting. Overall, it can be better to use a PLM with pretraining on code-mixed languages like Hinglish, but this may not be possible for all types of code-mixed languages. Moreover, it does not seem to prove advantageous when it comes to Naija. Thus, this study can be used as a starting point for further interpretability analysis of PLMs, to understand exactly why in some settings the pretraining languages matter, and in some settings they don't.

9 Limitations

A large limitation of this work is the ubiquity of English. With the exception of the AfriBERTa (which has seen Naija), the remaining PLMs in this study all included English in the pretraining data. As a result, it is difficult to disentangle the benefits of including relevant languages in the pretraining data, from the general benefits of including *English* in the pretraining data, for processing code-mixed text. To this effect, future work in examining the capacity of PLMs for code-mixed language would benefit from examining commonly code-mixed language pairs, that do not involve English (e.g. Turkish-German).

In a similar vein, our work is limited in that we did not try other non-English monolingual PLMs. For the Indic languages, this is because monolingual Indic PLMs typically use the Devanagari script, but the datasets in this paper are constrained to using the Latin script. For Naija, we likewise did not experiment with monolingual models for the other relevant Nigerian languages; to our knowledge, most publicly available PLMs for Hausa, Yoruba, and Igbo seem to be created through continued pretraining with monolingual data over existing multilingual PLMs. Thus, experimenting with these models still does not strictly control for English and other languages.

Beyond PLMs, another limitation of this work pertains to the error analysis, which hinges upon currently available LID technologies. As explored in detail by Caswell et al. (2020), most LID technologies operate on a document level, and thus intra-utterance LID is still an open problem. For code-mixed language, the lack of robust LID puts limits us to coarser-grained analysis of the data (e.g. partitioning samples by *mostly-English* or *mostly-Hindi*). Ideally, a finer-grained partition of the data could be useful in determining the extent to which a PLM's knowledge of English enables performance on downstream tasks.

10 Acknowledgements

The computational resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation - Flanders (FWO) and the Flemish Government - department EWI (for Kushal Tatariya and Miryam de Lhoneux). This work is also funded by the Carlsberg Foundation under an Accelerate career grant entitled "Multilingual Modelling for Resource-Poor Languages", grant code CF21-0454 (for Heather Lent).

References

Ife Adebara, AbdelRahim Elmadany, Muhammad Abdul-Mageed, and Alcides Alcoba Inciarte. 2022. Afrolid: A neural language identification tool for african languages. In *Conference on Empirical Methods in Natural Language Processing*.

David Ifeoluwa Adelani, Jade Abbott, Graham Neubig, Daniel D'souza, Julia Kreutzer, Constantine Lignos, Chester Palen-Michel, Happy Buzaaba, Shruti Rijhwani, Sebastian Ruder, Stephen Mayhew, Israel Abebe Azime, Shamsuddeen H. Muhammad, Chris Chinenye Emezue, Joyce Nakatumba-Nabende, Perez Ogayo, Aremu Anuoluwapo, Catherine Gitau, Derguene Mbaye, Jesujoba Alabi, Seid Muhie Yimam, Tajuddeen Rabiu Gwadabe, Ignatius Ezeani, Rubungo Andre Niyongabo, Jonathan Mukiibi, Verrah Otiende, Iroro Orife, Davis David, Samba Ngom, Tosin Adewumi, Paul Rayson, Mofetoluwa Adeyemi, Gerald Muriuki, Emmanuel Anebi, Chiamaka Chukwuneke, Nkiruka Odu, Eric Peter Wairagala, Samuel Oyerinde, Clemencia Siro, Tobius Saul Bateesa, Temilola Oloyede, Yvonne Wambui, Victor Akinode, Deborah Nabagereka, Maurice Katusiime, Ayodele Awokoya, Mouhamadane MBOUP, Dibora Gebreyohannes, Henok Tilaye, Kelechi Nwaike, Degaga Wolde, Abdoulaye Faye, Blessing Sibanda, Orevaoghene Ahia, Bonaventure F. P. Dossou, Kelechi Ogueji, Thierno Ibrahima DIOP, Abdoulaye Diallo, Adewale Akinfaderin, Tendai Marengereke, and Salomey Osei. 2021. MasakhaNER: Named entity recognition for African languages. *Transactions of the Association for Computational Linguistics*, 9:1116–1131.

Prabhat Agarwal, Ashish Sharma, Jeenu Grover, Mayank Sikka, Koustav Rudra, and Monojit Choudhury. 2017. I may talk in english but gaali toh hindi mein hi denge : A study of english-hindi code-switching and swearing pattern on social networks. In *2017 9th International Conference on Communication Systems and Networks (COMSNETS)*, pages 554–557.

Ogechi Florence Agbo and Ingo Plag. 2020. The relationship of nigerian english and nigerian pidgin in nigeria: Evidence from copula constructions in ice-nigeria. *Journal of Language Contact*.

Gustavo Aguilar, Fahad AlGhamdi, Victor Soto, Thamar Solorio, Mona Diab, and Julia Hirschberg, editors. 2018. *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching*. Association for Computational Linguistics, Melbourne, Australia.

Gustavo Aguilar, Sudipta Kar, and Thamar Solorio. 2020. LinCE: A centralized benchmark for linguistic code-switching evaluation. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1803–1813, Marseille, France. European Language Resources Association.

Akinmade T. Akande and Oladipo Salami, editors. 2021. *Current Trends in Nigerian Pidgin English A Sociolinguistic Perspective*. De Gruyter Mouton, Berlin, Boston.

Md Shad Akhtar, Ayush Kumar, Asif Ekbal, and Pushpak Bhattacharyya. 2016. A hybrid deep learning architecture for sentiment analysis. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 482–493, Osaka, Japan. The COLING 2016 Organizing Committee.

Jesujoba O. Alabi, David Ifeoluwa Adelani, Marius Mosbach, and Dietrich Klakow. 2022. Adapting pretrained language models to African languages via multilingual adaptive fine-tuning. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4336–4349, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Somnath Banerjee, Sudip Kumar Naskar, Paolo Rosso, and Sivaji Bandyopadhyay. 2016. The first cross-script code-mixed question answering corpus. In *MultiLingMine@ECIR*.

Utsab Barman, Amitava Das, Joachim Wagner, and Jennifer Foster. 2014. Code mixing: A challenge for language identification in the language of social media. In *CodeSwitch@EMNLP*.

Chatterjee Barnali. 2017. Code-switching and mixing in communication a study on language contact in indian media. *Social Science Research Network*.

Aditya Bohra, Deepanshu Vijay, Vinay Singh, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. A dataset of hindi-english code-mixed social media text for hate speech detection. In *PEOPLES@NAACL-HTL*.

Bernard Caron, Marine Courtin, Kim Gerdes, and Sylvain Kahane. 2019. A surface-syntactic UD treebank for Naija. In *Proceedings of the 18th International Workshop on Treebanks and Linguistic Theories (TLT, SyntaxFest 2019)*, pages 13–24, Paris, France. Association for Computational Linguistics.

Isaac Caswell, Theresa Breiner, Daan van Esch, and Ankur Bapna. 2020. Language ID in the wild: Unexpected challenges on the path to a thousand-language web text corpus. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6588–6608, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Kunal Chakma and Amitava Das. 2016. Cmir: A corpus for evaluation of code mixed information retrieval of hindi-english tweets. *Computación y Sistemas*, 20:425–434.

Bharathi Raja Chakravarthi, Navya Jose, Shardul Suryawanshi, Elizabeth Sherly, and John Philip McCrae. 2020a. A sentiment analysis dataset for code-mixed Malayalam-English. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, pages 177–184, Marseille, France. European Language Resources association.

Bharathi Raja Chakravarthi, Vigneshwaran Muralidaran, Ruba Priyadharshini, and John Philip McCrae. 2020b. Corpus creation for sentiment analysis in code-mixed Tamil-English text. In *Proceedings of the 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL)*, pages 202–210, Marseille, France. European Language Resources association.

Bharathi Raja Chakravarthi, Ruba Priyadharshini, Vigneshwaran Muralidaran, Navya Jose, Shardul Suryawanshi, Elizabeth Sherly, and John P. McCrae. 2021. Dravidiancodemix: Sentiment analysis and offensive language identification dataset for dravidian languages in code-mixed text. *CoRR*, abs/2106.09460.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Wietse de Vries, Martijn Bartelds, Malvina Nissim, and Martijn Wieling. 2021. Adapting monolingual models: Data can be scarce when language similarity is high. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4901–4907, Online. Association for Computational Linguistics.

Wietse de Vries, Martijn Wieling, and Malvina Nissim. 2022. Make the best of cross-lingual transfer: Evidence from POS tagging with over 100 languages. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7676–7685, Dublin, Ireland. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Sumanth Doddapaneni, Rahul Aralikatte, Gowtham Ramesh, Shreya Goyal, Mitesh M. Khapra, Anoop Kunchukuttan, and Pratyush Kumar. 2022. Indicxtreme: A multi-task benchmark for evaluating indic languages.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. 2021. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1286–1305, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

A. Seza Doğruöz, Sunayana Sitaram, Barbara E. Bullock, and Almeida Jacqueline Toribio. 2021. A survey of code-switching: Linguistic and social perspectives for language technologies. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1654–1666, Online. Association for Computational Linguistics.

Omowumi Olabode Steven Ekundayo. 2022. Naija: The cinderella for nigerian and west african national language, unity and identity. *Journal of General Education and Humanities*.

Vivek Kumar Gupta. 2019. "hinglish" language - modeling a messy code-mixed language. *CoRR*, abs/1912.13109.

Suchin Gururangan, Dallas Card, Sarah K. Dreier, Emily K. Gade, Leroy Z. Wang, Zeyu Wang, Luke Zettlemoyer, and Noah A. Smith. 2022. Whose language counts as high quality? measuring language ideologies in text data selection. *CoRR*, abs/2201.10474.

Clayton J. Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. *Proceedings of the International AAAI Conference on Web and Social Media*.

Aditya Joshi, Ameya Prabhu, Manish Shrivastava, and Vasudeva Varma. 2016. Towards sub-word level compositions for sentiment analysis of Hindi-English code mixed text. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2482–2491, Osaka, Japan. The COLING 2016 Organizing Committee.

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLPSuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for Indian languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4948–4961, Online. Association for Computational Linguistics.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, Shruti Gupta, Subhash Chandra Bose Gali, Vish Subramanian, and Partha Talukdar. 2021. Muril: Multilingual representations for indian languages.

Simran Khanuja, Sandipan Dandapat, Anirudh Srinivasan, Sunayana Sitaram, and Monojit Choudhury. 2020. GLUECoS: An evaluation benchmark for code-switched NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3575–3585, Online. Association for Computational Linguistics.

Anne Lauscher, Vinit Ravishankar, Ivan Vulic, and Goran Glavas. 2020. From zero to hero: On the limitations of zero-shot cross-lingual transfer with multilingual transformers. *CoRR*, abs/2005.00633.

Heather Lent, Kelechi Ogueji, Miryam de Lhoneux, Orevaoghene Ahia, and Anders Søgaard. 2022. What a creole wants, what a creole needs. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6439–6449, Marseille, France. European Language Resources Association.

Yu-Hsiang Lin, Chian-Yu Chen, Jean Lee, Zirui Li, Yuyan Zhang, Mengzhou Xia, Shruti Rijhwani, Junxian He, Zhisong Zhang, Xuezhe Ma, Antonios Anastasopoulos, Patrick Littell, and Graham Neubig. 2019. Choosing transfer languages for cross-lingual learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3125–3135, Florence, Italy. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Yash Madhani, Sushane Parthan, Priyanka A. Bedekar, Ruchi Khapra, Vivek Seshadri, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh M. Khapra. 2022. Aksharantar: Towards building open transliteration tools for the next billion users. *ArXiv*, abs/2205.03018.

Eyo O. Mensah and Roseline Ihuoma Ndimele. 2014. Linguistic creativity in nigerian pidgin advertising. *Sociolinguistic Studies*, 7:321–344.

Ibraheem Muhammad Moosa, Mahmud Elahi Akhter, and Ashfia Binte Habib. 2023. Does transliteration help multilingual language modeling?

Shamsuddeen Hassan Muhammad, Idris Abdulmumin, Abinew Ali Ayele, Nedjma Ousidhoum, David Ifeoluwa Adelani, Seid Muhie Yimam, Ibrahim Sa'id Ahmad, Meriem Beloucif, Saif Mohammad, Sebastian Ruder, Oumaima Hourrane, Pavel Brazdil, Felermino Dário Mário António Ali, Davis Davis, Salomey Osei, Bello Shehu Bello, Falalu Ibrahim, Tajuddeen Gwadabe, Samuel Rutunda, Tadesse Belay, Wendimu Baye Messelle, Hailu Beshada Balcha, Sisay Adugna Chala, Hagos Tesfahun Gebremichael, Bernard Opoku, and Steven Arthur. 2023. Afrisenti: A twitter sentiment analysis benchmark for african languages.

Ravindra Nayak and Raviraj Joshi. 2022. L3cube-hingcorpus and hingbert: A code mixed hindi-english dataset and bert language models.

Innocent Ndubuisi-Obi, Sayan Ghosh, and David Jurgens. 2019. Wetin dey with these comments? modeling sociolinguistic factors affecting code-switching behavior in Nigerian online discussions. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6204–6214, Florence, Italy. Association for Computational Linguistics.

Nancy Chiagolum Odiegwu. 2022. Review of current trends in nigerian pidgin english. a sociolinguistic perspective. *Corpus Pragmatics*, 6:89 – 93.

Kelechi Ogueji, Yuxin Zhu, and Jimmy Lin. 2021. Small data? no problem! exploring the viability of pretrained multilingual language models for low-resourced languages. In *Proceedings of the 1st Workshop on Multilingual Representation Learning*, pages 116–126, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wuraola Fisayo Oyewusi, Olubayo Adekanmbi, and Olalekan Akinsande. 2020. Semantic enrichment of nigerian pidgin english for contextual sentiment classification. *CoRR*, abs/2003.12450.

Braja Gopal Patra, Dipankar Das, and Amitava Das. 2018a. Sentiment analysis of code-mixed indian languages: An overview of sail_code-mixed shared task @icon-2017. *ArXiv*, abs/1803.06745.

Braja Gopal Patra, Dipankar Das, and Amitava Das. 2018b. Sentiment analysis of code-mixed indian languages: An overview of sail_code-mixed shared task @icon-2017. *CoRR*, abs/1803.06745.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

Krithika Ramesh, Sunayana Sitaram, and Monojit Choudhury. 2023. Fairness in language models beyond english: Gaps and challenges.

Koustav Rudra, Shruti Rijhwani, Rafiya Begum, Kalika Bali, Monojit Choudhury, and Niloy Ganguly. 2016. Understanding language preference for expression of opinion and sentiment: What do Hindi-English speakers do on Twitter? In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1131–1141, Austin, Texas. Association for Computational Linguistics.

Balogun Sarah and Murana Muniru Oladayo. 2021. Code-switching and code mixing in the selected tracks of the hip hop music of flavour and 9ice. *International Journal of English and Comparative Literary Studies*, 2(3):55–70.

Sagor Sarkar. 2020. Code switch.

Aditya Shah and Chandresh Maurya. 2021. How effective is incongruity? implications for code-mixed sarcasm detection. In *Proceedings of the 18th International Conference on Natural Language Processing (ICON)*, pages 271–276, National Institute of Technology Silchar, Silchar, India. NLP Association of India (NLPAI).

Vinay Singh, Deepanshu Vijay, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. Named entity recognition for Hindi-English code-mixed social media text. In *Proceedings of the Seventh Named Entities Workshop*, pages 27–35, Melbourne, Australia. Association for Computational Linguistics.

Sunayana Sitaram, Khyathi Raghavi Chandu, Sai Krishna Rallabandi, and Alan W. Black. 2019. A survey of code-switched speech and language processing. *CoRR*, abs/1904.00784.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Thamar Solorio, Shuguang Chen, Alan W. Black, Mona Diab, Sunayana Sitaram, Victor Soto, Emre Yilmaz, and Anirudh Srinivasan, editors. 2021. *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*. Association for Computational Linguistics, Online.

Thamar Solorio, Monojit Choudhury, Kalika Bali, Sunayana Sitaram, Amitava Das, and Mona Diab, editors. 2020. *Proceedings of the The 4th Workshop on Computational Approaches to Code Switching*. European Language Resources Association, Marseille, France.

Zeerak Talat, Aurélie Névéol, Stella Biderman, Miruna Clinciu, Manan Dey, Shayne Longpre, Sasha Luccioni, Maraim Masoud, Margaret Mitchell, Dragomir Radev, Shanya Sharma, Arjun Subramonian, Jaesung Tae, Samson Tan, Deepak Tunuguntla, and Oskar Van Der Wal. 2022. You reap what you sow: On the challenges of bias evaluation under multilingual settings. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 26–41, virtual+Dublin. Association for Computational Linguistics.

Sarah G. Thomason. 2001. *Language Contact*. Edinburgh University Press, Edinburgh.

Rob van der Goot, Ahmet Üstün, Alan Ramponi, Ibrahim Sharaf, and Barbara Plank. 2021. Massive choice, ample tasks (MaChAmp): A toolkit for multi-task learning in NLP. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 176–197, Online. Association for Computational Linguistics.

Deepanshu Vijay, Aditya Bohra, Vinay Singh, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. A dataset for detecting irony in hindi-english code-mixed social media text. In *EMSASW@ESWC*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Genta Indra Winata, Alham Fikri Aji, Zheng Xin Yong, and Thamar Solorio. 2022. The decades progress on code-switching research in nlp: A systematic survey on trends and challenges. *ArXiv*, abs/2212.09660.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.

Shijie Wu and Mark Dredze. 2020. Are all languages created equal in multilingual bert? *CoRR*, abs/2005.09093.

George-Eduard Zaharia, George-Alexandru Vlad, Dumitru-Clementin Cercel, Traian Rebedea, and Costin Chiru. 2020. UPB at SemEval-2020 task 9: Identifying sentiment in code-mixed social media texts using transformers and multi-task learning. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1322–1330, Barcelona (online). International Committee for Computational Linguistics.

A Appendix

A.1 MasakhaNER Results

MasakhaNER	
Model	**Score**
BERT	0.89
RoBERTa	0.89
mBERT	0.90
XLM-R	**0.91**
AfriBERTa	0.89
AfroXLMR	0.90

Table 7: Span-f1 scores for MasakhaNER Naija. These results are consistent with those reported in Section 5.2.

A.2 Language ID Results for Other Datasets

Language ID results for the other datasets are reported here. The tables below contain the percentage of *mostly English* and *mostly Not-English* examples that each PLM correctly classified.

AfriSenti

	Mostly English	**Mostly Not-Eng**
Proportion	97.58%	2.42%
BERT	76.80%	90.32%
RoBERTa	76.32%	80.65%
mBERT	75.52%	80.65%
XLM-R	77.44%	77.42%
AfriBERTa	74.88%	83.87%
AfroXLMR	77.92%	74.19%

NaijaVader

	Mostly English	**Mostly Not-Eng**
Proportion	91.79%	8.21%
BERT	72.61%	86.09%
RoBERTa	72.14%	84.35%
mBERT	72.22%	80.00%
XLM-R	72.61%	86.96%
AfriBERTa	70.97%	80.87%
AfroXLMR	72.68%	83.48%

TamilCodeMix

	Mostly English	**Mostly Not-Eng**
Proportion	35.99%	64.01%
BERT	71.56%	69.54%
RoBERTa	72.69%	69.54%
mBERT	72.46%	69.04%
XLM-R	72.46%	69.16%
MuRiL	72.23%	68.65%
IndicBERT	72.91%	69.42%

MalayalamCodeMix

	Mostly English	**Mostly Not-Eng**
Proportion	19.58%	80.42%
BERT	75.53%	70.21%
RoBERTa	76.60%	72.80%
mBERT	79.79%	71.50%
XLM-R	81.91%	72.02%
MuRiL	80.85%	70.98%
IndicBERT	78.72%	71.76%

DravidianCodeMix (Kannada)

	Mostly English	**Mostly Not-Eng**
Proportion	31.40%	68.60%
BERT	69.90%	64.44%
RoBERTa	73.30%	68.44%
mBERT	68.93%	70.00%
XLM-R	66.99%	67.33%
MuRiL	70.39%	70.22%
IndicBERT	74.75%	69.11%

DravidianCodeMix (Tamil)

	Mostly English	**Mostly Not-Eng**
Proportion	26.73%	73.27%
BERT	71.29%	63.00%
RoBERTa	71.38%	62.42%
mBERT	69.69%	63.07%
XLM-R	70.93%	63.75%
MuRiL	68.18%	62.35%
IndicBERT	68.53%	62.87%

DravidianCodeMix (Malayalam)

	Mostly English	**Mostly Not-Eng**
Proportion	13.47%	86.53%
BERT	80.24%	75.52%
RoBERTa	80.24%	73.89%
mBERT	78.63%	74.58%
XLM-R	80.65%	74.89%
MuRiL	82.66%	75.58%
IndicBERT	84.27%	74.51%

Table 8: Proportion of *mostly English* and *mostly Not-Eng* examples in the dev sets, and the proportion of correctly classified examples by the models for each dev set.

Can ChatGPT Understand Causal Language in Science Claims?

Yuheun Kim[1] **Lu Guo**[1] **Bei Yu**[1] **Yingya Li**[2]
[1]School of Information Studies, Syracuse University
`{ykim72, lguo15, byu}@syr.edu`
[2]Harvard Medical School and Boston Children's Hospital
`yingya.li@childrens.harvard.edu`

Abstract

This study evaluated ChatGPT's ability to understand causal language in science papers and news by testing its accuracy in a task of labeling the strength of a claim as causal, conditional causal, correlational, or no relationship. The results show that ChatGPT is still behind the existing fine-tuned BERT models by a large margin. ChatGPT also had difficulty understanding conditional causal claims mitigated by hedges. However, its weakness may be utilized to improve the clarity of human annotation guideline. Chain-of-thought prompting was faithful and helpful for improving prompt performance, but finding the optimal prompt is difficult with inconsistent results and the lack of effective method to establish cause-effect between prompts and outcomes, suggesting caution when generalizing prompt engineering results across tasks or models.

1 Introduction

Finding causal relationship is an important goal in scientific research. However, choosing appropriate causal language that accurately reflects the strength of evidence is a non-trivial task when describing research findings. Subjectivity and bias may affect how authors interpret the results. For example, some researchers argued that observational studies can not illuminate causal claims and thus causal language should not be used (e.g., Cofield et al., 2010), while others called for more confidence in causal inference with improved methods and guidelines, (e.g., Pearl and Mackenzie, 2018). On the other hand, average human readers reported difficulty in judging the strength of causal claims mitigated with hedges such as "may" or ambiguous terms like "linked to" (Adams et al., 2017). Manual fact-checking of causal claims in academic publications, news and social media posts also demonstrated evidence of prevalent exaggeration when reporting causal findings (Cofield et al., 2010; Sumner et al., 2014; Haber et al., 2018).

Prior studies have also looked into computational approaches for identifying claim strengths and exaggerated claims. The core component is a text classification task that categorizes research findings by their strengths. The original task definition was based on a manual content analysis (Sumner et al., 2014), which defined seven certainty levels: no statement, explicit statement of no relation, correlational (e.g. "drinking wine is associated with increased cancer rates"), ambiguous (e.g. "drinking wine linked to cancer risk"), conditional causal (e.g. "drinking wine might increase cancer risk"), can cause (e.g, "drinking wine can increase cancer risk"), and unconditionally causal (e.g. "drinking wine increases cancer risk"). However, Adams et al. (2017) found that average human readers can distinguish three categories of relationship only: direct cause statements (e.g. "makes"), can cause statements (e.g. "can make"); and moderate cause statements (e.g. "might cause", "linked", "associated with"), and they encountered difficulty in distinguishing the conditional causal statements and correlational statements in the last group.

In light of these observations, the subsequent computational modeling studies simplified the task to classify four categories: direct causal, conditional causal, correlational, or no relationship (Yu et al., 2019; Tan et al., 2021; Yu et al., 2020; Wright and Augenstein, 2021). These specialized models used various techniques to achieve high accuracy, such as fine-tuning pre-trained BERT models (Yu et al., 2019), or through causal augmentation (Tan et al., 2021). These models also have limitations, such as mistaking a no-relationship sentence as causal or correlational when confounding cues exist. They also rely on thousands of human-annotated training examples.

Recently, the debut of large language models (LLMs) such as ChatGPT shifted the NLP research paradigm toward the direction of "pre-train, prompt, and predict", where downstream tasks are

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 379–389
July 14, 2023 ©2023 Association for Computational Linguistics

reformulated into textual prompts on zero-shot or few-shot settings (Liu et al., 2023). LLMs trained on sufficiently large and diverse datasets demonstrate promising performance on reasoning tasks without additional task specific training (Radford et al., 2019; Brown et al., 2020). The promising results inspired hope for eliminating the need for specialized models and expensive human annotations (Gilardi et al., 2023). A question rises then - can ChatGPT "understand" causal language in science writing? More specifically, can ChatGPT label the strength of causal claims correctly? Furthermore, since ChatGPT was trained with a variety of textual data, did it inherit the confusion that human readers have regarding conditional causal claims?

In this study, we evaluate ChatGPT's ability to understand causal claims in science papers and news. We tested GPT3.5 (text-davinci-003) and ChatGPT (gpt-3.5-turbo) performance in classifying causal claim strength using the annotated corpora provided by Yu et al. (2019) and Yu et al. (2020). Specifically, we ask the following research questions:

- RQ1: Does ChatGPT outperform fine-tuned BERT models for classifying causal claim strength?
- RQ2: How does ChatGPT interpret conditional causal claims?
- RQ3: Do GPT3.5 and ChatGPT agree on their predictions? Does ChatGPT give similar answers to semantically-similar prompts?
- RQ4: How do instructional elements in prompts, such as Chain-of-Thought, context, and system messages, affect ChatGPT performance?

Our paper is organized as follows: Section 2 summarizes studies on prompt engineering and various classification tasks evaluated on ChatGPT. In the Methods section, we introduce the different prompt designs we experimented (section 3.1), explain how we evaluated the model's performance (section 3.2), provide a summary of the data we used (section 3.3) and present the API details for the experiment (section 3.4). We then report experiment results that address RQ1 and RQ2 (section 4.1) as well as RQ3 and RQ4 (section 4.2). Based on the experiment we test the entire dataset (section 4.3) and also evaluate the result of applying ensemble models (section 4.4). Finally, we discuss our limitations and present our conclusion with discussions in section 5.

2 Related Work

Since prompts provide crucial information for LLMs such as ChatGPT, a number of studies have explored prompt engineering strategies (Liu et al., 2023). Here we summarize several common prompt design approaches with a focus on text classification tasks, which are most relevant to our study.

Zero-shot, one-shot, and few-shot learning are three types of prompting designs used to guide LLMs. Zero-shot prompting provides task descriptions or instructions without explicit examples. One-shot prompting uses a single example for the desired task. Few-shot prompting is similar to the one-shot design, but it involves providing the model with a small number of examples instead of just one for the model to learn from and generate task-aligned responses (Brown et al., 2020).

Prompts can be generated either manually or automatically (Brown et al., 2020; Radford et al., 2019; Petroni et al., 2019). While manual, intuitive approach is straightforward, it can be time-consuming to identify the most effective prompt and there is no guarantee to find one (Jiang et al., 2020). Researchers then sought automatic approaches (Liu et al., 2023; Gao et al., 2021; Raffel et al., 2020), or even asking ChatGPT itself to generate prompts (Zhong et al., 2023). However, since LLMs sometimes do not follow instructions, their answers may be ill-formatted or even invalid. When that happens, human intervention is needed, which increases the time cost for post-processing LLM results (Kocoń et al., 2023). Therefore, in this study, we focused on manually-generated prompts.

Text classification tasks often use instruction prompts to explicitly tell LLMs what to do. For instance, Qin et al. (2023)'s prompt starts with an instruction of task description: *"For each snippet of text, label the sentiment of the text as positive or negative. The answer should be exact 'positive' or 'negative' "*, followed with the text to be labeled. Ye et al. (2023) formulate their prompt as *"Definition: ... Input: ... Answer: ..."* where an example of definition can be *determine the speaker of the dialogue, agent or customer*. More context information about the task may be added to the instruction prompt, such as providing the definition of genre for genre classification (Kocoń et al., 2023).

Another commonly-used instructional element is the Chain-of-Thought (CoT), which has been found to improve LLMs' performance on certain arithmetic, commonsense, and symbolic reasoning tasks (Wei et al., 2022). While CoT was initially an instance of few-shot prompts, a decent zero-shot performance in reasoning tasks and classification tasks was demonstrated by adding a simple CoT prompt such as *"Let's think step by step"* at the end of a question (Kojima et al., 2022; Zhong et al., 2023).

ChatGPT also provides a unique feature, system messages, which can be used as part of the prompt to guide the model's behavior (Shen et al., 2023), such as *"You are a helpful assistant that can classify sentences as either causal or correlational research findings"* which specifies the model to behave as a professional for our task.

Previous studies have evaluated ChatGPT on various classification tasks (Qin et al., 2023; Bang et al., 2023; Huang et al., 2023; Kocoń et al., 2023), using various prompt designs. The results indicate promise and limitations. In the case of sentiment classification, ChatGPT was found to have difficulty in understanding neutral sentiment, or give unbalanced predictions on negative vs. positive sentiment, raising questions on the extent to which ChatGPT really "understands" sentiment as a linguistic concept (Wang et al., 2023). It is also difficult to directly compare the results due to different sample sets and prompts. The sample sizes were usually small since most studies were conducted before OpenAI made the API available.

3 Methods

3.1 Prompt Design

We experimented with intuitive trial-and-error approaches as well as consulting prior studies on the prompt designs that have demonstrated good performance in other text classification tasks. In this study we focused on zero-shot prompting design for two reasons. First, it is the most common strategy that end users choose to interact with ChatGPT. Second, since ChatGPT likely captures the latent social information (Horton, 2023), we are curious how ChatGPT "interpret" causal language without seeing training examples annotated by domain experts. All prompt designs that we have evaluated are documented in Table 1.

Our process started with a number of carefully crafted, intuitive prompts that include specific instructions. We then selected the best performing prompt as the baseline (BASE) for further comparison with other manually-constructed prompts from four previous studies with minor modifications to suit our task (Huang et al., 2023; Kocoń et al., 2023; Kuzman et al., 2023; Qin et al., 2023).

After that, we augmented the best performing prompt with two additional instructional elements, context of the task and CoT prompts (Reynolds and McDonell, 2021). For the context we include explanations and cue words of causal, correlational, and no relationship from Yu et al. (2020). We examined whether the location of context affects the performance by adding the context before and then after the BASE prompt. We added CoT to the end of the prompt, a usual design, by appending the phrase *"Answer (causal, correlational or no relationship) the question step by step"*, which was inspired from Zhong et al. (2023).

We also conducted additional tests to evaluate whether setting system message affects ChatGPT performance.

3.2 Evaluation Method

A semi-automatic approach was taken to post-process ChatGPT answers, since ChatGPT sometimes does not provide answers in the requested format or even provides invalid answers. We used a set of heuristic rules to map ChatGPT and GPT3.5 answers to the category labels. For instance, if "correlational" is in the answer, but not "causal", the label would be "correlational". See post-processing code in Appendix A Listing 1. Ambiguous answers that cannot be automatically mapped were manually examined and mapped. The number of invalid answers (# of unlabeled) was documented for each experiment. For prompts with CoT, the results were manually examined to verify whether the reasoning is valid.

After the post-processing, the macro f1-score is calculated to measure each model's performance, such as ChatGPT-BASE, against the human-annotated labels. Cohen's Kappa (Cohen, 1960) was also used to evaluate the agreements between different models and prompts. We conjecture that if a prompt shows consistently good performance across GPT3.5 and ChatGPT, the prompt has more robustness than other prompts that perform well on only one of them.

	Prompt
BASE	Read the following sentence - _____ Answer this question as concisely as possible - Does the sentence describe any causal or correlational research finding?
BASE+“conditional”	Read the following sentence - _____ Answer this question as concisely as possible - Does the sentence describe any direct causal, **conditional causal**, or correlational research finding?
BASE+“possible”	Read the following sentence - _____ Answer this question as concisely as possible - Does the sentence describe any direct causal, **possible direct causal**, or correlational research finding?
Huang et al. (2023)	Given Sentence: ‘_____’. Answer causal or correlational if the sentence describes any research finding. Answer as concisely as possible.
Kocoń et al. (2023)	Which of the attributes: “causal”, “correlational”, “no relationship” describe the research finding of a given text? Write your answer in the form of a Python list containing the appropriate attributes. Text: _____
Kuzman et al. (2023)	Please classify the following text describing a research finding and explain your decision. You can choose from the following classes: Causal, Correlational, No Relationship. The text to classify: _____
Qin et al. (2023)	For each snippet of text, label the research finding of the text as causal or correlational or no relationship. The answer should be exact ‘causal’ or ‘correlational’ or ‘no relationship’. Text: _____ Label:
element: **CoT**	Answer (causal, correlational or no relationship) the question step by step.
element: **Context**	Correlational: The statement describes the association between variables, but causation cannot be explicitly stated. Language Cue: association, associated with, predictor, at high risk of... Causal: The statement says that the independent variable directly alters the dependent variable. Language Cue: increase, decrease, lead to, effective in, contribute to, reduce, can... No relationship: The statement is not for current study findings or no correlation/causation relationship is mentioned in the statement.
system message	`You are a helpful assistant that can classify sentences as either causal or correlational research findings.`

Table 1: Different prompt designs

3.3 Data

We utilized two open-access cross-genre datasets that were manually annotated for science claim strength. The first dataset includes a sample of 3,061 research conclusion sentences from structured abstracts in PubMed articles (Yu et al., 2019). The second dataset consists of 2,076 sentences from health-related press releases on EurekAlert!, a major science press release platform (Yu et al., 2020). These sentences were either headlines or the first two sentences in press releases. Both datasets were manually annotated with the same four-category labels including correlational, direct causal, conditional causal, and no relationship.

To compare the effectiveness of different prompt designs, we created a sample subset from the PubMed dataset as the development set. In a prior study, Gutiérrez et al. (2022) sample 100 examples for prompt design selection. To ensure an equal representation of each class, we sampled 50 sentences from each class with a total of 200 sentences. The main reason for choosing a relatively small development set is the time cost for post-processing the ambiguous answers. After these experiments, we selected the best prompt design and evaluated it on the entire PubMed and EurekAlert! datasets. Since we are particularly interested in ChatGPT's understanding of conditional causal claims, we conducted two sets of experiments, one with conditional causal category and one without.

3.4 API

OpenAI released a public API for both GPT3.5 and ChatGPT model. For GPT3.5 experiments we use "text-davinci-003" model with temperature set as 0 and max tokens set as 50. The temperature is set to 0 as Gilardi et al. (2023) found that lower temperatures result in more consistent outcomes, ideal for annotation tasks.

For ChatGPT experiments we use "gpt-3.5-turbo" model. We input our prompt designs in the user message as `{"role": "user", "content": prompt}`. When testing the efficacy of system message, we add in the system message `{"role": "system", "content": system message}` prior to the user message.

4 Results

4.1 First Set of Experiments: RQ1 and RQ2

The first set of experiments included conditional causal examples. We started with a prompt based on the human annotation instruction (see "BASE+conditional" in Table 1). The result in Table 2 shows the macro f1-score at .486, much lower than the .881 macro f1-score from a fine-tuned BioBERT model in Yu et al. (2019). Among the four categories, ChatGPT severely underperformed in the conditional causal category with a low .164 macro f1, which prompted us for further investigation.

	No relationship	Direct causal	Conditional causal	Correlational	F1-score	Macro f1-score
No relationship	**29**	1	2	18	0.674	0.486
Direct causal	2	**19**	0	29	0.537	
Conditional causal	4	2	**6**	38	0.164	
Correlational	1	1	0	**48**	0.570	

Table 2: ChatGPT initial confusion matrix: row stands for predicted label and column stands for actual ground truth label

To understand more about how ChatGPT interprets the concept of "conditional causal", we asked ChatGPT *"What is 'conditional causal relationship'?"*. It responded "causal under certain conditions" (see full response in appendix A). We further examined its interpretation by adding CoT - *"Answer (direct causal, conditional causal, correlational, or no relationship) the question step by step."* to the prompt. Again, the response was "causal under certain conditions". These results suggest that the category label "conditional causal" is a misnomer, at least to ChatGPT.

We then attempted to look for an alternative label that would align better with ChatGPT's interpretation. A re-examination of the CoT responses showed that all answers used the given labels "direct causal" or "conditional causal", except for three answers, in which ChatGPT used the terms "possible direct causal", "potential direct causal", and "potential causal". ChatGPT's answer to the question *"What is 'possible direct causal relationship'?"* also showed a better match with the original definition of "conditional causal" (see full answer in appendix A).

We then hypothesized that "possible" or "potential" may be a better term than "conditional" for ChatGPT. We replaced "conditional" with "possible” in the prompt and repeated the evaluation (see “BASE+possible” in Table 1). The results in Table 3 show that the new prompt drastically improved ChatGPT’s performance: the f1-score for conditional causal increased from .164 to .578; the f1-scores for the other three categories were also improved slightly; the macro f1 increased from .486 to .631. However, ChatGPT’s performance, even with misnomer corrected in the instruction, still falls behind the fine-tuned BioBERT model (RQ1).

	No relationship	**Direct causal**	**Possible direct causal**	**Correlational**	**F1-score**	**Macro f1-score**
No relationship	**27**	7	8	8	0.675 ↑	0.631 ↑
Direct causal	0	**29**	19	2	0.624 ↑	
Possible direct causal	1	5	**37**	7	0.578 ↑	
Correlational	2	2	14	**32**	0.646 ↑	

Table 3: ChatGPT confusion matrix (revise conditional causal to possible direct causal): row stands for predicted label and column stands for actual ground truth label

To further probe how the one-word switch in the prompt affected ChatGPT’s interpretation of conditional causal relationship, we looked into its interpretation of hedges.

Conditional causal relationships are usually expressed by hedges. Actually, the 50 conditional causal examples were covered by six hedge words: “may” 32 times, “appear” 6, “could” 6, “might” 4, “seem” 3, and “unlikely” once. Note the total is 52 since two sentences used two hedge words.

As an example, we examined the six sentences that used “appear(s/ed)” as conditional causal cues. With the “BASE+conditional” prompt, ChatGPT recognized half of them as correlational and the other half as direct causal. With the “BASE+possible” prompt, ChatGPT recognized four as “possible direct causal” and two as “direct causal”. It is an improvement, but still not perfect.

Overall, our results provide evidence that ChatGPT has difficulty interpreting hedges in conditional causal claims, even after the prompt instruction was adjusted to match its own interpretation of this concept. This indicates that ChatGPT may have inherited the confusion or bias among human readers regarding conditional causal claims (RQ2).

Note that the misnomer has not been a problem for human annotators since they can adapt their interpretation based on the given definition on “conditional causal”, which was semantically equivalent to possible/speculative/qualified causal (Sumner et al., 2014). It is not a problem for fine-tuned BERT models either, since the models learned the concept from training data instead of the category definitions. Despite that, ChatGPT’s lack of adaptability may be utilized to design or refine human annotation guidelines to reduce potential misnomers.

4.2 Second Set of Experiments: RQ3 and RQ4

	GPT3.5 Macro f1-score	**ChatGPT Macro f1-score**	**Cohen Kappa**
BASE	0.494(7)	**0.743**(3)	0.491
Huang et al. (2023)	0.330	0.504	0.147
Kocoń et al. (2023)	0.514(6)	0.545(6)	0.478
Kuzman et al. (2023)	0.558	0.629	0.530
Qin et al. (2023)	**0.699**	0.735	0.665
BASE + CoT	0.538	**0.772**	0.462
context + BASE	0.695	0.744	0.618
context + BASE + CoT	**0.709**	0.684	0.504
BASE + context	-	0.763	-
BASE + context + CoT	0.364	0.419(1)	0.029
system message + (BASE + CoT)	-	0.726	-

Table 4: Prompt results on a sample of 150 PubMed data. The Cohen’s Kappa score is calculated between GPT3.5 and ChatGPT labels. The numbers in parenthesis are unlabeled examples due to invalid answers, such as *“‘causal or correlational research finding’ neither is mentioned in the sentence”*.

In the remaining experiments we excluded conditional causal examples, shifting focus on ChatGPT’s ability in distinguishing direct causal, correlational, or no relationship.

We first compared the performance scores for different prompt designs listed in Table 1. The results are reported in Table 4. Note that the unlabeled examples were excluded when calculating f1. For example, for the BASE prompt, a total of 10 unlabeled examples, 7 from GPT3.5 and 3 from ChatGPT, were excluded, so that the two f1-scores are comparable. When comparing results across prompts, since only three prompts had unlabeled examples, including two with performance at the lower end, the result comparisons below were minimally affected, except that the .743 macro f1 for ChatGPT-BASE should be interpreted with caution.

The first group of results (in rows 2-6) are GPT3.5 and ChatGPT performance with our own BASE prompt and four other prompts inspired from prior studies. The prompt from Qin et al. (2023) performed best with GPT3.5 with .699 macro f1-score, and our own BASE prompt performed best with ChatGPT at .743. We also calculated Cohen’s Kappa between GPT3.5 and ChatGPT results with the same prompts, and found that the agreements varied vastly from .147 for the prompt from Huang et al. (2023) to .665 from Qin et al. (2023).

Although our BASE prompt achieved the highest macro f1 .743 among the five prompts across GPT3.5 and ChatGPT, the prompt from Qin et al. (2023) shows consistently high performance across GPT3.5 and ChatGPT (.699 and .735 macro f1-scores) and highest inter-model agreement (.665 Cohen's Kappa), demonstrating strong robustness. In comparison, the BASE prompt used the format of a question, while the prompt from Qin et al. (2023) was formatted as a labeling task with stricter formatting instructions. Further studies are needed to examine what design features contributed to the performance differences (RQ3).

We then tested the impact of additional instructional elements, i.e. context and CoT (see results in rows 7-9). We observed a slight improvement in performance when separately incorporating CoT and context to the BASE prompt, resulting in a macro f1-score of .772 and .744 respectively.

Note that adding CoT to the prompt does not guarantee an answer with a reasoning process. We found that only 42% answers to the BASE + CoT prompt included the reasoning process. For the context + BASE + CoT prompt, the response rate increased to 85%. However, a higher CoT response rate did not translate to better performance. Instead, the macro f1-score decreased from .772 to .684.

Our finding that CoT improved ChatGPT performance on a zero-shot setting is consistent with prior literature (Kojima et al., 2022). However, it is worth noting that changing wording in CoT can also impact the results. In our experiments on the development set, we tested two variations of CoT. The first prompt was "BASE + Let's think step by step." This yielded a macro f1 score of .732. The second prompt was "BASE + Answer (causal, correlational or no relationship) the question step by step.", referred to as BASE+CoT in Table 4, which achieved a higher macro f1 score of .772.

We also examined whether the interpretations in CoT responses were faithful, which means ChatGPT's interpretation is consistent with its answer (Jacovi and Goldberg, 2020). After checking all responses in the ChatGPT-Base + CoT experiment, we found that all CoT interpretations were faithful. In other experiments, unfaithful interpretations were occasionally spotted but rare. For example, one answer included a 4-step reasoning process. Step 2 implied that the sentence has no relationship: *step 2: does the sentence describe the research finding as causal or correlational? no*. However, Step 4 changed the final answer to correlational: *step 4: therefore, the answer is "correlational"*.

We also tested if adding context before or after prompt would make any difference. The second and the third group of results in Table 4 show that it did not affect ChatGPT significantly, which performed slightly better with context after prompt (.763 macro f1 for BASE + context vs. .744 for context + BASE). However, stark contrast was observed with GPT3.5, which had a decent performance at .695 macro f1 for context + BASE; however, it failed to output any valid response when context was added after the BASE prompt, indicating that the context after the prompt distracted GPT3.5 away from completing the task. When further adding CoT after the context (i.e. BASE + context + CoT), GPT3.5 performance was still poor at .364 macro f1. Surprisingly, the BASE + context + CoT prompt also dragged ChatGPT performance down to .419. These results suggest that prompt design with additional instructional elements is not always "the more the merrier". The inconsistent performance between GPT3.5 and ChatGPT also indicates the uncertainty when experimenting with prompt engineering across LLMs.

Our last prompt engineering attempt was to add a system message to the best prompt for ChatGPT-BASE + CoT. It did not help as the performance was slightly decreased from .772 to .726.

In summary, context + BASE + CoT resulted in the best GPT3.5 performance prompt at .709 macro f1, and BASE + CoT resulted in the best ChatGPT performance at .772. For ChatGPT, adding CoT helped, but adding both context and CoT hurt. For GPT3.5, adding CoT helped, as well as adding context before prompt, but adding context after prompt distracted it (RQ4).

4.3 ChatGPT Results on Full Data Sets

After finding the best performing prompt on the development set, we applied it to the entire PubMed dataset and EurekAlert! dataset, still excluding the conditional causal examples. The distribution of sentences per label is shown in Table 5. Since ChatGPT consistently outperformed GPT3.5 in previous experiments, we proceeded to test with ChatGPT only. We also repeated the test once a day for three days (April 21-23, 2023) to check the consistency of results among different runs, since ChatGPT cannot guarantee result reproducibility.

	PubMed	**EurekAlert**
No relationship	1,353	486
Causal	494	568
Correlational	995	738
Total	2,842	1,972

Table 5: Dataset description for each label

Table 6 shows that the macro f1-scores for both datasets decreased from the best performance (.772) on the development set to the range of .695 to .698 for the PubMed dataset and the range of .628 to .638 for EurekAlert!. Overall the unlabeled examples are not a major issue with its ratios all below 0.5%. However, the results among the three runs disagreed to some extent, as measured by average Kappa values at .813 and .701 respectively, raising concerns for result reproducibility if used as an off-the-shelf text classification model.

	PubMed			**Eureka**		
	Macro f1-score	**# of unlabeled**	avg_k	**Macro f1-score**	**# of unlabeled**	avg_k
1st	0.698	5	0.813	0.628	2	0.701
2nd	0.695	14		0.638	6	
3rd	0.695	14		0.634	6	

Table 6: ChatGPT performance on entire PubMed dataset and EurekAlert! dataset for 3 days. The avg_k represents the average Cohen Kappa value.

4.4 Performance of Ensemble Models

Despite the promising performance of GPT3.5 and ChatGPT with various prompts, they are still relatively weak models with macro f1-scores below 0.8. The correlations among these models were also in low to mid range, as measured by Cohen's Kappa. For example, the Kappa values between GPT3.5 and ChatGPT range from .147 to .665 in Table 4. The Kappa values among the ChatGPT results with the five different manual prompts range from .199 to .656. These observations suggest the possibility of constructing an ensemble model through simple majority vote (Dietterich, 2000). Therefore we tried two ensemble models (1) combining five ChatGPT models with the five manual prompts, and (2) combining ten models, five from GPT3.5 and five from ChatGPT.

We used a straightforward majority voting approach to ensemble each model's outcomes. In case of a tie, we used a weighted voting approach that takes the macro f1-score of each model as the weight, favoring the better-performing models.

The result in Table 7 shows that the ensemble of five ChatGPT models with weighted tie-breaking resulted in .743 macro f1, which did not beat the .772 best performance with BASE + CoT. The ensemble of both GPT3.5 and ChatGPT models performed even worse, at .705 macro f1. In summary, the simple majority vote ensemble did not lead to a better-performing model.

	ChatGPT	**GPT3.5+ChatGPT**
No relationship	0.788	0.724
Causal	0.691	0.684
Correlational	0.748	0.707
Macro f1-score	0.743	0.705
# of ties	5	6

Table 7: Ensemble results. ChatGPT refers to an ensemble of ChatGPT models on five manual prompts and GPT3.5+ChatGPT refers to an ensemble of both GPT3.5 and ChatGPT.

5 Conclusion and Discussion

Causal language is an important rhetorical device in science communication. However, subjectivity in causal language use and understanding is a challenge for science writing and reading. Since ChatGPT captures latent social information to some extent, this study evaluated its ability to understand causal language in science papers and news by testing their accuracy in a task of claim strength classification. The results show that (1) ChatGPT is still behind the existing fine-tuned BERT models by a large margin; (2) ChatGPT seems to have inherited the confusion observed among average human readers when judging the strength of conditional causal claims that were mitigated by hedges; (3) ChatGPT performance varied substantially with semantically-similar prompts and across different model versions; (4) CoT responses were faithful and helpful. ChatGPT was able to reproduce its results at the level of 0.7-0.8 measured by Cohen's Kappa. However, the inconsistency in performance across model versions and semantically-similar prompts suggests caution when generalizing prompt engineering results across tasks and models.

While we were conducting our experiment, another study posted to arxiv (Chen et al., 2023) reported their ChatGPT evaluation on the PubMed data set. Both studies shared the findings that prompt engineering required significant investment and a slight difference in prompts could lead to substantial change in performance. Both studies on the same task found that CoT helped performance. While both zero-shot and few-shot settings were tested in Chen et al. (2023), we tested the zero-shot setting only. Our study has better performance under the zero-shot setting. Comparing the prompts, we hypothesize that explicitly asking the causal relationship in the prompt may have helped. However, a systematic method is still lacking to infer causality between word choices in prompts and the performance.

Despite our effort for a systematic review of ChatGPT's understanding of causal claims, our study design has some limitations as the evaluation methodology for prompt engineering is still under development in the NLP community. We arbitrarily decided on the size of the development set. Our study focused on zero-shot setting with the purpose of evaluating the latent understanding on causal claims within ChatGPT. Further exploration could be conducted to investigate the impact of few-shot settings by carefully selecting examples based on recent progress in few-shots prompting methods (Lu et al., 2022; Liu et al., 2022).

We conclude that ChatGPT has a promising but still limited ability in understanding causal language in science writing. CoTs improved prompt performance, but finding the optimal prompt is difficult with inconsistent results and the lack of effective methods to establish cause-effect between prompts and outcomes. Following instruction is an important prerequisite for using ChatGPT as a text classification tool, to avoid high labor cost for post-processing its answers. However, ChatGPT provides a new, simulation-style approach for designing and evaluating human annotation guidelines.

Acknowledgements

This research is supported by the US National Science Foundation under grant 1952353.

References

Rachel C Adams, Petroc Sumner, Solveiga Vivian-Griffiths, Amy Barrington, Andrew Williams, Jacky Boivin, Christopher D Chambers, and Lewis Bott. 2017. How readers understand causal and correlational expressions used in news headlines. *Journal of experimental psychology: applied*, 23(1):1.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Shan Chen, Yingya Li, Sheng Lu, Hoang Van, Hugo JWL Aerts, Guergana K Savova, and Danielle S Bitterman. 2023. Evaluation of chatgpt family of models for biomedical reasoning and classification. *arXiv preprint arXiv:2304.02496*.

Stacey S Cofield, Rachel V Corona, and David B Allison. 2010. Use of causal language in observational studies of obesity and nutrition. *Obesity facts*, 3(6):353–356.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46.

Thomas G Dietterich. 2000. Ensemble methods in machine learning. In *Multiple Classifier Systems: First International Workshop, MCS 2000 Cagliari, Italy, June 21–23, 2000 Proceedings 1*, pages 1–15. Springer.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd-workers for text-annotation tasks. *arXiv preprint arXiv:2303.15056*.

Bernal Jiménez Gutiérrez, Nikolas McNeal, Clay Washington, You Chen, Lang Li, Huan Sun, and Yu Su. 2022. Thinking about gpt-3 in-context learning for biomedical ie? think again. *arXiv preprint arXiv:2203.08410*.

Noah Haber, Emily R Smith, Ellen Moscoe, Kathryn Andrews, Robin Audy, Winnie Bell, Alana T Brennan, Alexander Breskin, Jeremy C Kane, Mahesh Karra, et al. 2018. Causal language and strength of inference in academic and media articles shared in social media (claims): A systematic review. *PloS one*, 13(5):e0196346.

John J Horton. 2023. Large language models as simulated economic agents: What can we learn from homo silicus? *arXiv preprint arXiv:2301.07543*.

Fan Huang, Haewoon Kwak, and Jisun An. 2023. Is chatgpt better than human annotators? potential and limitations of chatgpt in explaining implicit hate speech. *arXiv preprint arXiv:2302.07736*.

Alon Jacovi and Yoav Goldberg. 2020. Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness? *arXiv preprint arXiv:2004.03685*.

Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Jan Kocoń, Igor Cichecki, Oliwier Kaszyca, Mateusz Kochanek, Dominika Szydło, Joanna Baran, Julita Bielaniewicz, Marcin Gruza, Arkadiusz Janz, Kamil Kanclerz, et al. 2023. Chatgpt: Jack of all trades, master of none. *arXiv preprint arXiv:2302.10724*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *arXiv preprint arXiv:2205.11916*.

Taja Kuzman, Igor Mozetic, and Nikola Ljubešic. 2023. Chatgpt: Beginning of an end of manual linguistic data annotation? use case of automatic genre identification. *ArXiv, abs/2303.03953*.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Judea Pearl and Dana Mackenzie. 2018. *The book of why: the new science of cause and effect*. Basic books.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is chatgpt a general-purpose natural language processing task solver? *arXiv preprint arXiv:2302.06476*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–7.

Xinyue Shen, Zeyuan Chen, Michael Backes, and Yang Zhang. 2023. In chatgpt we trust? measuring and characterizing the reliability of chatgpt. *arXiv preprint arXiv:2304.08979*.

Petroc Sumner, Solveiga Vivian-Griffiths, Jacky Boivin, Andy Williams, Christos A Venetis, Aimée Davies, Jack Ogden, Leanne Whelan, Bethan Hughes, Bethan Dalton, et al. 2014. The association between exaggeration in health related science news and academic press releases: retrospective observational study. *Bmj*, 349.

Fiona Anting Tan, Devamanyu Hazarika, See Kiong Ng, Soujanya Poria, and Roger Zimmermann. 2021. Causal augmentation for causal sentence classification. In *Proceedings of the First Workshop on Causal Inference and NLP*, pages 1–20.

Zengzhi Wang, Qiming Xie, Zixiang Ding, Yi Feng, and Rui Xia. 2023. Is chatgpt a good sentiment analyzer? a preliminary study. *arXiv preprint arXiv:2304.04339*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Dustin Wright and Isabelle Augenstein. 2021. Semi-supervised exaggeration detection of health science press releases. *arXiv preprint arXiv:2108.13493*.

Seonghyeon Ye, Hyeonbin Hwang, Sohee Yang, Hyeongu Yun, Yireun Kim, and Minjoon Seo. 2023. In-context instruction learning. *arXiv e-prints*, pages arXiv–2302.

Bei Yu, Yingya Li, and Jun Wang. 2019. Detecting causal language use in science findings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4664–4674.

Bei Yu, Jun Wang, Lu Guo, and Yingya Li. 2020. Measuring correlation-to-causation exaggeration in press releases. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4860–4872.

Qihuang Zhong, Liang Ding, Juhua Liu, Bo Du, and Dacheng Tao. 2023. Can chatgpt understand too? a comparative study on chatgpt and fine-tuned bert. *arXiv preprint arXiv:2302.10198*.

A Appendix

ChatGPT’s response to the question “What is ‘conditional causal relationship’?”

A conditional causal relationship is a type of relationship between two variables, where one variable (the cause) has a causal effect on the other variable (the effect) under certain conditions. In other words, the causal effect of the cause variable is dependent on the presence or absence of a third variable, known as the "conditioning variable."

For example, let’s say that we are interested in studying the effect of a new medication (the cause) on blood pressure (the effect), but we know that the effect of the medication on blood pressure may depend on other factors such as age or body weight (the conditioning variables). In this case, we might say that there is a conditional causal relationship between the medication and blood pressure, where the causal effect of the medication on blood pressure is conditional on the values of the conditioning variables.

In summary, a conditional causal relationship is a relationship between two variables where the causal effect of one variable on the other is conditional on the presence or absence of a third variable.

ChatGPT’s response to the question “What is ‘potential causal relationship’?”

A potential causal relationship refers to the possibility that one variable or event may be responsible for causing another variable or event. It suggests that there may be a cause-and-effect relationship between two phenomena, but it does not necessarily prove that one caused the other.

Establishing a potential causal relationship typically involves observing an association between two variables and then conducting further research or analysis to determine whether there is a causal link between them. This can be challenging because there may be other factors that are responsible for the observed association, or there may be reverse causation (i.e., the effect is actually causing the cause).

In order to establish a true causal relationship, it is often necessary to use rigorous scientific methods such as randomized controlled trials or longitudinal studies, which can help to rule out alternative explanations and establish a causal relationship with greater certainty.

Listing 1: Post-processing code

```
def chatgpt_label(answer):
    if 'no relationship' in answer:
        return 'norel'
    if 'correlational' in answer and
         'causal' not in answer:
        return 'correlational'
    elif 'causal' in answer and
         'correlational' not in answer:
        return 'causal'
    elif 'causal' not in answer and
         'correlational' not in answer:
        if answer.startswith('no') or
             answer.startswith('unclear'):
            return 'norel'
        return '???'
    else:
        return '???'
```

Systematic Evaluation of GPT-3 for Zero-Shot Personality Estimation

Adithya V Ganesan[*] **Yash Kumar Lal**[*] **August Håkan Nilsson**
Stony Brook University Oslo Metropolitan University

H. Andrew Schwartz
Stony Brook University

{avirinchipur, ylal}@cs.stonybrook.edu

Abstract

Very large language models (LLMs) perform extremely well on a spectrum of NLP tasks in a zero-shot setting. However, little is known about their performance on human-level NLP problems which rely on understanding psychological concepts, such as assessing personality traits. In this work, we investigate the zero-shot ability of GPT-3 to estimate the Big 5 personality traits from users' social media posts. Through a set of systematic experiments, we find that zero-shot GPT-3 performance is somewhat close to an existing pre-trained SotA for broad classification upon injecting knowledge about the trait in the prompts. However, when prompted to provide fine-grained classification, its performance drops to close to a simple most frequent class (MFC) baseline. We further analyze where GPT-3 performs better, as well as worse, than a pretrained lexical model, illustrating systematic errors that suggest ways to improve LLMs on human-level NLP tasks. The code for this project is available on Github[1].

1 Introduction

Human-level NLP tasks, rooted in computational social science, focus on the link between social or psychological characteristics and language. Example tasks include personality assessment (Mairesse and Walker, 2006; Kulkarni et al., 2017; Lynn et al., 2020), demographic estimation (Sap et al., 2014; Preotiuc-Pietro and Ungar, 2018), and mental health-related tasks (Coppersmith et al., 2014; Guntuku et al., 2017; Matero et al., 2019). Although using LMs as embeddings or fine-tuning them for human-level NLP tasks is becoming popular (V Ganesan et al., 2021; Butala et al., 2021; Yang et al., 2021b), very little is known about zero-shot performance of LLMs on such tasks.

In this paper, we test the zero-shot performance of a popular LLM, GPT-3, to perform personality trait estimation. We focus on personality traits because they are considered the fundamental characteristics that distinguish people, persisting across cultures, demographics, and time (Costa and McCrae, 1992; Costa Jr and McCrae, 1996). These characteristics are useful for a wide range of social, economic, and clinical applications such as understanding psychological disorders (Khan et al., 2005), choosing content for learning styles (Komarraju et al., 2011) or occupations (Kern et al., 2019), and delivering personalized treatments for mental health issues (Bagby et al., 2016). Focusing on zero-shot evaluation of GPT-3 on these fundamental characteristics forms a strong benchmark for understanding how much and what dimensions of traits GPT-3 encodes out-of-the-box. Further, while fine-tuned LMs have only had mixed success beyond lexical approaches (Lynn et al., 2020; Kerz et al., 2022), using zero-shot capable LLMs could help lead to better estimates.

The NLP community has a growing interest in understanding the capabilities and failure modes of LLMs (Wei et al., 2022a; Yang et al., 2021c), and we explore questions that surround LLMs in the context of fundamental human traits of personality. Zero-shot performance can depend heavily on the explicit information infused in the prompt (Lal et al., 2022). Personality, defined by information in its well-established questionnaire tests, presents new opportunities for information infusion.

Our **contributions** address: **(1)** what information about personality is useful for GPT-3, **(2)** how its performance compares to current SotA, **(3)** the relation between ordinality of outcome labels with performance and **(4)** whether GPT-3 predictions stay consistent given similar external knowledge.

2 Background

Psychological traits are stable individual characteristics associated with behaviors, attitudes, feelings, and habits (APA, 2023). The "Big 5" is a popu-

---

[*] These authors contributed equally

[1] github.com/humanlab/gpt3-personality-estimation

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 390–400
July 14, 2023 ©2023 Association for Computational Linguistics

lar personality model that breaks characteristics into five fundamental dimensions, validated across hundreds of studies across cultures, demographics, and time (Costa and McCrae, 1992; McCrae and John, 1992). The approach is rooted in the *lexical hypothesis* that the most important traits must be encoded in language (Goldberg, 1990). We investigate all five factors from this model: openness to experience (OPE– intellectual, imaginative and open-minded), conscientiousness (CON– careful, thorough and organized), extraversion (EXT– energized by social and interpersonal interactions), agreeableness (AGR– friendly, good natured, conflict avoidant) and neuroticism (NEU– less secure, anxious, and depressive).

LLMs like PaLM (Chowdhery et al., 2022) have shown significant improvement in performance on various NLP tasks (Wei et al., 2022b; Suzgun et al., 2022), even without finetuning. There is a growing body of work investigating one of the ubiquitous LLMs, GPT-3, under different settings (Wei et al., 2022a; Shi et al., 2022; Bommarito et al., 2023). Inspired by this, we systematically study the ability of GPT-3 to perform personality assessment under zero-shot setting. Following evidence that incorporating knowledge about the task can improve performance (Vu et al., 2020; Yang et al., 2021b; Lal et al., 2022), we evaluate the impact of three different types of knowledge to determine which type improves personality estimation.

Modeling personality traits through natural language has been extensively studied using a wide range of approaches, from simple count-based models (Pennebaker and Stone, 2003; Golbeck et al., 2011) to complex hierarchical neural networks (Read et al., 2010; Yang et al., 2021a). Finetuning LMs has become the mainstream approach for this task only recently (V Ganesan et al., 2021). With the advent of GPT-3, zero- or few-shot settings have become the primary approach to leverage LLMs in other NLP applications, but are yet untested for personality estimation.

3 Dataset

To get a sample of language associated with personality, we followed the paradigm set forth in Jose et al. (2022) whereby consenting participants shared their own Facebook posts along with taking a battery of psychological assessments, including the big five personality test (Donnellan et al., 2006; Kosinski et al., 2013). The dataset comprises of 202 participants with outcomes of interests who had also shared their Facebook posts. First, we filter the data to only include user posts from the last year of data collection (Eichstaedt et al., 2018). Next, we only retain users for whom we have exactly 20 Facebook posts, similar to the approach described in other human-level NLP works (Lynn et al., 2020; Matero et al., 2021). Finally, we anonymize the data by replacing personable identifiable information using SciPy's (Virtanen et al., 2020) NER model. We also remove phone numbers and email IDs using regular expressions. Finally, we are left with anonymized Facebook posts for 142 users and their associated 5 personality traits. This population (all from US) has a gender ratio of 79:18:3 (female:male:others). The age ranged from 21 to 66 (median=37). The big 5 personality trait scores fall in the continuous range of [1, 5]. We discretize the outcome values into the desired number of bins/classes using a quantile discretizer (in Pandas). We explain why we choose to discretize the outcome values in §4.

4 Experimental Design

In this work, GPT-3 is evaluated in a zero-shot setting. We frame the problem of personality prediction as classifying the degree (i.e. high/low or high/medium/low) to which a person exhibits a trait. Ideally, because the big 5 are considered continuously valued variables (McCrae and Costa Jr, 1989), one would model as a regression task, but we found this simplification to classification necessary to get any meaningful insights from GPT-3's zero-shot capability. We also investigate the degradation of performance for tertiary classification instead of binary in §5.

We devise a simple, reasonable prompt (BASIC)[2] to first estimate the ability of GPT-3 to predict the Big 5 personality traits. Building on this, we investigate whether adding external knowledge about these traits helps the model perform better. We use three types of knowledge: **(1) TEXTBOOK**: a concise definition of these traits from Roccas et al. (2002), **(2) WORDLIST**: frequent and infrequent words[3] used by people exhibiting those traits, and **(3) ITEMDESC**: survey items[4] (a positive and a negative) users responded to, based on which their personality scores were estimated.

---

[2] Examples of all prompts are in Appendix Figure 2.

[3] We use the wordlist from Schwartz et al. (2013).

[4] See Appendix Table 7 for detailed item descriptions.

Model	OPE	CON	EXT	AGR	NEU	Avg
Benchmarks						
MFC	0.352	0.427	0.411	0.372	0.333	0.379
WT-LEX (Park et al.)	**0.492**	0.393	0.516	**0.609**	**0.578**	**0.518**
Zero-Shot GPT-3						
BASIC	0.329†	0.385	0.521	0.435‡	0.333‡	0.400
TEXTBOOK	0.328†	0.401	0.496	0.506*	0.364‡	0.419
WORDLIST	0.366†	0.457	0.445	0.544	0.393‡	0.441
ITEMDESC	0.342†	**0.521**†	**0.569**	0.488†	0.349‡	0.454

Table 1: MACRO F1 scores for different kinds of knowledge added to the prompt. TEXTBOOK refers to adding the definition of the trait as described in Roccas et al. (2002), WORDLIST refers to adding the top 5 positively and negatively correlated unigrams with the trait reported by Schwartz et al. (2013), ITEMDESC refers to adding the items that were a part of the personality questionnaire (Table 7). WT-LEX refers to the SotA model described in §4. The findings indicate a statistically significant distinction when compared to the WT-LEX model, with significance levels of $p < 0.05$ (*), $p < 0.01$ (†), and $p < 0.001$ (‡).

Baseline and Evaluation. The baseline, WT-LEX, is a ridge regression model from Park et al. 2015 trained on dimensionally reduced feature set of n-grams and LDA-based topics extracted from Kosinski et al. (2013) Facebook data. The number of parameters in this model is orders of magnitude less than GPT-3. Even complex neural models (Lynn et al., 2020) have been unsuccessful to surpass its performance. WT-LEX also produces predictions in the continuous scale within the range of [1, 5]. In order to make a fair comparison with GPT-3, we perform the quantile discretization described in §3 and calculate MACRO F1. We evaluate the predictions using macro F1 scores.

5 Results

Table 1 shows GPT-3's performance on different personality traits, with and without knowledge. We find that ITEMDESC prompts the best performance with GPT-3 on average. Surprisingly, the model is able to directly use survey items (ITEMDESC) to predict EXT and CON the best. Utilizing these is hard since it requires relating abstract concepts described in these survey items to the ecological language in the posts. The top frequent and infrequent words (WORDLIST) help model perform the most on AGR, OPE and NEU. We hypothesize that simple, lexical cues are more helpful here since it is easier to draw relations from the surface form in posts. We also note that estimating NEU is difficult for the model, which also is difficult for humans to estimate in zero-acquaintance contexts, (Kenny, 1994), including estimating neuroticism from Facebook profiles. Overall, GPT-3's predictions are heavily biased towards predicting individuals to be high openness and low in neuroticism.

We also tried incorporating all types of knowledge into a prompt and found that performance dropped below BASIC. However, combining knowledge types involves non-trivial decisions such as the order of knowledge types and its composition. We leave this to future work.

Using ITEMDESC, we establish the best possible GPT-3 performance for personality estimation. Although GPT-3's average performance over all traits is still lower than WT-LEX, it outperforms the MFC baseline. Prior work (V Ganesan et al., 2022; Matero et al., 2022) has shown dimensions of mental health constructs and personality traits being captured through language use patterns in LMs. GPT-3's performance in zero-shot setting provides reasonable evidence to believe that language patterns associated with these traits are encoded in its embedding space as well.

6 Analysis

To better understand the utility of GPT-3 for personality estimation, we analyze the effect of **(1)** problem framing, and **(2)** effect of survey items. Furthermore, we perform error analysis of GPT-3 to suggest avenues for improvement.

Problem Framing. When personality estimation is framed as a binary classification, GPT-3 is worse than SoTA on average in a zero-shot setting. Upon looking closer, we note that it is the best model for 2 out of the 5 traits. However, these observations are made in a simplified two-class setting, whereas the big 5 personality model produces a real valued outcome. In order to assess GPT-3's practical viability, we prompt it (ItemDesc) to provide more fine-grained predictions by presenting trait estimation as a three-class classification problem.

# class	OPE	CON	EXT	AGR	NEU	Avg
2	0.342	0.521	0.569	0.488	0.349	0.454
3	0.141	0.288	0.240	0.160	0.320	0.230

Table 2: Macro F1 scores of classifying the outcomes into varying number of classes using GPT-3. We find a sharp drop in performance on increasing the number of classes from 2 to 3. Hence, framing personality estimation as a binary classification is the simplest for GPT-3

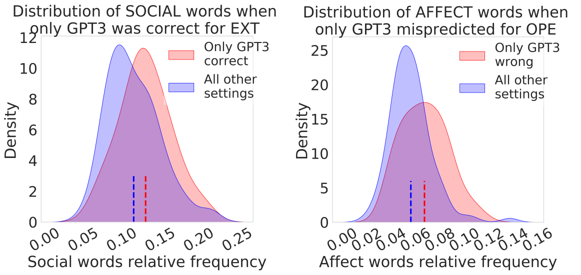
Table 2 shows that problem framing has a major impact on GPT-3 performance for all traits. Three class framing of the problem is harder than the binary framing which is evident from GPT-3's drop in performance (0.229) to close to MFC (0.212). This trend indicates that GPT-3 is ineffective in performing more fine-grained prediction tasks and consequently regression, which is the natural way to estimate the Big 5 traits. Clearly, GPT-3 is yet unsuited for fine-grained personality estimation.

Consistency with Survey Items. The standard questionnaire used to create the dataset had a total of 4 survey items per trait (2 positive and 2 negative). For ItemDesc, we use one positive and one negative item to describe each trait (see Figure 2). To investigate whether GPT-3 performance can be attributed to specific items in the prompt, we perform ItemDesc with all possible combinations of a positive and a negative survey item for all traits.

	Avg
ItemDesc	0.454
BothAltItems	0.448
AltPos	0.430
AltNeg	0.448

Table 3: Macro F1 scores for different pairs of positive and negative survey items combinations. Table 7 in Appendix contains the survey items that correspond to these four combination labels.

Table 3 shows that there is no meaningful difference in performance when provided different item combinations. This shows that GPT-3 is not sensitive to the items of the personality questionnaire. This is in line with data in Table 8, which shows that factor loading values (Fabrigar and Wegener, 2011) of these item combinations have similar powers to distinguish the corresponding traits.



(a) (b)

Figure 1: (a) SOCIAL words distributions compared for GPT-3 and Wt-Lex under two prediction settings: (1) only GPT-3 correct, and Wt-Lex incorrect, and (2) both models correct or GPT-3 incorrect. (right) AFFECT words distributions compared for GPT-3 and Wt-Lex under two prediction settings: (1) only GPT-3 incorrect, and Wt-Lex correct, and (2) both models incorrect or GPT-3 correct.

Error Analysis. Finally, we examine the linguistic variables that account for the errors in GPT-3 and the areas where it excels as compared to a traditional, lexical-based technique Wt-Lex. Figure 1a shows the distributions of SOCIAL words (Tausczik and Pennebaker, 2010) between users that were correctly predicted by only GPT-3 and the users that were either misclassfied by GPT-3 or correctly predicted by Wt-Lex for EXT task. SOCIAL words are better captured by LLMs probably owing to its ability to produce contextualized embeddings. Figure 1b depicts the distributions of AFFECT words between the users that were misclassified only by GPT-3 and the users that were either correctly classified by GPT-3 or Wt-Lex misclassifies for OPE task[5].

7 Conclusion

We performed a systematic investigation of GPT-3's zero-shot performance on personality estimation. While using a simple prompt did not yield strong performance, injecting knowledge about the traits themselves led to significant improvement. Even so, it falls short of using a strong, extensively-trained, supervised model (Wt-Lex). Further, we find that it is much harder for GPT-3 to provide more fine-grained predictions (when asked to select between 3 labels instead of 2), suggesting that LLMs may not be as capable at making dimen-

---

[5] We also looked at the differences in other LIWC categories for EXT and OPE tasks measured using Cohen's d (Diener, 2010) and logs odds ratio with informative dirichlet prior (Monroe et al., 2008) that offers more explanations for the errors and correctness of GPT-3 in Appendix C.

sional estimates about personality. Our systematic investigation helps understand GPT-3's zero-shot capabilities for a human-level NLP task, contextualizing its failure modes and showing avenues for LLM improvements.

Ethics Statement

Our work seeks to advance interdisciplinary NLP-psychology research for understanding human attributes associated with language. This research is intended to inform Computational Social Science researchers about the ability of LLMs to estimate psychological rating scales as well as for LLM researchers to understand types of psychological information that LMs capture. We intend for our work on personality trait assessments to have an impact on social, NLP, and clinical use cases to improve the well-being of people. We strongly condemn malevolent adoption of these technologies for targeted advertising, directed misinformation campaigns, and other malicious acts that could have potential harms on mental health.

If used for clinical practice, we strongly recommend that any use of LLM-based personality estimates be overseen by clinical psychology experts. During trials, models should be extensively tested for their failure mode rates (e.g. False-positive vs False-negative rates), and error disparities (Shah et al., 2020).

This interdisciplinary computer science, psychology, and health study had extensive privacy & ethical human subjects research protocols. All procedures were approved by an academic institutional review board. All contributors are certified to perform human subject research, and took steps and precautions while collecting and analyzing data to keep participants protected. The Facebook posts shared by consenting users were anonymized as described in §3 to prevent the participants from being identified.

Limitations

The Big 5 personality trait model measures the fundamental dimensions of human on a continuous scale. This real valued representation preserves more information and is more descriptive of inter-individual differences. While we acknowledge that the binary classification of Big 5 traits fails the purpose of the model, it is a necessary simplification to understand the ability of LLMs to perform personality assessment. Our investigation shows potential to improve the practical utility of LLMs in personality estimation.

Despite the strong results from existing works in support of in-context learning and larger message history for better performance, we were limited by the significant multiplicative cost these experiments entailed, as the GPT-3 API is billed based on token usage. Further, since each user's post history is typically long, it is infeasible to experiment with all in-context learning options due to GPT-3's context window size limitation. This is worthy of exploration, to understand the sample efficiency of GPT-3 and the impact of post history on its performance.

Acknowledgement

This work wouldn't have been possible without the support of AVG's and YKL's dear friends, Swanie Juhng, Aakanksha Rajiv Kapoor, Aravind Parthasarathy, Aditya Krishna, Akshay Bharadhwaj, and Somadutta Bhatta, who provided OpenAI API keys for running the experiments. We would also like to extend our gratitude to Matthew Matero, Niranjan Balasubramanian, Harsh Trivedi and Sid Mangalik for providing valuable feedback. AVG, AHN, and HAS were supported in part by NIH grant R01-AA028032. YKL was supported in part by the Air Force Research Laboratory (AFRL), DARPA, for the KAIROS program under agreement number FA8750-19-2-1003 and in part by the NSF under the award IIS #2007290.

References

American Psychological Association APA. 2023. Psychology.

R Michael Bagby, Tara M Gralnick, Nadia Al-Dajani, and Amanda A Uliaszek. 2016. The role of the five-factor model in personality assessment and treatment planning. *Clinical Psychology: Science and Practice*, 23(4):365.

Jillian Bommarito, Michael Bommarito, Daniel Martin Katz, and Jessica Katz. 2023. Gpt as knowledge worker: A zero-shot evaluation of (ai) cpa capabilities. *arXiv preprint arXiv:2301.04408*.

Laura Burdick, Jonathan K. Kummerfeld, and Rada Mihalcea. 2022. Using paraphrases to study properties of contextual embeddings. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4558–4568, Seattle, United States. Association for Computational Linguistics.

Yash Butala, Kanishk Singh, Adarsh Kumar, and Shrey Shrivastava. 2021. Team phoenix at WASSA 2021: Emotion analysis on news stories with pre-trained language models. In *Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 274–280, Online. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Glen Coppersmith, Mark Dredze, and Craig Harman. 2014. Quantifying mental health signals in Twitter. In *Proceedings of the Workshop on Computational Linguistics and Clinical Psychology: From Linguistic Signal to Clinical Reality*, pages 51–60, Baltimore, Maryland, USA. Association for Computational Linguistics.

Paul T Costa and Robert R McCrae. 1992. Normal personality assessment in clinical practice: The neo personality inventory. *Psychological assessment*, 4(1):5.

Paul T Costa Jr and Robert R McCrae. 1996. Mood and personality in adulthood. In *Handbook of emotion, adult development, and aging*, pages 369–383. Elsevier.

Marc J. Diener. 2010. *Cohen's d*, pages 1–1. John Wiley and Sons, Ltd.

M Brent Donnellan, Frederick L Oswald, Brendan M Baird, and Richard E Lucas. 2006. The mini-ipip scales: tiny-yet-effective measures of the big five factors of personality. *Psychological assessment*, 18(2):192.

Johannes C Eichstaedt, Robert J Smith, Raina M Merchant, Lyle H Ungar, Patrick Crutchley, Daniel Preoţiuc-Pietro, David A Asch, and H Andrew Schwartz. 2018. Facebook language predicts depression in medical records. *Proceedings of the National Academy of Sciences*, 115(44):11203–11208.

Leandre R Fabrigar and Duane T Wegener. 2011. *Exploratory factor analysis*. Oxford University Press.

Jennifer Golbeck, Cristina Robles, and Karen Turner. 2011. Predicting personality with social media. *CHI '11 Extended Abstracts on Human Factors in Computing Systems*.

Lewis R Goldberg. 1990. An alternative" description of personality": the big-five factor structure. *Journal of personality and social psychology*, 59(6):1216.

Sharath Chandra Guntuku, David Bryce Yaden, Margaret L. Kern, Lyle H. Ungar, and Johannes C. Eichstaedt. 2017. Detecting depression and mental illness on social media: an integrative review. *Current Opinion in Behavioral Sciences*, 18:43–49.

R Jose, M Matero, G Sherman, B Curtis, S Giorgi, HA Schwartz, and LH Ungar. 2022. Using facebook language to predict and describe excessive alcohol use. *Alcoholism, Clinical and Experimental Research*, 46(5):836–847.

David A Kenny. 1994. *Interpersonal perception: A social relations analysis*. Guilford Press.

Margaret L Kern, Paul X McCarthy, Deepanjan Chakrabarty, and Marian-Andrei Rizoiu. 2019. Social media-predicted personality traits and values can help match people to their ideal jobs. *Proceedings of the National Academy of Sciences*, 116(52):26459–26464.

Elma Kerz, Yu Qiao, Sourabh Zanwar, and Daniel Wiechmann. 2022. Pushing on personality detection from verbal behavior: A transformer meets text contours of psycholinguistic features. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 182–194, Dublin, Ireland. Association for Computational Linguistics.

Amir A Khan, Kristen C Jacobson, Charles O Gardner, Carol A Prescott, and Kenneth S Kendler. 2005. Personality and comorbidity of common psychiatric disorders. *The British Journal of Psychiatry*, 186(3):190–196.

Meera Komarraju, Steven J Karau, Ronald R Schmeck, and Alen Avdic. 2011. The big five personality traits, learning styles, and academic achievement. *Personality and individual differences*, 51(4):472–477.

M. Kosinski, D. Stillwell, and T. Graepel. 2013. Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the National Academy of Sciences*, 110:5802 – 5805.

Vivek Kulkarni, Margaret L. Kern, David Stillwell, Michal Kosinski, Sandra C. Matz, Lyle H. Ungar, Steven Skiena, and H. A. Schwartz. 2017. Latent human traits in the language of social media: An open-vocabulary approach. *PLoS ONE*, 13.

Yash Kumar Lal, Niket Tandon, Tanvi Aggarwal, Horace Liu, Nathanael Chambers, Raymond Mooney, and Niranjan Balasubramanian. 2022. Using commonsense knowledge to answer why-questions. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1204–1219, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Veronica Lynn, Niranjan Balasubramanian, and H. Andrew Schwartz. 2020. Hierarchical modeling for user personality prediction: The role of message-level attention. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5306–5316, Online. Association for Computational Linguistics.

François Mairesse and Marilyn Walker. 2006. Automatic recognition of personality in conversation. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 85–88, New York City, USA. Association for Computational Linguistics.

Matthew Matero, Albert Hung, and H. Andrew Schwartz. 2022. Evaluating contextual embeddings and their extraction layers for depression assessment. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 89–94, Dublin, Ireland. Association for Computational Linguistics.

Matthew Matero, Akash Idnani, Youngseo Son, Salvatore Giorgi, Huy Vu, Mohammad Zamani, Parth Limbachiya, Sharath Chandra Guntuku, and H. Andrew Schwartz. 2019. Suicide risk assessment with multi-level dual-context language and BERT. In *Proceedings of the Sixth Workshop on Computational Linguistics and Clinical Psychology*, pages 39–44, Minneapolis, Minnesota. Association for Computational Linguistics.

Matthew Matero, Nikita Soni, Niranjan Balasubramanian, and H. Andrew Schwartz. 2021. MeLT: Message-level transformer with masked document representations as pre-training for stance detection. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2959–2966, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Robert R McCrae and Paul T Costa Jr. 1989. Reinterpreting the myers-briggs type indicator from the perspective of the five-factor model of personality. *Journal of personality*, 57(1):17–40.

Robert R McCrae and Oliver P John. 1992. An introduction to the five-factor model and its applications. *Journal of personality*, 60(2):175–215.

Burt L Monroe, Michael P Colaresi, and Kevin M Quinn. 2008. Fightin'words: Lexical feature selection and evaluation for identifying the content of political conflict. *Political Analysis*, 16(4):372–403.

Gregory Park, H. A. Schwartz, J. Eichstaedt, M. Kern, M. Kosinski, D. Stillwell, Lyle H. Ungar, and M. Seligman. 2015. Automatic personality assessment through social media language. *Journal of personality and social psychology*, 108 6:934–52.

James W Pennebaker and Lori D Stone. 2003. Words of wisdom: language use over the life span. *Journal of personality and social psychology*, 85(2):291.

Daniel Preotiuc-Pietro and Lyle H. Ungar. 2018. User-level race and ethnicity predictors from twitter text. In *International Conference on Computational Linguistics*.

Stephen J Read, Brian M Monroe, Aaron L Brownstein, Yu Yang, Gurveen Chopra, and Lynn C Miller. 2010. A neural network model of the structure and dynamics of human personality. *Psychological review*, 117(1):61.

Sonia Roccas, Lilach Sagiv, Shalom H. Schwartz, and Ariel Knafo. 2002. The big five personality factors and personal values. *Personality and Social Psychology Bulletin*, 28(6):789–801.

Maarten Sap, Gregory Park, Johannes Eichstaedt, Margaret Kern, David Stillwell, Michal Kosinski, Lyle Ungar, and Hansen Andrew Schwartz. 2014. Developing age and gender predictive lexica over social media. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1146–1151, Doha, Qatar. Association for Computational Linguistics.

H Andrew Schwartz, Johannes C Eichstaedt, Margaret L Kern, Lukasz Dziurzynski, Stephanie M Ramones, Megha Agrawal, Achal Shah, Michal Kosinski, David Stillwell, Martin EP Seligman, et al. 2013. Personality, gender, and age in the language of social media: The open-vocabulary approach. *PloS one*, 8(9):e73791.

Deven Santosh Shah, H. Andrew Schwartz, and Dirk Hovy. 2020. Predictive biases in natural language processing models: A conceptual framework and overview. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5248–5264, Online. Association for Computational Linguistics.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, et al. 2022. Language models are multilingual chain-of-thought reasoners. *arXiv preprint arXiv:2210.03057*.

Mirac Suzgun, Nathan Scales, Nathanael Scharli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed Huai hsin Chi, Denny Zhou, and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *ArXiv*, abs/2210.09261.

Y. Tausczik and J. Pennebaker. 2010. The psychological meaning of words: Liwc and computerized text analysis methods. *Journal of Language and Social Psychology*, 29:24 – 54.

Adithya V Ganesan, Matthew Matero, Aravind Reddy Ravula, Huy Vu, and H. Andrew Schwartz. 2021. Empirical evaluation of pre-trained transformers for human-level NLP: The role of sample size and dimensionality. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4515–4532, Online. Association for Computational Linguistics.

Adithya V Ganesan, Vasudha Varadarajan, Juhi Mittal, Shashanka Subrahmanya, Matthew Matero, Nikita Soni, Sharath Chandra Guntuku, Johannes Eichstaedt, and H. Andrew Schwartz. 2022. WWBP-SQT-lite:

Multi-level models and difference embeddings for moments of change identification in mental health forums. In *Proceedings of the Eighth Workshop on Computational Linguistics and Clinical Psychology*, pages 251–258, Seattle, USA. Association for Computational Linguistics.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.

Huy Vu, Suhaib Abdurahman, Sudeep Bhatia, and Lyle Ungar. 2020. Predicting responses to psychological questionnaires from participants' social media posts and question text embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1512–1524, Online. Association for Computational Linguistics.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022a. Emergent abilities of large language models. *Transactions on Machine Learning Research*. Survey Certification.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022b. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Feifan Yang, Xiaojun Quan, Yunyi Yang, and Jianxing Yu. 2021a. Multi-document transformer for personality detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14221–14229.

Feifan Yang, Tao Yang, Xiaojun Quan, and Qinliang Su. 2021b. Learning to answer psychological questionnaire for personality detection. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1131–1142, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Yumao Lu, Zicheng Liu, and Lijuan Wang. 2021c. An empirical study of gpt-3 for few-shot knowledge-based vqa. In *AAAI Conference on Artificial Intelligence*.

Tal Yarkoni. 2010. Personality in 100,000 words: A large-scale analysis of personality and word use among bloggers. *Journal of research in personality*, 44(3):363–373.

A GPT-3

A.1 GPT-3 settings

We used a temperature of 0.0 for all the experiments to select the most likely token at each step, as this setting allows for reproducibility.

```
response = openai.Completion.create(
            model="text-davinci-003",
            prompt=prompt,
            temperature=0,
            max_tokens=1,
            top_p=1.0,
            frequency_penalty=0.1,
            presence_penalty=0.0
)
```

We restricted the model outputs to just one token. Only “Yes" or “No" are considered valid answers for our binary classification task. For the 3-class classification, “High", “Medium" and “Low" are considered valid answers.

For one data point in the WORDLIST EXT experiment, the model output was a newline character instead of Yes/No. By adding another newline to the prompt, we were able to get it to generate an answer (in this case, No). For one data point in the BASIC OPE experiment, the model output contained irrelevant tokens instead of High/Medium/Low. By adding another 2 newlines to the prompt, we were able to get it to generate an answer (in this case, High).

A.2 Prompt Design

For our binary classification task, we used the following prompt template:

```
Read the stream of Facebook posts from a
user below. Each newline represents a new
post. The posts are in order of date, the
last one is the most recent.
{messages}
{knowledge} Given these messages from a
user, is this user {trait} according to
the Big 5 personality traits? Select
between yes or no
```

A user’s posts are concatenated with the most recent post presented at the end to fill the *messages* field. Options for *trait* are agreeable, extraverted, open to experiences, neurotic, and conscientious.

For our 3-class problem framing, we used the following prompt template:

```
Read the stream of Facebook posts from a
user below. Each newline represents a new
post. The posts are in order of date, the
last one is the most recent.
{messages}
{knowledge} Given these messages from a
user, rate their {trait}. The options on
the scale are low, medium, high.
{trait}:
```

Options for *trait* are agreeableness, extraversion, openness to experiences, neuroticism, and conscientiousness. The different types of knowledge injected into the prompt for each personlity trait can be found in Figure 2.

B Glossary

We include the survey items from the questionnaires used in the study to collect data from consenting users along with their associated personality trait in Table 7, as well as the categories of language from the LIWC error analysis model in Table 4.

Category Abbrev	Category	Examples
NUMBER	Numbers	second, thousand
SOCIAL	Social Processes	mate, talk, they
AFFILIATION	Affiliation	ally, friend, social
YOU	2nd Person	you, your, thou
TIME	Time	end, until, season
FAMILY	Family	daughter, dad, aunt
PPRON	Personal Pronoun	I, them, her
POSEMO	Positive Emotion	love, nice, sweet
AFFECT	Affective Processes	happy, cried
FRIEND	Friends	neighbor, buddy
THEY	3rd Person plural	they, their, they’d
FOCUSPAST	Past Focus	ago, did, talked
ACHIEVE	Achievement	success, win, better
SHEHE	3rd person singular	she, him, her
NEGATE	Negation	not, never, no
PRONOUN	Total Pronouns	I, them, itself

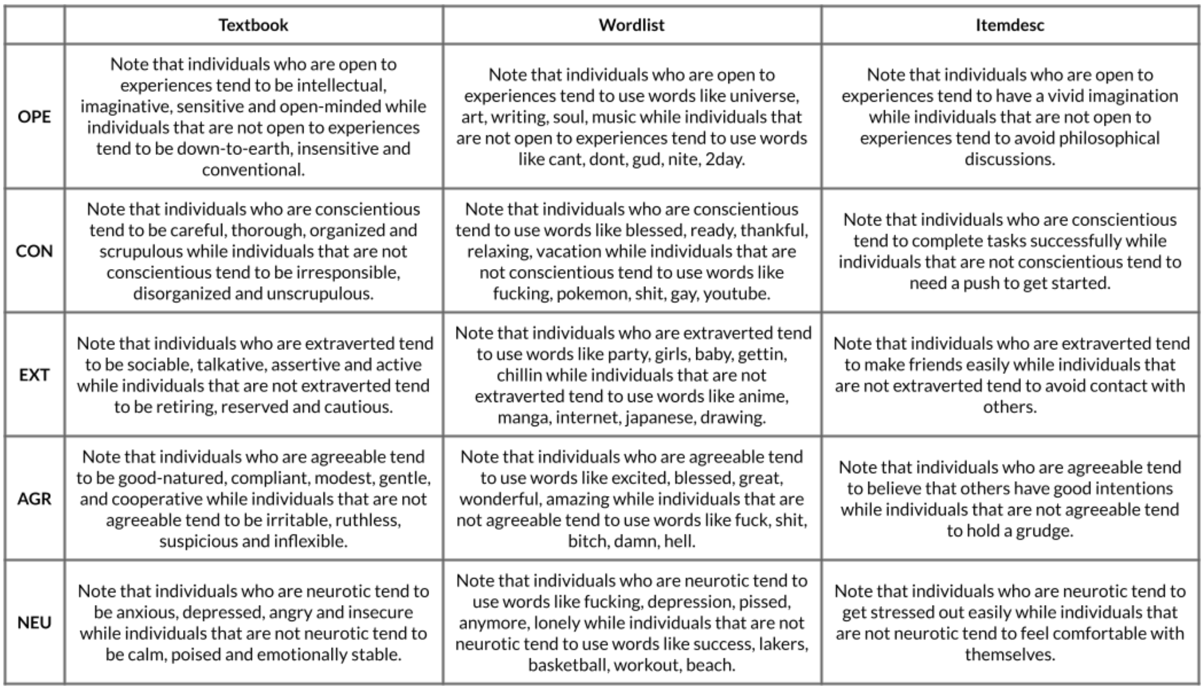
Table 4: LIWC glossary to map the category abbreviation with its full form and a few examples for each row.

C Error Analysis

We examine where GPT-3 differs from WT-LEX: **(1)** performing better on EXT in Table 5, and **(2)** predicting OPE worse in Table 6. Results from Table 5 suggest that GPT-3 encodes language categories[6] (Tausczik and Pennebaker, 2010) highly predictive of EXT such as social processes (SOCIAL), group identification (AFFILIATION), and use of second person pronoun (YOU), all of which have been shown to have strong significant association with this trait (Schwartz et al., 2013). GPT-3 can disambiguate common social lexicons occurring in different contexts (Burdick et al., 2022) (e.g., "party" in the context of gathering vs political ideology), which count-based lexical models can’t do.

---

[6] See Table 4 for details on LIWC categories



	Textbook	Wordlist	Itemdesc
OPE	Note that individuals who are open to experiences tend to be intellectual, imaginative, sensitive and open-minded while individuals that are not open to experiences tend to be down-to-earth, insensitive and conventional.	Note that individuals who are open to experiences tend to use words like universe, art, writing, soul, music while individuals that are not open to experiences tend to use words like cant, dont, gud, nite, 2day.	Note that individuals who are open to experiences tend to have a vivid imagination while individuals that are not open to experiences tend to avoid philosophical discussions.
CON	Note that individuals who are conscientious tend to be careful, thorough, organized and scrupulous while individuals that are not conscientious tend to be irresponsible, disorganized and unscrupulous.	Note that individuals who are conscientious tend to use words like blessed, ready, thankful, relaxing, vacation while individuals that are not conscientious tend to use words like fucking, pokemon, shit, gay, youtube.	Note that individuals who are conscientious tend to complete tasks successfully while individuals that are not conscientious tend to need a push to get started.
EXT	Note that individuals who are extraverted tend to be sociable, talkative, assertive and active while individuals that are not extraverted tend to be retiring, reserved and cautious.	Note that individuals who are extraverted tend to use words like party, girls, baby, gettin, chillin while individuals that are not extraverted tend to use words like anime, manga, internet, japanese, drawing.	Note that individuals who are extraverted tend to make friends easily while individuals that are not extraverted tend to avoid contact with others.
AGR	Note that individuals who are agreeable tend to be good-natured, compliant, modest, gentle, and cooperative while individuals that are not agreeable tend to be irritable, ruthless, suspicious and inflexible.	Note that individuals who are agreeable tend to use words like excited, blessed, great, wonderful, amazing while individuals that are not agreeable tend to use words like fuck, shit, bitch, damn, hell.	Note that individuals who are agreeable tend to believe that others have good intentions while individuals that are not agreeable tend to hold a grudge.
NEU	Note that individuals who are neurotic tend to be anxious, depressed, angry and insecure while individuals that are not neurotic tend to be calm, poised and emotionally stable.	Note that individuals who are neurotic tend to use words like fucking, depression, pissed, anymore, lonely while individuals that are not neurotic tend to use words like success, lakers, basketball, workout, beach.	Note that individuals who are neurotic tend to get stressed out easily while individuals that are not neurotic tend to feel comfortable with themselves.

Figure 2: Different types of knowledge used for each trait in the prompt.

Category	d	OR_{IDP}
NUMBER	0.699	0.140
SOCIAL	0.595	0.191
AFFILIATION	0.459	0.140
YOU	0.451	0.132
TIME	0.448	0.115
FAMILY	0.395	0.108
PPRON	0.359	0.104
POSEMO	0.341	0.102
AFFECT	0.242	0.061
FRIEND	0.217	0.057

Table 5: Lexical categories that are more prevalent when GPT-3 performs better than WT-LEX that explain their EXT predictions. d: Cohen's d – standardized difference in means (Diener, 2010); OR_{IDP}: log **o**dds **r**atio with informative dirichlet prior (Monroe et al., 2008).

Table 6 indicates that GPT-3 fails for OPE on language reflective of social processes (SOCIAL) and affect (AFFECT). Previous work on lexical correlates of personality showed that these categories are discussed more for users low in openness (Yarkoni, 2010), suggesting (together with our result) that GPT-3 misses the connection between these categories of language and personality. These are areas to improve the human-level capabilities of GPT-3.

Category	d	OR_{IDP}
THEY	0.701	0.126
FOCUSPAST	0.692	0.166
AFFECT	0.676	0.132
ACHIEVE	0.629	0.104
SOCIAL	0.608	0.168
SHEHE	0.588	0.172
PPRON	0.559	0.139
NEGATE	0.517	0.082
PRONOUN	0.510	0.105
POSEMO	0.482	0.118

Table 6: Lexical categories that are more prevalent when GPT-3 performs worse for the OPE task than WT-LEX. d: Cohen's d – standardized difference in means of errors (Diener, 2010); OR_{IDP}: log **o**dds **r**atio with informative dirichlet prior (Monroe et al., 2008) on errors.

Trait	Survey Item	Polarity	ItemDesc	AltPos	AltNeg	BothAltItems
OPE	Have a vivid imagination	+	✓		✓	
	Avoid philosophical discussions	-		✓	✓	
	Enjoy wild flights of fantasy	+		✓		✓
	Do not like poetry	-	✓			✓
CON	Complete tasks successfully	+	✓	✓		
	Need a push to get started	-	✓		✓	
	Am always prepared	+			✓	✓
	Shirk my duties	-		✓		✓
EXT	Do not mind being the centre of attention	+			✓	✓
	Make friends easily	+	✓	✓		
	Keep in the background	-		✓	✓	
	Avoid contact with others	-	✓			✓
AGR	Hold a grudge	-	✓	✓		
	Believe that others have good intentions	+	✓		✓	
	Cut others to pieces	-			✓	✓
	Am easy to satisfy	+		✓		✓
NEU	Feel comfortable with myself	-	✓	✓		
	Often feel blue	+		✓		✓
	Get stressed out easily	+	✓		✓	
	Am not easily bothered by things	-			✓	✓

Table 7: Survey items from the questionnaires answered by people for Big 5 personality assessment along with the combination labels these items were a part of (referenced in Table 3.

Trait	**Item Combination**	**Positive Item**	**Negative Item**	**Factor Loading**	**Macro F1**
OPE	ItemDesc	Have a vivid imagination	Do not like poetry	0.703	0.335
	AltNeg	Have a vivid imagination	Avoid philosophical discussions	0.714	0.342
	AltPos	Enjoy wild flights of fantasy	Avoid philosphical discussions	0.720	0.342
	BothAltItems	Enjoy wild flights of fantasy	Do not like poetry	0.787	0.374
CON	ItemDesc	Complete tasks successfully	Need a push to get started	0.781	0.521
	AltNeg	Am always prepared	Need a push to get started	0.800	0.457
	AltPos	Complete tasks successfully	Shirk my duties	0.821	0.476
	BothAltItems	Am always prepared	Shirk my duties	0.837	0.481
EXT	ItemDesc	Make friends easily	Avoid contact with others	0.766	0.569
	AltNeg	Do not mind being the centre of attention	Keep in the background	0.843	0.528
	AltPos	Make friends easily	Keep in the background	0.846	0.551
	BothAltItems	Do not mind being the centre of attention	Avoid contact with others	0.860	0.523
AGR	AltPos	Am easy to satisfy	Hold a grudge	0.725	0.501
	ItemDesc	Believe that others have good intentions	Hold a grudge	0.741	0.488
	AltNeg	Believe that others have good intentions	Cut others to pieces	0.809	0.509
	BothAltItems	Am easy to satisfy	Cut others to pieces	0.813	0.523
NEU	AltPos	Often feel blue	Feel comfortable with myself	0.697	0.333
	AltNeg	Get stressed out easily	Am not easily bothered by things	0.804	0.364
	ItemDesc	Get stressed out easily	Feel comfortable with myself	0.829	0.349
	BothAltItems	Often feel blue	Am not easily bothered by things	0.835	0.333

Table 8: Comparison of factor loading values of the aggregation of a positive item and a negative item from the Big 5 personality questionnaire and the performance of GPT-3 (ItemDesc) for the corresponding Itemdesc pairs. The factor loadings were calculated on an exeternal dataset (Kosinski et al., 2013) with larger number of samples (N=741). There's very little difference in the factor loading values (distinguisginh power) over the four combinations for almost all traits, which is in line with the minor performance differences observed in the consistency experiments explained in §section 6

Utterance Emotion Dynamics in Children's Poems: Emotional Changes Across Age

Daniela Teodorescu, Alona Fyshe
University of Alberta
`{dteodore,alona}@ualberta.ca`

Saif M. Mohammad
National Research Council Canada
`saif.mohammad@nrc-cnrc.gc.ca`

Abstract

Emerging psychopathology studies are showing that patterns of changes in emotional state — *emotion dynamics* — are associated with overall well-being and mental health. More recently, there has been some work in tracking emotion dynamics through one's utterances, allowing for data to be collected on a larger scale across time and people. However, several questions about how emotion dynamics change with age, especially in children, and when determined through children's writing, remain unanswered. In this work, we use both a lexicon and a machine learning based approach to quantify characteristics of emotion dynamics determined from poems written by children of various ages. We show that both approaches point to similar trends: consistent increasing intensities for some emotions (e.g., anger, fear, joy, sadness, arousal, and dominance) with age and a consistent decreasing valence with age. We also find increasing emotional variability, rise rates (i.e., emotional reactivity), and recovery rates (i.e., emotional regulation) with age. These results act as a useful baselines for further research in how patterns of emotions expressed by children change with age, and their association with mental health.

1 Introduction

Emotions play a key role in overall well-being (Kuppens and Verduyn, 2017; Houben et al., 2015; Silk et al., 2011; Sperry et al., 2020). People's emotional states are constantly changing in response to internal and external events, and the way in which we regulate emotions (Zimmermann and Iwanski, 2014; McRae et al., 2012). Patterns of emotion change over time have been shown to be related to general well-being and psychopathology (the scientific study of mental illness and disorders) (Houben et al., 2015; Sperry et al., 2020; Scott et al., 2020; Sheppes et al., 2015), academic success (Graziano et al., 2007), and social interactions in children (Sosa-Hernandez et al., 2022).

Several psychopathology studies have introduced metrics to quantify and understand the trajectories and patterns in emotions across time (Kuppens and Verduyn, 2017). These metrics are referred to as *Emotion Dynamics* and include features of the emotional episode (e.g., duration) and of the emotional trajectory (e.g., emotional variability, covariation, inertia) (Kuppens and Verduyn, 2017). In psychology, emotion dynamics have usually been captured through self-report surveys over periods of time (e.g., five times a day for ten days). However, obtaining such self-reports is arduous work; limiting the amount of data collected. Further, self-reports are prone to a number of biases (e.g., social pressures to be perceived as being happy).

Inspired by the emotion dynamics work in psychology, Hipson and Mohammad (2021) recently introduced the idea that patterns of emotion change can also be explored in the utterances of an individual, which can reflect their inner emotion dynamics. They refer to this as *utterance emotion dynamics (UED).* They generate emotion arcs from streams of text (e.g., sentences in a story, tweets over time, etc.), which are in turn used to determine various UED metrics.[1] Different UED metrics capture different aspects of emotion change (e.g., variability, rate of change, etc.).

Teodorescu and Mohammad (2022) performed experiments on 36 diverse datasets to show that the quality of emotion arcs generated using emotion lexicons is comparable to those generated from machine learning (ML) methods. The lexicon approach is able to perform well through the power of aggregating information (e.g., 50–100 instances per bin). Moreover, the lexicon approach obtains high performance even when using translations of

---

[1] An emotion arc is a series of time step–emotion value pairs that acts as a digital representation of how one's emotions change over time. There are several works in NLP that capture emotion arcs from streams of text (e.g., sentences in a story, tweets over time, etc.) (Mohammad, 2011, 2012; Reagan et al., 2016; Teodorescu and Mohammad, 2022, 2023).

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 401–415
July 14, 2023 ©2023 Association for Computational Linguistics

an English lexicon into low-resource languages, such as indigenous African languages (Teodorescu and Mohammad, 2023). Emotion lexicons have the benefit of interpretability, accessibility, and efficiency compared to ML models. Thus, we primarily used a lexicon-based approach in our experiments. However, we also show that the use of ML models points to the same trends as discovered by the lexicon approach.[2]

UED metrics, calculated from emotion arcs, can be computed for a *speaker* over time (e.g., main character in a narrative, tweets of a user over time), for multiple speakers at a time (e.g., treating all users in a geographic region as a *speaker* for whom we can compute UED), or at an *instance* level (e.g., independent posts where we compute UED metrics per post). While emotion dynamics have been studied in psychology for the past decades, UED was proposed only recently and has been applied to only a small number of domains (literature and tweets). Important questions such as *how do UED metrics change over development from toddlers to young adults?* and *how do the metrics change across one's adult life?*, remain unanswered.

Generally, children's writing is a less studied domain in NLP, and there is limited data available. Also, research regarding children has guidelines and regulations in place to protect this vulnerable section of society (Hipson and Mohammad, 2020). Yet, careful and responsible work such as the work done on the Child Language Data Exchange System (CHILDES) (MacWhinney, 2014) for understanding child language acquisition can be tremendously influential. Similarly, applying UED metrics to children's writing will allow us to infer the emotional states of children across age. Such work provides important information for psychologists and child development specialists, as emotion dynamics have been shown to underlie well-being, psychopathology, and success.

Poetry is a domain of growing interest in NLP (e.g., poem generation (Van de Cruys, 2020; Gonçalo Oliveira, 2017)). Poems are written to evoke emotions (Wassiliwizky et al., 2017; Johnson-Laird and Oatley, 2022) and a medium through which emotions are expressed (Whissell, 2004; Belfi et al., 2018). The intersection of poems and children's writing is an unexplored area which has the potential to unlock patterns in emotion word usage by children as they age.

In this paper we contribute to the knowledge of emotion change over time as children age by studying poems written by children. Our goal is to apply existing NLP techniques to study emotion change in childrens' writing rather than developing a novel algorithm for better detecting emotion. We investigate the following questions:

- How do the *average* emotions vary across grades? How does this compare for discrete emotions (e.g., anger, fear, joy, and sadness) and emotion dimensions (e.g., valence, arousal, and dominance)?
- How *variable* are emotion changes?

These first two questions help us set crucial metrics in UED, building on work by Hipson and Mohammad (2020). Next, to better understand patterns in emotion changes we look at:

- How does the rate at which children reach peak emotional states (*rise rate*) change with age? Rise rate is analogous to emotional reactivity, which is associated with well-being.
- How does the rate at which children recover from peak emotional states back to steady state (*recovery rate*) change with age? Recovery rate plays a role in emotion regulation, which is also associated with well-being.
- How do *utterance emotion dynamics* compare for adults vs. children?

Answers to these questions provide baseline metrics for emotion change in children's poems across age. In order to answer these questions, we use a dataset of $\sim 61K$ poems written by children (Hipson and Mohammad, 2020) to calculate various UED metrics and examine how they vary across age. The scores for the metrics and the analysis will act as useful baselines for further research on emotion dynamics in children's writing, and their implications on mental health and well-being.

2 Related Work

Below we review related work on emotion dynamics and its ties to well-being, the UED framework, and previous work on children's texts.

2.1 Emotion Dynamics

The *emotion dynamics* framework studies change in emotion over time as it is key to the study of emotions and overall well-being (Houben et al., 2015; Silk et al., 2011). Emotion dynamics metrics include *emotion intensity* and *emotion variability*.

---

[2] We did not find any poem datasets annotated for emotions that could be used to train an ML model; so we fine-tuned a pretrained ML model on emotion annotated tweets.

Dataset	# of Poems	#Words per Poem
PoKi	61,330	14.3
Grade 1	900	37.3
Grade 2	3,174	32.1
Grade 3	6,712	35.2
Grade 4	10,899	39.3
Grade 5	11,479	44.5
Grade 6	11,011	49.6
Grade 7	7,831	59.7
Grade 8	4,546	67.6
Grade 9	1,284	91.5
Grade 10	1,171	91.8
Grade 11	667	103.0
Grade 12	1,656	97.2
FPP	50	181.02

Table 1: Number of poems and the average lengths of poems in *PoKi* (by grade) and in *FPP*.

Emotion intensity is the average emotion over time. Whereas emotion variability is how much emotion changes from the average, often expressed as the standard deviation. These metrics have been used in various contexts in psychology to better understand well-being, often through self-reports or ecological momentary assessments.

The relationship between various metrics in emotion dynamics and well-being have been the topic of numerous psychology studies. Higher positive and negative affect variability have been shown to be associated with lower psychological well-being and more mental health symptoms in youth (Silk et al., 2003; van Roekel et al., 2016) and adults (Houben et al., 2015). Houben et al. (2015) showed that emotion variability has significant correlation with numerous psychological well-being categories: positive correlation with negative emotionality (e.g., negative affect and neuroticism), depression (e.g., depressive symptoms, depressive diagnosis), anxiety, borderline personality disorder, etc. On the other hand, emotional variability is negatively correlated with self-esteem, quality of life, and other signs of high psychological well-being (Houben et al., 2015).

Similarly, a vast number of studies explored the relationship between emotional regulation and reactivity with overall well-being. Hofmann et al. (2012) showed that mood and anxiety disorders are a result of emotion dysregulation of negative emotions, along with lacking positive emotions. Likewise, emotion dysregulation is thought of as the core of anxiety disorders (Mennin et al., 2007; Carthy et al., 2010). Children with anxiety disorders had higher negative emotion reactivity, and were less successful at implementing emotion regulation strategies (Carthy et al., 2010).

2.2 Utterance Emotion Dynamics

As work in psychology measures emotion dynamics through self-report measures, emotion dynamics can also be determined from text using NLP techniques such as sentiment analysis.

The UED framework (Hipson and Mohammad, 2021) tracks emotions dynamics in utterances, inspired by metrics in psychology. Such metrics include:

- *Home base*: The steady (most common) state where one is on average in emotional space.
- *Variability*: How much one's emotional state changes with time.
- *Rise Rate*: The rate at which one reaches peak emotional intensity, i.e., emotional reactivity.
- *Recovery Rate*: The rate at which one recovers from peak emotional intensity to home base, i.e., emotional regulation.

This framework was used to study emotion arcs of movie characters (Hipson and Mohammad, 2021), and to analyze emotional patterns across geographic regions through Twitter data (Vishnubhotla and Mohammad, 2022). Seabrook et al. (2018) studied the association between depression severity and emotion dynamics metrics such as variability on Facebook and Twitter. It was found that increased negative emotional variability was an indicator for lower depression severity on Twitter.

2.3 Children's Writing

Few work studies children's writing due to the limited data available. One of the most commonly known datasets is the Child Language Data Exchange System (CHILDES) (MacWhinney, 2014) and in French, E-CALM (Doquet, 2013). These datasets are limited in that they contain parent-child dialogue for children approximately age one to seven, and have limited quantities of text.

Very few works look at emotions in children's writing. Manabe et al. (2021) performed sentiment analysis on narratives written by youth for mental illness detection, as self-disclosure is not the norm in some cultures. Participants wrote an imaginative story and answered a questionnaire on their tendencies toward psychological distress. It was found that youth who had higher tendencies toward psychological distress used significantly more positive words, and therefore had higher valence. In this work, we study the patterns of emotion word changes in poems written by children.

3 Poem Datasets

For our experiments, we used a dataset of poems written by children as well as a dataset of poems written by adults (as control). Table 1 shows key statistics of each dataset.

Poems Written by Children (PoKi): Hipson and Mohammad (2020) compiled and curated a dataset of close to 61 thousand poems written by children in grades one to twelve. The poems were published and publicly available on the Scholastic Corporation website.[3] In the PoKi dataset each poem is released with the child's school grade (which can be used as a proxy for age) and first name.

The average emotional patterns for emotion dimensions (valence, arousal, and dominance), along with discrete emotions (anger, fear, joy and sadness) were analysed across grades. Additionally, these patterns were contrasted to those found in poems written by adults (data described below). It was found that as children grow from early childhood into adolescence, valence decreases reaching a minimum at grade 11. Whereas arousal increases with age, aligning with how adults display emotions more visibly (Dreyfuss et al., 2014; Somerville et al., 2011). Likewise, dominance increases with age. Consistently there was higher arousal in poems written by children with names commonly among males compared to those with names common among females. All intensities for anger, fear, joy and sadness increased across grades, with a particularly strong increase in sadness.

Poems Written by Adults (FPP): Hipson and Mohammad (2020) also compiled and used poems written by adults which were published on the *Famous Poets and Poems* website.[4] We will refer to this dataset as *FPP*. The poems are publicly available online and contain works by famous writers such as Edgar Allan Poe, and E.E. Cummings.

Preprocessing: We preprocessed both poem datasets by removing extra whitespace, punctuation, unescaping HTML (if any), tokenizing and lowercasing the text using the Twokenize[5] library. Additionally, stop words were removed.

4 Types of UED Metrics

In the past, UED metrics have been calculated for the speaker or jointly for text from a set of speakers (meta-speaker). We propose a third form of UED metrics not explored before — *instance* level UED metrics. All three of these types of UED metrics are described below:

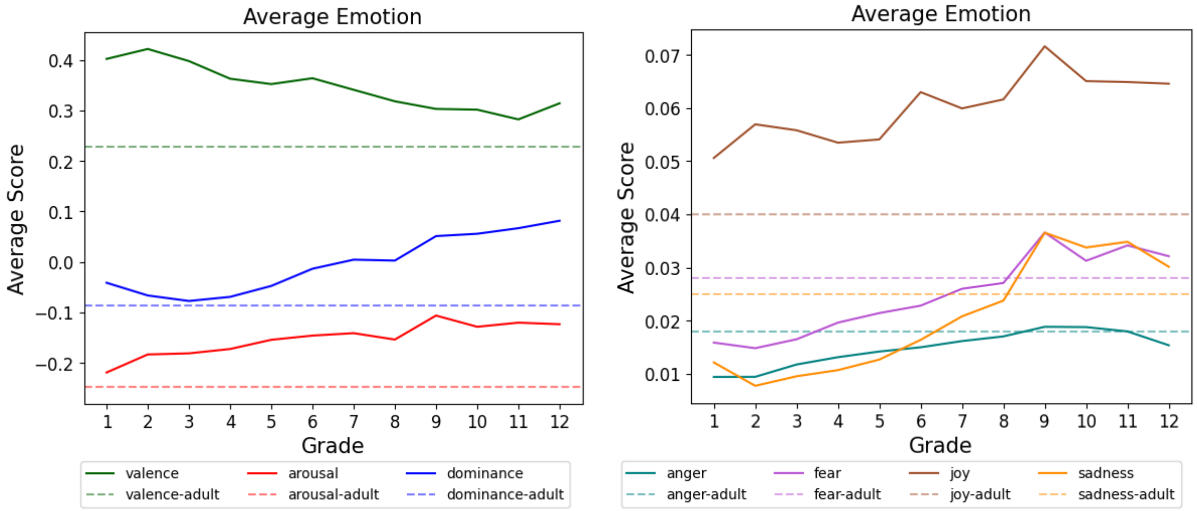
- **Speaker UED Metrics:** Here all available utterances by a speaker are placed in temporal order to form the text from which the UED metrics for the speaker is determined. For each metric, UED scores from multiple speakers can be averaged to determined the average UED score for that metric for the population. In the past, speaker UED metrics have been determined for characters in movie dialogues (Hipson and Mohammad, 2021), and for users on Twitter during the pandemic (Vishnubhotla and Mohammad, 2022).
- **Meta-Speaker UED Metrics:** If one is interested in analyzing change of emotions in a discourse by multiple speakers, for example, analyzing changes in emotion patterns in a Reddit thread, then we can treat each discourse (e.g., each Reddit thread) as text produced by a meta-speaker. Here we arrange each of the utterances in each of the discourses (e.g., Reddit thread) in temporal order and determine the UED metrics for each discourse. UED metrics for all of the discourses can be averaged to determine the average UED metric scores for a set of discourses. In the past, discourse UED metrics have been determined for users from geographic regions, such as treating all users on Twitter in a country as a speaker (Vishnubhotla and Mohammad, 2022).
- **Instance UED Metrics:** If one is interested in the change of emotions in individual pieces of text (or instances) such as a novel, a poem, a tweet, or a blog post, then we can simply apply the UED metrics to each instance. Such a metric is useful at individual instance level if the instance is long enough (otherwise the score for the metric is not a reliable on its own). However, even for smaller pieces of text, the UED scores from a large number of instances can give a reliable estimate of the distribution of these instance-level UED metrics. Such metrics can even be used to compare patterns of emotion change across different sets, where each set is composed of (a) instances from many speakers and (b) instances that are temporally unordered (either because that information is not available or because we are

---

[3] Hipson and Mohammad (2020) obtained permission to use these poems for research.

[4] http://famouspoetsandpoems.com/top_poems.html

[5] https://github.com/myleott/ark-twokenize-py



(a) Valence, arousal and dominance

(b) Anger, fear, joy, and sadness

Figure 1: Average emotion across grades. The horizontal dashed lines represent values in poems written by adults.

not interested in temporal ordering of items within a set). Examples of instance-level UED use include: determining UEDs of presidential speeches, comparing average UEDs of stream of consciousness essays of different age groups, etc.

In this work we are interested in children's poems (instances of poetic text) across age and not how each individual child has a different writing style. Therefore, we calculate UED metrics for each poem and average the scores for each grade.

5 Experiments

Our goal is to analyze how patterns of emotion words change with age in children's poems. In order to do so, we generate an emotion arc per poem and compute instance-level UED metrics. Afterwards, we average the UED metrics per grade to compare results across age. We use the Emotion Dynamics toolkit[6] to calculate UED metrics and our code for the experiments is available online.[7]

We use text windows of size five (excluding words with a neutral emotion score) and a step size of one to create an emotion arc per poem. We only considered poems that included at least five emotion words[8]. For each research question we computed the corresponding metrics: average emotional state, emotional variability, rise rate and recovery rate. Analyzing average emotion and variability allows us to build foundational knowledge into changes in patterns of emotion words. We then look at rise rate and recovery rate to further our understanding of children's emotion dynamics.

While older children (e.g., grade 10–12) tend to write on average longer poems than younger children (e.g., grade 1–3), these UED metrics are not affected by the length of the poems.[9] Other metrics calculate the number of displacements from home base or the length of displacements to peaks which are affected by poem length. Additionally, because poems are shorter than text streams such as novels, the number of windows that can be created from a poem is limited, so metrics specific to emotional displacement are not computed since they are more suitable for longer texts.

Each metric is computed for both dimensional emotions (e.g., valence, arousal, dominance) and discrete emotions (e.g., anger, fear, joy and sadness). We used the NRC Valence, Arousal, and Dominance (VAD) Lexicon (Mohammad, 2018a) and the NRC Emotion Intensity Lexicon (Mohammad, 2018b) for word-emotion scores.

In Section 5.1 we use the lexicon-based approach to generate emotion arcs. We explain how the metrics are computed, contrast the trends across grades and compare the results to poems written by adults. We discuss the ties of these results with work in psychology and implications for emotional development. In Section 5.2 we explore the same questions using an ML model for generating emotion arcs. We find similar trends across grade with the ML approach as when using the lexicon approach.

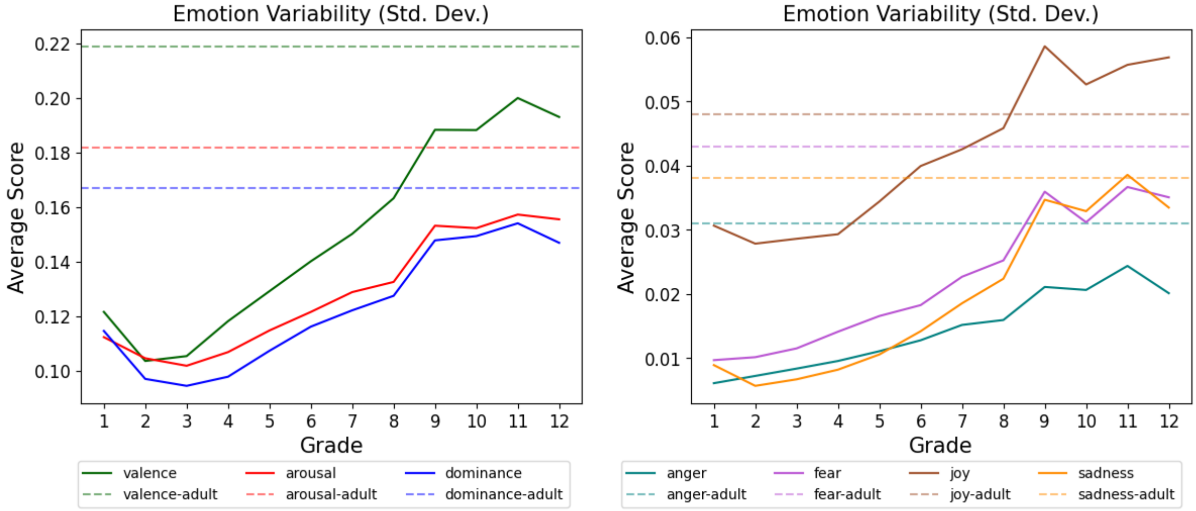
---

[6] https://github.com/Priya22/EmotionDynamics

[7] https://github.com/dteodore/EmotionArcs

[8] as per the NRC VAD lexicon

[9] We show in Appendix A that similar patterns in UED metrics hold when controlling for poem length across grades.



(a) Valence, arousal and dominance

(b) Anger, fear, joy, and sadness

Figure 2: Emotion variability across grades. The horizontal dashed lines represent values in poems written by adults.

UED Metric	Valence	Arousal	Dominance	Anger	Fear	Joy	Sadness	Psych. Construct
Average	0.228	-0.247	-0.087	0.018	0.028	0.040	0.025	Intensity
Variability	0.219	0.182	0.167	0.031	0.043	0.048	0.038	Emotional Variability
Rise Rate	0.134	0.114	0.084	0.115	0.109	0.066	0.113	Emotional Reactivity
Recovery Rate	0.127	0.105	0.086	0.024	0.028	0.023	0.020	Emotional Regulation

Table 2: The values for UED metrics in poems written by adults, and the corresponding construct in psychology.

5.1 Utterance Emotion Dynamics: PoKi

We begin with a question on how average emotion word score changes with grade–a question that Hipson and Mohammad (2020) already answered in their work. We replicated the experiment to make sure any differences in preprocessing the data or code development did not lead to different results. We then answer the other questions on how specifically do the trajectories of emotion change across grade differ, which have not been addressed yet. Likewise, we compute the UED metrics on the poems written by adults. We show the results in Table 2 and contrast them to PoKi below.

5.1.1 How does the average emotion expressed change across age?

Method: An average emotion score is calculated per window in the poem using word-emotion scores from the lexicon, and then the average is computed across windows in a poem.

Results: Below we present results on both the valence, arousal, and dominance (VAD) dimensions as well as for discrete emotion categories (Anger, Fear, Joy, Sadness).

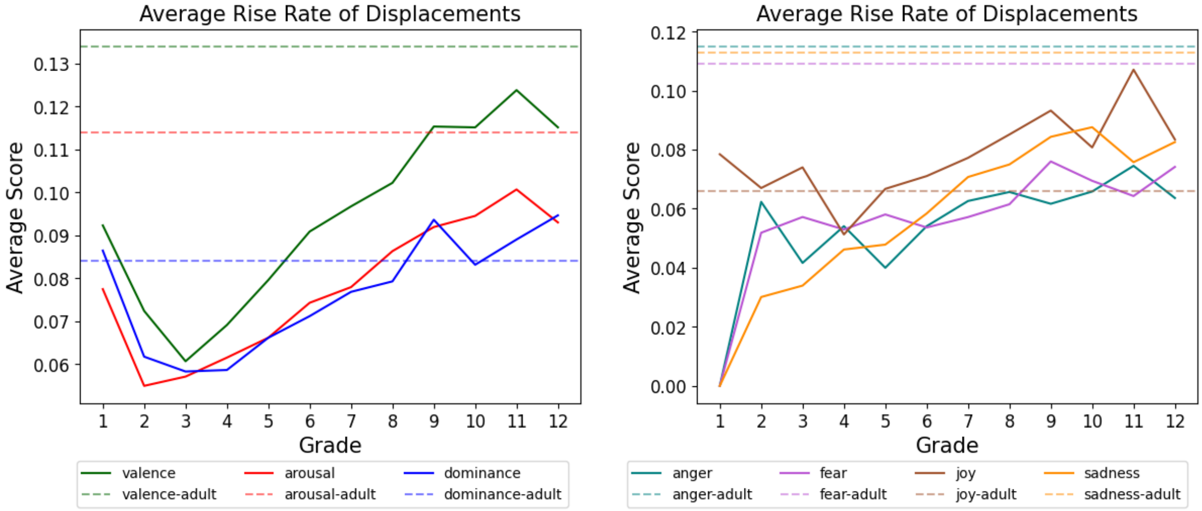
PoKi VAD: In Figure 1a, we show the average VAD emotions expressed across grade. Overall, we see a downward trend in valence from Grade 1 to Grade 12. This means that the poems written by younger children have, on average, more positive emotion words than those written by older children. There is a slight peak at grade 6, however a consistent downwards trend overall. Arousal and dominance similarly both trend upwards with age. There is a steeper increase for arousal and dominance at grade 9. This means that children are expressing more active and powerful emotions in poems as they age.

FPP VAD: The average valence of 0.228 is notably lower than the valence across grades, where the lowest is reached by grade 11s at 0.28. The average arousal at -0.247 and dominance at -0.087 are lower than those of children across all ages, and interestingly most similar to younger children.

PoKi Anger, Fear, Joy, Sadness: In Figure 1b, we see that the average discrete emotions all increase across grades. Anger, while increasing from grade 1 to 9, has a downward trend from grade 10 to 12. All emotions tend to have a peak around grade 9 and plateau afterwards.

FPP Anger, Fear, Joy, Sadness: Anger, fear and sadness tends to match to those of older children around grade 8 to 9. Children from grade 9 to 12 reach even higher values than adults for fear and sadness. On the other hand, joy tends to remain below those of children across all age, and has the most similar values to younger children at 0.04.

Discussion: These findings align with those by



(a) Valence, arousal and dominance

(b) Anger, fear, joy and sadness

Figure 3: Rise rate in poems across grades. The horizontal dashed lines represent values in poems written by adults.

Hipson and Mohammad (2020) which similarly computed the mean emotion in poems across grade. Numerous works in psychology have found similar trends through self-report studies for valence (Frost et al., 2015; Larson et al., 2002; Simmons et al., 1987; Weinstein et al., 2007), and arousal (Carstensen et al., 2000; Gunnar et al., 2009; Somerville, 2013). Likewise, as sadness increased with age, Holsen et al. (2000) have shown that teenagers are more likely to experience a negative and depressed mood. This trend matters because we are seeing similar trends in the emotion words used by children when writing poems as those in psychology self-reports, although they were not told to explicitly talk about how they are feeling. This work further contributes to the current findings on emotional development in children.

5.1.2 How variable are emotions across age?

Method: Variability is computed as the standard deviation of emotion values for windows in a poem.

Results:

PoKi VAD: In Figure 2a, variability for valence, arousal, and dominance all trend upward with age; stabilizing in grades 11 and 12.

FPP VAD: For all three emotions variability was most similar to those of older children, reaching slightly above grades 10–12.

PoKi Anger, Fear, Joy, Sadness: In Figure 2b, we see that variability for all emotions trend upwards from grade 1 to 9, and start to level out around grade 10 to 12. Anger, fear, and sadness all have a peak at grade 9 and grade 11. Joy has an especially pronounced peak at grade 9.

FPP Anger, Fear, Joy, Sadness: Variability in anger, fear and sadness is higher for adults than those expressed by children across all grades, and is most similar to older children around grade 11. Likewise, variability for joy in adults is more similar to older children, however around grade 8.

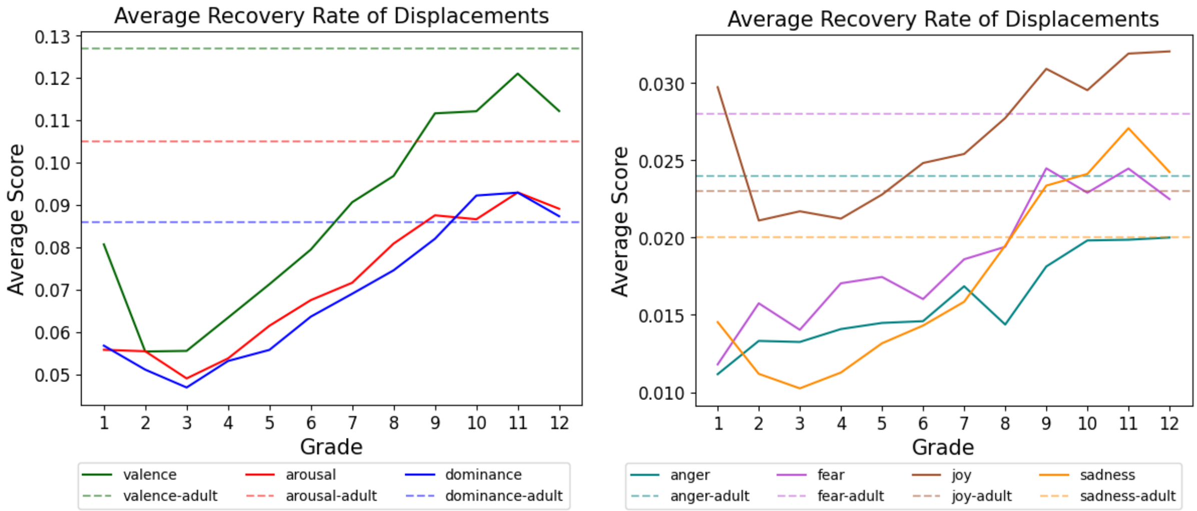
Discussion: The overall trend of increasing emotional variability with age, followed by stabilizing supports findings in psychology. Larson et al. (2002) found that emotional variability increased over early adolescence and stabilized around mid-adolescence. Further, during adolescence important cognitive, social and psychical changes occur which are thought to increase emotional variability (Buchanan et al., 1992; Arnett, 1999; Steinberg, 2005). Reitsema et al. (2022) found that sadness variability statistically increased with age. These trends are important as they support those found in psychology which are strongly associated with mental well-being (Reitsema et al., 2022).

5.1.3 At what rate do emotions change from home to peak state?

Method: The average rise rate is calculated as the average of the rise rate for windows in a poem. The rise rate is *peak distance* (how far away the peak is from the home base) divided by the number of words during the rise period. The rise rate disregards the direction of the peak.

Results:

PoKi VAD: In Figure 3a, we see that rise rate increases for all three emotions across grade, and plateaus around grade 10 to 12. The rise rate is comparably higher for valence, followed by arousal



(a) Valence, arousal and dominance

(b) Anger, fear, joy and sadness

Figure 4: Recovery rate in poems across grades. The horizontal dashed lines represent values for poems by adults.

and then dominance.

FPP VAD: The rise rate for valence and arousal in adults is higher than those across all grades, and is most similar to older students in grade 11. The rise rate for dominance in adults also matches those of older children, however starting at grade 8 (with grade 9, 11 and 12 having a higher rate than adults).

PoKi Anger, Fear, Joy, Sadness: In Figure 3b, the rise rate for the discrete emotions all increase with grade. Joy has a small dip around grade 4, and then increases matching the average rise rate of anger, fear, and sadness which all started at slightly lower values in grade 2. We note that at grade 1 we could not compute the average rise rate for anger, fear, and sadness as the poems had too few displacements (the number of poems which had displacements was less than our pre-chosen threshold of 5).

FPP Anger, Fear, Joy, Sadness: The rise rate for anger, fear, and sadness in adult poems is higher than those expressed in children across all ages, with most similar values to older children. However, the rise rate for joy is lower and corresponds with younger children around grade 2 to 5.

Discussion: Rise rate is seen as analogous to reactivity in psychology, which has been found to increase during adolescence (Somerville, 2016). Our findings support these trends. As mentioned in 2.1, emotional reactivity is at the core of anxiety and attention disorder, impacting overall well-being.

5.1.4 At what rate do emotions recover?

Method: Recovery rate is computed similarly to rise rate, however divides peak distance by the number of words during the recovery period. Recovery rate does not distinguish between peak direction.

Results:

PoKi VAD: Figure 4a we see the recovery rate increases for all three emotions with age and plateaus around grade 10 to 12. While the valence recovery rate has a larger magnitude than the other emotions, all rates trend upwards. Recovery rate can be thought of *emotion regulation*, indicating that older children are able to return to their home base emotional states after a peak more quickly than younger children.

FPP VAD: The recovery rate of adults for valence and arousal corresponds most closely with older children (e.g., grades 9–12), however is higher than across all grades. The recovery rate of dominance is similar to grade 9 students, and slightly below those of older children.

PoKi Anger, Fear, Joy, Sadness: In Figure 4b, we similarly see increasing average recovery rates for all 4 emotions across age. The magnitude of joy's recovery rate is considerably higher than for the other 3 emotions.

FPP Anger, Fear, Joy, Sadness: The recovery rate for fear and anger is above those across all ages, and is most similar to older children in grade 9–12. On the other hand, the recovery rate for joy and sadness matches those of younger children, around grade 5 for joy and 8 for sadness.

Discussion: Recovery rate, which is analogous to emotion regulation, has been studied extensively in psychology. Zeman et al. (2006) detail the progression of emotional regulation from infancy to adolescence, in which an increase in emotion regulation occurs alongside developments in strategies and motivations. Not only does emotion regulation have ties with well-being, it also plays a role in aca-

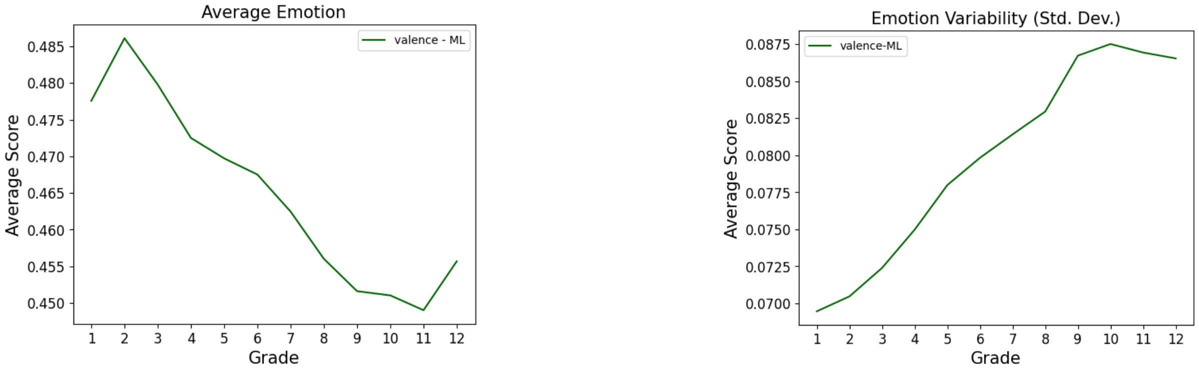


Figure 5: Average emotion and emotion variability for valence using the ML *n-gram* approach on the PoKi dataset.

demic success of children (Graziano et al., 2007) and adults (Phillips et al., 2002).

5.2 Utterance Emotion Dynamics - ML Approach: PoKi

To perform a comprehensive analysis and ensure the trends in emotion change are consistent regardless of the emotion labelling method used, we also performed experiments using a ML model. Previously, individual words were emotion labelled using a lexicon. Now, we use a *n-gram* approach where a ML model assigns emotion scores to windows of text in the poem of length *n*. We are not trying to determine which of these two approaches is *better* at computing UED metrics as this would be challenging - there are no existing annotated datasets for emotion arcs, or UED metrics. Rather, we are supporting the trends found by the word-level lexicon approach, with those found by ML models as they are commonly used on downstream tasks (e.g., sentiment analysis) and are known for their strong performance. If the ML approach did not perform well, we would not expect any trends in UED metrics to appear.

Datasets: We use the same poem datasets as in Section 3, creating n-gram windows of length 5. The only difference is that words not found in the emotion lexicon or neutral words can be included. We choose this approach as ML models are trained on sequential text.

Experiments: We fine-tuned a RoBERTa (Liu et al., 2019) base model for fine-grain sentiment analysis using the SemEval 2018 Task 1 dataset (Mohammad et al., 2018). This means that we were able to predict an emotion score between 0 and 1. Details on the model training are in Appendix B.1. After emotion labelling text windows, we performed similar experiments as in Section 5: compute the UED metrics per poem and take the average per grade for each metric.

Results: Overall we found similar trends as with the *word-level* lexicon approach. We note that a direct comparison between the lexicon and ML approach can not be made as they are using different units of measurement (e.g., windows contain either sequential words found in the lexicon or natural sequences of words). We can simply compare the trends in emotion change rather than the magnitude of change or the values themselves. We discuss the results for valence below (Figure 5 and Appendix B.2). We also show the results for the discrete emotions in Appendix B.2, as the trends were similar to the lexicon approach.

Average Emotion: As grade increases, we see a similar downwards trend and a stabilization at grades 10–12.

Emotional Variability: Older children tend to show increased variability.

Rise Rate and Recovery Rate: With age, children are writing with increased rates of emotional reactivity and also emotion regulation.

Overall, these results show that there are patterns of emotion change in childrens' poems with age, and trends found using the lexicon approach are also replicated using ML models.

6 Conclusion

We explored four utterance emotion dynamics metrics (average, variability, rise rate, and recovery rate), and seven emotions (three dimensional and four discrete) on poems written by children and adults. We found that the patterns of emotion changes in poetry by children supported previous results and findings in the psychology literature (e.g., increased variability, rise rate, and recovery rates with age).

As future work, we would like to examine poetry by adults more in-depth, such as how do patterns of emotion change look for experts vs. novices? And how do UED compare across geographic regions, and time periods.

Limitations

A limitation of this work is that the poems written by adults are by experienced writers who are often known for their poetry. These poems may therefore not be representative of poems written by adults in general, and could affect the patterns and trends in emotion words we see. As future work we would like to expand the collection of poems written by adults to include those written by novices as well.

Ethics Statement

Our research interest is to study emotions at an aggregate/group level. This has applications in emotional development psychology and in public health (e.g., overall well-being and mental health). However, emotions are complex, private, and central to an individual's experience. Additionally, each individual expresses emotion differently through language, which results in large amounts of variation. Therefore, several ethical considerations should be accounted for when performing any textual analysis of emotions (Mohammad, 2022, 2023). The ones we would particularly like to highlight are listed below:

- Our work on studying emotion word usage should not be construed as detecting how people feel; rather, we draw inferences on the emotions that are conveyed by users via the language that they use.
- The language used in an utterance may convey information about the emotional state (or perceived emotional state) of the speaker, listener, or someone mentioned in the utterance. However, it is not sufficient for accurately determining any of their momentary emotional states. Deciphering true momentary emotional state of an individual requires extra-linguistic context and world knowledge. Even then, one can be easily mistaken.
- The inferences we draw in this paper are based on aggregate trends across large populations. We do not draw conclusions about specific individuals or momentary emotional states.

Acknowledgements

We thank Will Hipson for collecting the PoKi poems dataset and for setting the groundwork in this direction of research. We also thank Krishnapriya Vishnubhotla for the toolkit she created to help compute UED metrics.

References

JJ Arnett. 1999. Adolescent storm and stress, reconsidered. *The American psychologist*, 54(5):317—326.

Amy M Belfi, Edward A Vessel, and G Gabrielle Starr. 2018. Individual ratings of vividness predict aesthetic appeal in poetry. *Psychology of Aesthetics, Creativity, and the Arts*, 12(3):341.

CM Buchanan, JS Eccles, and JB Becker. 1992. Are adolescents the victims of raging hormones: evidence for activational effects of hormones on moods and behavior at adolescence. *Psychological bulletin*, 111(1):62—107.

LL Carstensen, M Pasupathi, U Mayr, and JR Nesselroade. 2000. Emotional experience in everyday life across the adult life span. *Journal of personality and social psychology*, 79(4):644—655.

Tal Carthy, Netta Horesh, Alan Apter, and James J. Gross. 2010. Patterns of emotional reactivity and regulation in children with anxiety disorders. *Journal of Psychopathology and Behavioral Assessment*, 32(1):23–36. This study was supported by a research fund of the Adler Research Center in Tel-Aviv University. The authors would like to thank the Anxiety Disorders Clinic in 'Schneider's Children Medical Center of Israel' for support and collaboration. Special thanks to Ronit Jossifoff, Maya Ferber, Yael Tadmor and Hilit Pritsch for their important contribution to the recruitment and examination of the participants.

Claire Doquet. 2013. Ancrages théoriques de l'analyse génétique des textes d'élèves. In Catherine Bore et Eduardo Calil, editor, *L'Ecole, l'écriture et la création. Etudes françaises et brésiliennes.*, Sciences du langage - Carrefour et points de vue, pages 33–53. Academia Bruylant.

Michael Dreyfuss, Kristina Caudle, Andrew T Drysdale, Natalie E Johnston, Alexandra O Cohen, Leah H Somerville, Adriana Galván, Nim Tottenham, Todd A Hare, and BJ Casey. 2014. Teens impulsively react rather than retreat from threat. *Developmental neuroscience*, 36(3-4):220—227.

Allison Frost, Lindsay T. Hoyt, Alissa Levy Chung, and Emma K. Adam. 2015. Daily life with depressive symptoms: Gender differences in adolescents' everyday emotional experiences. *Journal of Adolescence*, 43:132–141.

Hugo Gonçalo Oliveira. 2017. A survey on intelligent poetry generation: Languages, features, techniques, reutilisation and evaluation. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 11–20, Santiago de Compostela, Spain. Association for Computational Linguistics.

Paulo A. Graziano, Rachael D. Reavis, Susan P. Keane, and Susan D. Calkins. 2007. The role of emotion regulation in children's early academic success. *Journal of School Psychology*, 45(1):3–19.

Megan R Gunnar, Sandi Wewerka, Kristin Frenn, Jeffrey D Long, and Christopher Griggs. 2009. Developmental changes in hypothalamus-pituitary-adrenal activity over the transition to adolescence: normative changes and associations with puberty. *Development and psychopathology*, 21(1):69—85.

Will Hipson and Saif M. Mohammad. 2020. PoKi: A large dataset of poems by children. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1578–1589, Marseille, France. European Language Resources Association.

Will E. Hipson and Saif M. Mohammad. 2021. Emotion dynamics in movie dialogues. *PLOS ONE*, 16(9):1–19.

Stefan G. Hofmann, Alice T. Sawyer, Angela Fang, and Anu Asnaani. 2012. Emotion dysregulation model of mood and anxiety disorders. *Depression and Anxiety*, 29(5):409–416.

Ingrid Holsen, Pal Kraft, and Joar Vitterso. 2000. Stability in depressed mood in adolescence: Results from a 6-year longitudinal panel study. *Journal of Youth and Adolescence*, 29(1):61–78. Copyright - Copyright Plenum Publishing Corporation Feb 2000; Last updated - 2023-02-07; CODEN - JYADA6.

Marlies Houben, Wim Van Den Noortgate, and Peter Kuppens. 2015. The relation between short-term emotion dynamics and psychological well-being: A meta-analysis. *Psychological Bulletin*, 141(4):901–930.

Philip N. Johnson-Laird and Keith Oatley. 2022. How poetry evokes emotions. *Acta Psychologica*, 224:103506.

Peter Kuppens and Philippe Verduyn. 2017. Emotion dynamics. *Current Opinion in Psychology*, 17:22–26. Emotion.

Reed W. Larson, Giovanni Moneta, Maryse H. Richards, and Suzanne Wilson. 2002. Continuity, stability, and change in daily emotional experience across adolescence. *Child Development*, 73(4):1151–1165.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Brian MacWhinney. 2014. *The CHILDES project: Tools for analyzing talk, Volume II: The database*. Psychology Press.

Masae Manabe, Kongmeng Liew, Shuntaro Yada, Shoko Wakamiya, and Eiji Aramaki. 2021. Estimation of psychological distress in japanese youth through narrative writing: Text-based stylometric and sentiment analyses. *JMIR Form Res*, 5(8):e29500.

Kateri McRae, James J. Gross, Jochen Weber, Elaine R. Robertson, Peter Sokol-Hessner, Rebecca D. Ray, John D.E. Gabrieli, and Kevin N. Ochsner. 2012. The development of emotion regulation: an fMRI study of cognitive reappraisal in children, adolescents and young adults. *Social Cognitive and Affective Neuroscience*, 7(1):11–22.

Douglas S. Mennin, Robert M. Holaway, David M. Fresco, Michael T. Moore, and Richard G. Heimberg. 2007. Delineating components of emotion and its dysregulation in anxiety and mood psychopathology. *Behavior Therapy*, 38(3):284–302.

Saif Mohammad. 2011. From once upon a time to happily ever after: Tracking emotions in novels and fairy tales. In *Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 105–114, Portland, OR, USA. Association for Computational Linguistics.

Saif Mohammad. 2023. Best practices in the creation and use of emotion lexicons. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1825–1836, Dubrovnik, Croatia. Association for Computational Linguistics.

Saif M. Mohammad. 2012. From once upon a time to happily ever after: Tracking emotions in mail and books. *Decision Support Systems*, 53(4):730–741. 1) Computational Approaches to Subjectivity and Sentiment Analysis 2) Service Science in Information Systems Research : Special Issue on PACIS 2010.

Saif M. Mohammad. 2018a. Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 english words. In *Proceedings of The Annual Conference of the Association for Computational Linguistics (ACL)*, Melbourne, Australia.

Saif M. Mohammad. 2018b. Word affect intensities. In *Proceedings of the 11th Edition of the Language Resources and Evaluation Conference (LREC-2018)*, Miyazaki, Japan.

Saif M. Mohammad. 2022. Ethics sheet for automatic emotion recognition and sentiment analysis. *To Appear in Computational Linguistics*.

Saif M. Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. Semeval-2018 Task 1: Affect in tweets. In *Proceedings of International Workshop on Semantic Evaluation (SemEval-2018)*, New Orleans, LA, USA.

Louise H Phillips, Louise H Phillips, Rebecca Bull, Ewan Adams, and Lisa Fraser. 2002. Positive mood and executive function: evidence from stroop and fluency tasks. *Emotion (Washington, D.C.)*, 2(1):12—22.

Andrew J. Reagan, Lewis Mitchell, Dilan Kiley, Christopher M. Danforth, and Peter S. Dodds. 2016. The emotional arcs of stories are dominated by six basic shapes. *EPJ Data Science*, 5(1):1–12. Copyright -

EPJ Data Science is a copyright of Springer, 2016; Last updated - 2017-02-06.

Anne M Reitsema, Bertus F Jeronimus, Marijn van Dijk, and Peter de Jonge. 2022. Emotion dynamics in children and adolescents: A meta-analytic and descriptive review. *Emotion (Washington, D.C.)*, 22(2):374—396.

Lori N Scott, Sarah E Victor, Erin A Kaufman, Joseph E Beeney, Amy L Byrd, Vera Vine, Paul A Pilkonis, and Stephanie D Stepp. 2020. Affective dynamics across internalizing and externalizing dimensions of psychopathology. *Clinical psychological science : a journal of the Association for Psychological Science*, 8(3):412—427.

Elizabeth M Seabrook, Margaret L Kern, Ben D Fulcher, and Nikki S Rickard. 2018. Predicting depression from language-based emotion dynamics: Longitudinal analysis of facebook and twitter status updates. *J Med Internet Res*, 20(5):e168.

Gal Sheppes, Gaurav Suri, and James J. Gross. 2015. Emotion regulation and psychopathology. *Annual Review of Clinical Psychology*, 11(1):379–405. PMID: 25581242.

Jennifer S. Silk, Erika E. Forbes, Diana J. Whalen, Jennifer L. Jakubcak, Wesley K. Thompson, Neal D. Ryan, David A. Axelson, Boris Birmaher, and Ronald E. Dahl. 2011. Daily emotional dynamics in depressed youth: A cell phone ecological momentary assessment study. *Journal of Experimental Child Psychology*, 110(2):241–257. Special Issue: Assessment of Emotion in Children and Adolescents.

Jennifer S Silk, Laurence Steinberg, and Amanda Sheffield Morris. 2003. Adolescents' emotion regulation in daily life: links to depressive symptoms and problem behavior. *Child development*, 74(6):1869—1880.

RG Simmons, R Burgeson, S Carlton-Ford, and DA Blyth. 1987. The impact of cumulative change in early adolescence. *Child development*, 58(5):1220—1234.

Leah H Somerville. 2013. Special issue on the teenage brain: Sensitivity to social evaluation. *Current directions in psychological science*, 22(2):121—127.

Leah H Somerville, Todd Hare, and BJ Casey. 2011. Frontostriatal maturation predicts cognitive control failure to appetitive cues in adolescents. *Journal of cognitive neuroscience*, 23(9):2123—2134.

LH Somerville. 2016. Emotional development in adolescence. *Handbook of emotions*, pages 350–365.

Linda Sosa-Hernandez, McLennon Wilson, and Heather A Henderson. 2022. Emotion dynamics among preadolescents getting to know each other: Dyadic associations with shyness. *Emotion (Washington, D.C.)*.

Sarah H. Sperry, Molly A. Walsh, and Thomas R. Kwapil. 2020. Emotion dynamics concurrently and prospectively predict mood psychopathology. *Journal of Affective Disorders*, 261:67–75.

Laurence Steinberg. 2005. Cognitive and affective development in adolescence. *Trends in Cognitive Sciences*, 9(2):69–74.

Daniela Teodorescu and Saif M. Mohammad. 2022. Frustratingly easy sentiment analysis of text streams: Generating high-quality emotion arcs using emotion lexicons.

Daniela Teodorescu and Saif M. Mohammad. 2023. Generating high-quality emotion arcs for low-resource languages using emotion lexicons.

Tim Van de Cruys. 2020. Automatic poetry generation from prosaic text. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2471–2480, Online. Association for Computational Linguistics.

Eeske van Roekel, Elise C Bennik, Jojanneke A Bastiaansen, Maaike Verhagen, Johan Ormel, Rutger C M E Engels, and Albertine J Oldehinkel. 2016. Depressive symptoms and the experience of pleasure in daily life: An exploration of associations in early and late adolescence. *Journal of abnormal child psychology*, 44(5):999—1009.

Krishnapriya Vishnubhotla and Saif M. Mohammad. 2022. Tweet Emotion Dynamics: Emotion word usage in tweets from US and Canada. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4162–4176, Marseille, France. European Language Resources Association.

Eugen Wassiliwizky, Stefan Koelsch, Valentin Wagner, Thomas Jacobsen, and Winfried Menninghaus. 2017. The emotional power of poetry: neural circuitry, psychophysiology and compositional principles. *Social cognitive and affective neuroscience*, 12(8):1229—1240.

Sally M. Weinstein, Robin J. Mermelstein, Benjamin L. Hankin, Donald Hedeker, and Brian R. Flay. 2007. Longitudinal patterns of daily affect and global mood during adolescence. *Journal of Research on Adolescence*, 17(3):587–600.

Cynthia Whissell. 2004. Poetic emotion and poetic style: The 100 poems most frequently included in anthologies and the work of emily dickinson. *Empirical Studies of the Arts*, 22(1):55–75.

Janice Zeman, Michael Cassano, Carisa Perry-Parrish, and Sheri Stegall. 2006. Emotion regulation in children and adolescents. *Journal of developmental and behavioral pediatrics : JDBP*, 27(2):155—168.

Peter Zimmermann and Alexandra Iwanski. 2014. Emotion regulation from early adolescence to emerging adulthood and middle adulthood: Age differences,

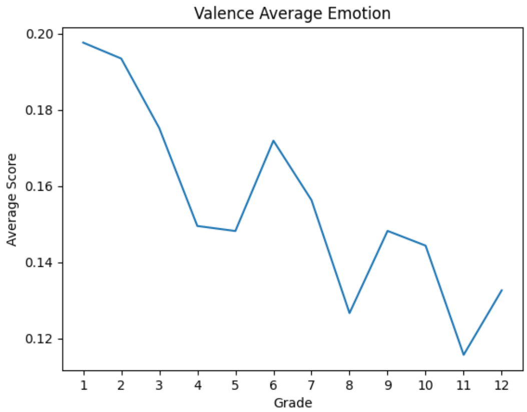


Figure 6: Average valence across grades for poems of length 10 to 20 words.

gender differences, and emotion-specific developmental variations. *International Journal of Behavioral Development*, 38(2):182–194.

A Poem Length on UED Metrics

As mentioned in Section 5, certain UED metrics which rely on distances (e.g., length of displacements to peaks) could be influenced by poem length. Therefore, we selected metrics which are based on rates or averages. To verify these metrics are not impacted by the increasing poem lengths with age, we investigated if the same trends hold when controlling for the length of poems across grade. In Figure 6 we show the results for the average valence across grades for poems of length 10 to 20 words (not including stop words). As grade increases, we similarly see a decrease in valence. Similar trends occur with other metrics.

B Machine Learning Approach

In Section B.1 we detail the model training process and in Section B.2 we show the results on the PoKi dataset using a ML approach.

B.1 Model Training

We fine-tuned the pretrained RoBERTa (Liu et al., 2019) base model available on HuggingFace[10]. For training we used the SemEval 2018 Task 1 dataset (Mohammad et al., 2018) which contains tweets annotated with emotion scores for valence, anger, fear, joy and sadness.[11] The dataset contains both fine-grain emotion scores (real-valued numbers between 0 and 1) and categorical labels (e.g., -1, 0, 1). We use the real-valued emotion scores to compute more fine-grained emotion arcs. More details on this dataset are available in Table 3 and Table 4.

We used the Trainer pipeline from HuggingFace[12] to fine-tune the pretrained model. For the loss function we used mean-square loss.

We tuned the following hyperparameters on the development set and selected the best model using mean-square error: learning rate (2e-5, 3e-5), number of epochs (5, 10, 20) and batch size (16, 32). Note that our aim here is not to overly fine-tune the model as we are applying it to a different domain (i.e., poems). The best parameters for each emotion model are shown in Table 5. After determining the *best* model on the development set we apply it to windows of text in the PoKi poem dataset.

B.2 UED Metric Results

In Figure 7 we show the results for rise rate and recovery rate for valence using the fine-tuned ML model on the PoKi dataset. Both rise rate and recovery rate increase with age. These trends support those seen when using the lexicon approach.

In Figure 8 we show the UED metrics across grade for the discrete emotions (e.g., anger, fear, joy, and sadness). The trends for fear and sadness are similar to trends found when using the lexicon approach: emotion intensity, variability, rise rate and recovery rate increase across grades. Overall, the patterns of emotion change for anger are flatter across metrics. Perhaps anger is a more challenging emotion for automatic systems to detect (Mohammad et al., 2018). The average intensity for joy has a similar pattern to that of valence. These two emotions could appear similar to the ML model resulting in similar trajectories.

---

[10] `https://huggingface.co/roberta-base`

[11] We could not train models for arousal and dominance as there are no corresponding annotated datasets.

[12] `https://huggingface.co/docs/evaluate/main/en/transformers_integrations#trainer`

Dataset	**Source**	**Domain**	**Dimension**	**Label Type**	**# Instances**
SemEval 2018 (EI-Reg)	Mohammad et al. (2018)	tweets	anger, fear joy, sadness	continuous (0 to 1)	3092, 3627, 3011, 2095
SemEval 2018 (V-Reg)	Mohammad et al. (2018)	tweets	valence	continuous (0 to 1)	2567

Table 3: Dataset descriptive statistics. The No. of instances includes the train, dev, and test sets for the Sem-Eval 2018 Task 1 (EI-Reg and V-Reg).

Emotion	**Train**	**Dev.**	**Test**
Valence	1181	449	937
Anger	1701	388	1002
Fear	2252	389	986
Joy	1616	290	1105
Sadness	1533	397	975

Table 4: The number of tweets in each of the dataset splits for the SemEval 2018 Task 1.

Emotion	**Learning Rate**	**No. Epochs**	**Batch Size**
Valence	3e-05	32	5
Anger	2e-05	32	10
Fear	3e-05	32	10
Joy	2e-05	16	5
Sadness	2e-05	32	10

Table 5: The optimal hyperparameter settings when fine-tuning the RoBERTa base model on the SemEval 2018 Task 1 dataset for each emotion.

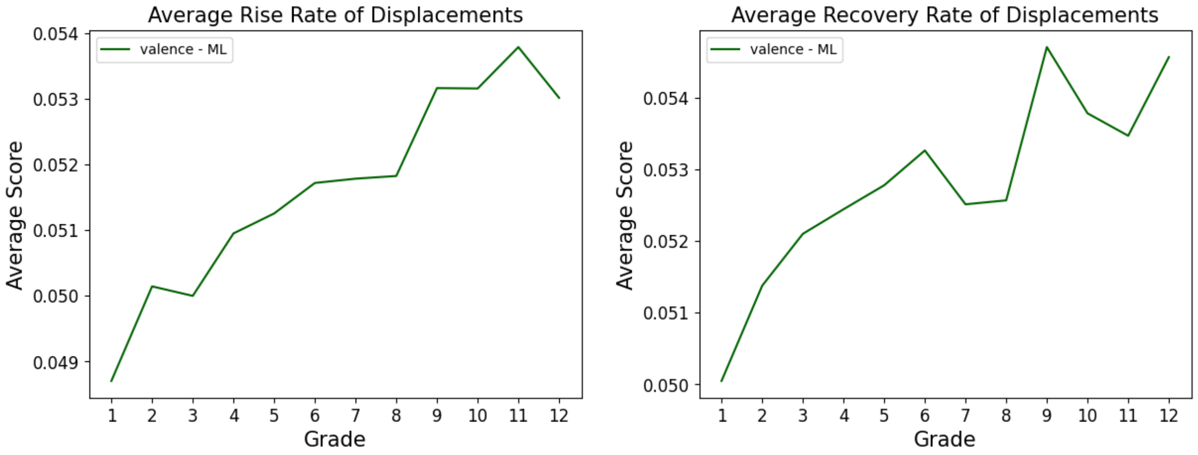


Figure 7: Rise rate and recovery rate for valence using the ML *n-gram* approach on the PoKi dataset.

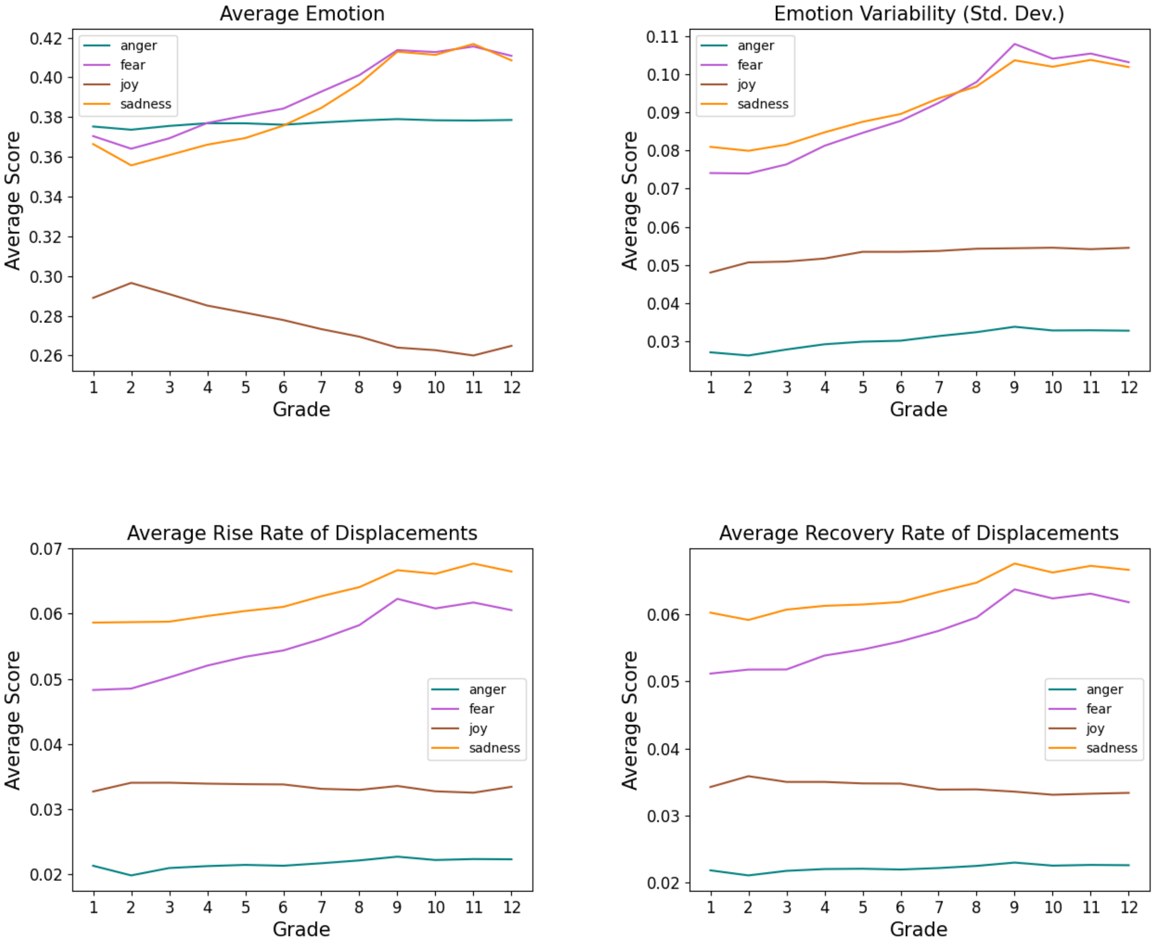


Figure 8: UED metrics for anger, fear, joy and sadness using the ML *n-gram* approach on the PoKi dataset.

Annotating and Training for Population Subjective Views

Maria Alexeeva† **Caroline Campbell Hyland**† **Keith Alcock**† **Allegra A. Beal Cohen**‡

Hubert Kanyamahanga◁ **Isaac Kobby Anni**◁ **Mihai Surdeanu**†

† University of Arizona, Tucson, AZ, USA
‡ University of Florida, Gainesville, FL, USA
◁ International Crops Research Institute for the Semi-Arid Tropics, Dakar, Senegal
`{alexeeva, msurdeanu}@arizona.edu`

Abstract

In this paper, we present a dataset of subjective views (beliefs and attitudes) held by individuals or groups.[1] We analyze the usefulness of the dataset by training a neural classifier that identifies belief-containing sentences that are relevant for our broader project of interest—scientific modeling of complex systems. We also explore and discuss difficulties related to annotation of subjective views and propose ways of addressing them.

1 Introduction

Collecting annotated data for training natural language processing (NLP) models is a difficult and expensive task, involving selection of data to annotate, preparing guidelines, training annotators, and more. With best prepared annotation efforts, one has to deal with disagreement among annotators, also known as Human Label Variation (Plank, 2022), and find ways to mitigate or embrace it.

The issue is even more prominent when it comes to annotation of tasks that deal with subjectivity—when an annotation assignment is not guaranteed to have one correct answer, but is open to interpretation. An objective task, e.g., determining whether a word is a noun under an annotation schema informed by a certain linguistic framework, would be less complicated than a subjective task of determining whether or not a tweet is sarcastic. Other examples of highly subjective tasks are emotion (e.g., Davani et al., 2022), humour (e.g., Meaney et al., 2021), and, to an extent, fake news detection (e.g., Pomerleau and Rao, 2017, Thorne et al., 2018).

In this paper, we present work on another task that has a high level of subjectivity: identifying subjective views of populations. We describe the task, the associated annotation effort, model-training experiments with the resulting dataset, and initial work on using the outputs of the trained model.

This work and its goals, as well as the definitions of subjective views and related terms, stem from work on using computational models to understand complex systems, e.g., agricultural value chains (AVC), food supply chains, or pandemics. Philosophers such as Heidegger argue that our "being in the world" means that all our decisions are subjective and depend on our current operating context (Dreyfus, 1990). With people being active participants and decision makers in the systems that modeling experts are trying to understand, these systems have to be, to an extent, driven by subjective beliefs of the human participants. Thus, in order to have a complete mechanistic understanding of a complex process, it is crucial for modelers to access subjective views of the populations involved. With the abundance of information available online making it difficult for modelers to identify relevant subjective views, our goal is to identify them automatically.

With this paper, we make the following contributions:

- We release a dataset for identifying subjective views of individuals or groups.
- We train a model for identifying such subjective views in text using the created dataset.
- We discuss ways in which we mitigate issues related to human label variation and provide support for embracing it through error analysis of the model predictions and application of the models trained on the data by intended users.

2 Dataset

2.1 Task Definition

With this project, we aim to help scientific modelers improve their models of complex systems through

---

[1] The dataset and the code are available at `https://github.com/clulab/releases/tree/master/wassa2023-beliefs`.

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 416–430
July 14, 2023 ©2023 Association for Computational Linguistics

incorporating views of populations, which could potentially impact those systems. With this in mind, we annotate two types of subjective views: *beliefs* and *attitudes*.

We define beliefs as people's views on how the world works, or in other words, their mental models or parts of it. For instance, the following example shows people's understanding of the relation between price and quality, which can impact their purchasing behaviors, and, in turn, impact the food supply chain:

> *Consumers generally recognize that cheaper prices correspond with lower quality and tend to remain loyal to their preferences when prices increase.*

We define attitudes as subjective views that indicate people's feelings towards objects and events. The example below shows how people's attitude (wanting to secure more food) led to their behavior (cultivating crops twice within a season):

> *However, the members cultivated rice twice in 2009/10 [...] because they did not plan to cultivate rice in 2010/11 and wanted to secure a whole year's worth of rice for their own consumption.*

What unites these two types of subjective views is that they both have a potential to impact human behavior, which can in turn impact complex systems that need to be modeled. For simplicity, we refer to both of them as *beliefs* in this paper. For a comparison between our definition of beliefs and that in other datasets and related tasks (e.g., opinion mining and stance detection), see Section 6.

In this effort, sentences are annotated with respect to a trigger word—a word that can potentially indicate a belief, e.g., *think, feel, hope*, and *want*. The list of trigger words used (further referred to as *known triggers*) was created by modeling domain experts and augmented by the authors during initial data analysis. In cases where a trigger or the sentence can have multiple possible meanings, annotators are encouraged to use the paragraph context of the sentence for disambiguation.

2.2 Annotation Criteria

The guidelines used for the annotation exercise were created in consultation with scientific modeling domain experts based on the needs of the broader modeling project this work is part of.[2]

To be usable for the broader project (further, referred to as *modeling project*), sentences we annotate as beliefs have to satisfy a number of criteria, as detailed below.

Beliefs have to actually be held by some individual or population. That is, we annotate existing beliefs, e.g., *Rice production is considered a supplementary, non-commercial activity in the region.* Based on this criterion, we exclude sentences that contain (a) hypothetical beliefs, (b) variables without values (i.e., a type of belief is mentioned, but it is not stated whether or not the belief is held by anyone), (c) statements about research methodology of individual studies, and (d) recommendations:

(a) *If local actors perceive too much initial risk to invest in their own brands, [...]* (**Cf.** beliefs that are true under some conditions: *If these debts are subsequently collected, they are considered to be income subject to tax.*)

(b) *Willingness to discuss experiences of violence may also differ according to the cultural context.* (**Cf.** *Farmers expressed willingness to [...]*)

(c) *For the purposes of this report, the Northwest Territories and Nunavut was considered one jurisdiction.*

(d) *Farmers should believe that land can be rehabiliated.*

Beliefs have to be specific. We annotate complete beliefs that do not leave ambiguity as to the objects of the belief. Under this criterion, we exclude (e) beliefs that require coreference resolution outside of the sentence and (f) beliefs in restrictive clauses, in which the belief is what helps identify an object instead of a belief being held about an otherwise specified object:

(e) *It is considered the most numerous bird worldwide with population numbers totaling about 1.500 million [...]*

(f) *[This] requires subjective judgments about subgroups which are believed to be present in large numbers.* (**Cf.** non-restrictive clauses, which provide additional information about a known object: *[...] plus marked increases in Asian and Hispanic populations, who prefer rice.*)

---

[2] `https://www.darpa.mil/program/habitus`

Beliefs should not be simply reporting on facts. Beliefs are frequently discussed with a pattern <*Believer*> *said* followed by a subjective judgment, e.g.: *They said the chemicals were harmful*. We exclude beliefs that include people merely reporting facts, e.g., (g) in reported speech or (h) as research findings:

(g) *The UN said that 5.2 million people in the northeast remained in urgent need of food assistance.*

(h) *Genesee found that students in early , delayed and late immersion programs displayed no negative effects on the development of their first language*

2.3 Dataset Description

For this dataset, we annotate sentences as containing or not containing beliefs with respect to a given trigger word (when present). We annotate over a collection of scientific publications and reports in PDF format written in English on a number of topics, including education, agriculture, finance, etc. in several countries.

The dataset consists of two partitions. The training partition was annotated via the crowdsourcing platform Mechanical Turk (MTurk)[3] and quality controlled by the authors of this paper (further referred to as *the team*) for adherence to the annotation guidelines. The testing partition was created in collaboration with a modeler domain expert and supplemented with annotations provided by the team.

Each partition contains two main categories of data points: those with and without known belief triggers. In both partitions, the known trigger subset was manually annotated for presence of beliefs in the sentence. In the training partition, the triggerless examples were presumed to not contain beliefs based on the absence of known belief triggers and served as negative examples for the classifier training. In the testing partition, the number of triggerless examples was low enough to be manually annotated as well, so it contains both positive and negative examples. The training partition additionally includes the subset of triggerless examples that was used in our experiments. The statistics on each partition are reported in Table 5 of Appendix A.

At a minimum, each data point in the dataset comes with the sentence annotated, the paragraph and the name of the document that the sentence appeared in, and the annotation field, indicating whether or not the sentence contains a belief. Data points with known triggers also contain a field for the trigger and a separate field for a short text span around the trigger, the latter to specify the location of the trigger within the sentence in case the sentence contains multiple instances of the same trigger. Sentences annotated with MTurk additionally list all annotations that we accepted, i.e., that we did not discard based on annotator-level filtering criteria (see Section 2.4.2).

2.4 Data Collection Procedure

2.4.1 Document Collection and Preprocessing

The documents for annotation were collected in two ways: manually by the modelers involved in the modeling project and by querying the Google API. In the first case, the documents were collected based on their relevance to either the domain of the modeling project (agriculture) or the geographic area of interest (Senegal). This set of documents was used for creating the test set for testing how well the models we train handle the modeling project use case.

In the second case, we extracted documents with information on several countries that contained key terms relevant for the modeling project, e.g., *agriculture* or *rice*; however, since those key words can be mentioned in a number of different contexts, the resulting set of documents ended up being on a variety of topics. We attempted to exclude papers on sensitive topics, e.g., domestic violence, but information like that may still have made it into the dataset if it was present in papers on other topics. The documents collected using Google API were used for creating the training partition.

For preprocessing, we converted the PDF documents to text using a package that combines the Science Parse[4] converter and a set of methods to refine text, e.g., to eliminate words broken between lines, fix encoding issues, and find appropriate paragraph breaks. We processed text using the `processors` library[5] to break it into sentences. We filtered out strings of text that were erroneously tokenized to be sentences with simple heuristics, e.g., filtering by length and excluding uncapitalized and non-letter-symbol-heavy strings. We then ex-

---

[3] `https://www.mturk.com/`

[4] `https://github.com/allenai/science-parse`

[5] `https://github.com/clulab/processors`

tracted potential belief-containing sentences using a set of string-match-based rules that capture sentences containing known belief triggers and stored the sentence and the trigger. We extracted triggerless examples, which we presume to be negative (non-belief-containing) data points, in a similar way: using a rule, we only selected sentences that did not contain known belief triggers. For instance, the following sentence is such a negative example: *Agriculture occupies 44% of the workforce and accounts for 25% of the GDP*.

We attempted to make the training partition thematically varied. To achieve that, we sampled sentences on several topics (education, technology, agriculture, traditions, etc). For every topic, we ranked all available belief trigger sentences by their similarity to the topic and took the top N sentences, with N depending on the sample size needed. The similarity of sentences to topics was calculated using the `SentenceTransformers package`[6] (Reimers and Gurevych, 2019) with the *all-mpnet-base-v2* model.[7] We used the model to encode potential beliefs and the topic names (e.g., education) and calculated the similarity between the belief and the topic embeddings as a dot product.

2.4.2 Annotation

For annotation, we wanted to follow a realistic annotation protocol where crowd sourcing is used to generate the training data and the test data are generated in a controlled environment by domain experts.

Team Annotation. We started the annotation process by annotating a set of sentences with guidance from a domain expert, which allowed us to decide on the initial guidelines and create the modeling project domain test set. The test set was later supplemented with additional annotations from team members. Before working on the task, the team annotators were asked to complete a series of qualification tasks of 20 sentences each followed by feedback, until their Cohen's kappa annotator agreement score calculated against the answer key reached the higher bound of moderate agreement or higher. Moderate agreement, considered to be in the 0.41– 0.60 range, was deemed sufficient for this task given its complexity.

Annotators were encouraged to provide comments along with annotations, which aligns with field recommendations (see, for instance, Plank 2022).

Mechanical Turk Annotation. The training partition of the dataset was annotated using the crowdsourcing platform Mechanical Turk with additional quality control by the team members. For every data point, we collected annotations from three MTurk workers. Before starting the task, workers had to read through the guidelines and pass a short qualification task, which covered points of potential confusion, with a score of 90% or higher, which allowed one incorrect answer. To qualify, workers also had to be over 18 years old, located in the U.S., and had completed at least 100 assignments with at least 97% assignment acceptance rate. The workers were compensated at 5 cents per data point (i.e., for evaluating one sentence as containing or not containing a belief).

With the task being highly subjective, we do not have many ways to eliminate possible bad faith annotators other than the qualification task and the annotator statistics filters. However, we removed a small number of annotations that were provided by workers that marked every sentence as containing a belief. We additionally removed the annotations provided by workers who annotated fewer than 10 sentences as they may not have had enough exposure to the task.

2.5 Annotation Issues

2.5.1 Mechanical Turk Quality Control

We evaluated the MTurk annotations by asking two team members to provide their feedback on subsets of annotations. With about 50% of annotations requiring correction in order to align with the guidelines, we chose to proceed with manual quality control of crowd-sourced data.

During quality control, a team member read the sentence and, when needed for disambiguation, the paragraph, and marked their agreement with the MTurk annotation. In complicated cases, a team member provided feedback and had the option to request a second opinion from another team member. In cases of disagreement, a third team member was available as well. Overall, 47% of the labels we assigned during quality control did not match those assigned based on majority vote on MTurk annotations.

Even with the need for quality control, we still collected the data through MTurk for several rea-

---

[6] `https://www.sbert.net/`

[7] `https://huggingface.co/sentence-transformers/all-mpnet-base-v2`

sons. First, at a rate of about two data points a minute, quality control took less time for team members than providing annotations from scratch. Second, we found quality control task to be less mentally taxing than annotating from scratch. Finally, we believe that annotations from MTurk workers, although frequently misaligned with the guidelines, provide some useful signal that helps the quality controller to make a faster decision regarding the label to assign. For experimental support for quality control, see Section 4.1.

2.5.2 Task-specific Difficulties

As mentioned above, we believe the task of belief annotation is complicated because it is inherently subjective. Both triggers and sentences can have multiple meanings and be open to interpretation. In the example below, a positive connection between use of a fertilizer and the health of a plant is discussed, but this case can be interpreted as either the farmers believing it or stating facts. In cases like this, we err on the side of over-annotating beliefs:

> *[...] and some farmers apply urea (called 'salt'), saying that leaf color becomes healthy.*

Additionally, sentences can be seen as containing or not containing beliefs based on the context in which they appear. The sentence below in bold, without the context, can be interpreted as a belief held by an organization. However, the broader context indicates that this is not a belief held by any population, but a study-specific definition introduced by the researcher:

> *I therefore consider that the global rice VC is part of the context, and I do not make it the focus of the research.* ***Nevertheless, since importers are involved through government intervention in trading the rice produced in Senegal, they are considered as part of the domestic VC.***

2.5.3 Human Factor

Team annotation. In many cases, especially in the more complicated ones, team annotators provided comments on their annotations, both during initial annotation and quality control. These comments helped us pinpoint a few issues that may arise during annotation exercises.

The guidelines that were provided to the team annotators were quite extensive. From the comments, we learn that different annotators focused on different aspects of the guidelines. This can be illustrated by the following example:

> *Because of their precarious employment conditions, they are considered to be in "vulnerable" types of employment.*

During the quality control, two annotators discussed via comments the meaning of the sentence while deciding on whether or not it should be annotated as containing a belief, disregarding the guideline to exclude sentences where belief is not complete (we do not know who *they* refers to).

Similarly, while explaining why they did not annotate some sentences as beliefs, some annotators kept listing the same criterion (e.g., completeness or clarity on the believer) as a reason for multiple, unrelated cases.

Mechanical Turk. During several rounds of team annotations, the proportion of sentences annotated as beliefs mainly remained in the 30–50% range. However, we observed high level of variation in the proportion of belief annotations between MTurk workers (59% mean with a standard deviation of 22%, a minimum of 11%, and a maximum of 94%). This could be an indicator of either the difficulty of the task, inadequacy of the guidelines (e.g., not informative enough or overly detailed and, therefore, not read in full), or bad-faith annotation. Another indicator pointing to possible bad-faith annotation is marking beliefs in sentences with belief triggers used in the meaning clearly not stating a belief, e.g., the known belief trigger *think* in the collocation *think tank* or the trigger *trust* in *partnerships, joint ventures, and trusts*.

3 Belief Identification

3.1 The Model

We use our dataset to fine-tune a model mimicking the task performed by the annotators: the model is intended to *provide a binary label indicating whether or not a given sentence contains a belief.* We start with the pretrained BERT model (Devlin et al., 2018) and fine-tune it for the task using the MTurk-annotated examples with known triggers from the training partition combined with a sample of triggerless examples four times the number of sentences annotated as beliefs. We run fine-tuning for 20 epochs, with a batch size of 16 and weight decay of 0.01. We do not do any hyper-parameter tuning.

3.2 Evaluation

We evaluate the performance of the model in two ways. We use cross validation ($k = 5$) to evalu-

Model	**P**	**R**	**F1**
In-domain	$0.68_{\pm 0.05}$	$0.73_{\pm 0.07}$	$0.7_{\pm 0.02}$
Out-of-domain	$0.77_{\pm 0.03}$	$0.80_{\pm 0.03}$	$0.78_{\pm 0.02}$

Table 1: Performance (means and standard deviations) of the belief identifier on cross-validation (in-domain) and the test set aligned with the goals of the modeling project (out-of-domain).

ate model performance in-domain, that is, to evaluate its performance on the data from the same distribution it is trained on—the quality-controlled MTurk data. We further test the model on the out-of-domain test set—the set annotated by the team in collaboration with a modeling domain expert—after training a model on all the training data available. In both cases for all experiments, we report means and standard deviations; for cross-validation, they are calculated over k folds, and for the evaluation of the full model on the test set, they are calculated using bootstrap resampling.

We note that here we use the term *domain* loosely since there may be a thematic overlap between the two sets. What the two sets differ in is that the training set is expected to have more thematic variety and was annotated in a different way. The results are reported in Table 1 as precision, recall and F1 score for the positive label.

3.3 Error Analysis

For error analysis, we manually analyze the sentences that were marked as incorrect during cross-validation evaluation in one of the five cross-validation partitions (folds). We also use the `lime` package[8] (Ribeiro et al., 2016) to analyze how the model assigns weights to features.

False Positives. By using `lime`, we learn that the model is learning to pay attention to the words related to our set of known belief triggers (Figure 1, Appendix B). However, it does not always successfully disambiguate multiple meanings of the triggers. In Example 3 (Figure 1), the model successfully learns a previously unknown (i.e., not used during training as a known trigger) trigger *enjoy*, but fails to pick up on its less frequent meaning *have* as in *enjoying a competitive advantage* and falsely predicts the sentence as containing a belief.

Some false positive predictions turn out to not be false, but result from the fact that during training we make an assumption that examples with no known belief triggers do not contain beliefs. Instead of being incorrect, these examples demonstrate that the model is able to generalize and find new belief triggers (e.g., *aspiring* in the example below), which can later be used for belief extraction:

> *When asked about the type of job they would like, more than 80 percent of those currently employed in agriculture, indicate to be aspiring a job outside agriculture.*

Some types of errors stem from the decisions that we made for the annotation exercise that may need to be reconsidered. For instance, the model predicts sentences requiring out-of-sentence coreference resolution as beliefs. This tells us that imposing artificial constraints on annotations with the desire to simplify the task may not be feasible:

> *The youth of today understand this–think about courageous young people like Greta Thunberg and others like her.* (**Note:** the pronoun *this* is unresolved, i.e., we do not know what the youth understand).

False Negatives. Based on the analysis of false negatives, we believe that the model learned several incorrect heuristics for belief identification:

- possible anti-modal verb bias (Figure 2 of Appendix B), which could be explained by the fact that we avoid hypothetical beliefs, but applies even when modal verbs are not modifying a belief;
- possible anti-long sentence bias (Figure 2 of Appendix B), with long sentences potentially providing more opportunity for certain non-belief terms to appear, skewing the prediction;
- possible anti-first person bias (Figure 3 of Appendix B)—since we mainly focus on reported beliefs, the model may learn that the word *we* is an indicator of non-beliefs.

General Observations. In both false positive and false negative cases, we find that some examples were possibly mislabeled by annotator, most likely because of either possibly conflicting interpretations of the guidelines (e.g., Example 3 in Figure 1, which could be interpreted as either an attitude or reporting facts) or because of the complicated structure of the sentence:

> *In December, 44 people arbitrarily detained for what local NGOs considered to have been*

---

[8] https://github.com/marcotcr/lime

Amnesty International Report 2017/18 395 politically motivated reasons were released [...] (**Note:** the use of pronoun *what* can lead to a false conclusion that the sentence requires out-of-sentence coreference resolution.)

Importantly, from the `lime` analysis, we observe that the model does learn new, previously unknown potential belief triggers, e.g., *likely*, *enjoy*, and *problematic*.

3.4 Discussion

The model performs better on the out-of-domain (i.e., team annotated data developed together with the modeling project domain expert) test set (Table 1). This could be explained by an existing thematic overlap between the train and test data, the lower number of topics in the test set, and the fact that in the test set, both known and unknown trigger examples are annotated, which means there can be no false positive predictions based on unknown triggers.

From the error analysis, we learn that in some cases, e.g., when annotations show that multiple interpretations are possible for a sentence, human label variation is to be expected and should be embraced as it can help guide the development of annotation guidelines. Human label variation should also be taken into account when evaluating systems: as discussed in literature (e.g., Plank, 2022) and seen from our manual error analysis, doing evaluation only on hard labels may not be informative.

We can add that providing rationale during quality control also helped with error analysis since it made it possible to determine the meaning of the sentence and the issues that could arise without rereading the whole sentence and paragraph.

4 Additional Experiments

4.1 MTurk Annotation Threshold

Along with mitigating annotation quality issues with manual quality control, we explored the possibility of automatically cleaning the original MTurk annotations. For every data point, after filtering out suspected bad faith annotators, we had between two and three MTurk worker annotations. From our evaluation of one of the MTurk trial runs, we observed that about 65% of sentences annotated as beliefs by three annotators were judged by the quality controller to indeed be beliefs, while it was about 25% for sentences annotated as beliefs by only one or two annotators.

Partition	Setting	**P**	**R**	**F1**
CV	MTurk0.5	$0.72_{\pm0.06}$	$0.82_{\pm0.08}$	$0.76_{\pm0.02}$
	MTurk1.0	$0.41_{\pm0.07}$	$0.49_{\pm0.09}$	$0.44_{\pm0.04}$
	MTurkQC	$0.68_{\pm0.05}$	$0.73_{\pm0.07}$	$0.7_{\pm0.02}$
Test	MTurk0.5	$0.54_{\pm0.03}$	$0.87_{\pm0.02}$	$^{*}0.67_{\pm0.02}$
	MTurk1.0	$0.54_{\pm0.04}$	$0.42_{\pm0.03}$	$0.47_{\pm0.03}$
	MTurkQC	$0.77_{\pm0.03}$	$0.8_{\pm0.03}$	$^{*}0.78_{\pm0.02}$

Table 2: Performance of the models trained on different versions of the MTurk data (CV: cross validation, in-domain performance. Test: test partition, out-of-domain performance). On the team-annotated test set, the quality controlled data model (MTurkQC) significantly(*) outperforms the next best model, which used the original MTurk data with a belief label majority vote threshold of 50% (MTurk0.5). The MTurk0.5 model outperforms the other two models on cross-validation evaluation.

With that in mind, we conducted an experiment to evaluate which belief-annotation-proportion threshold results in best performance of the model and whether using the original MTurk data can compete with the quality controlled version. We try two thresholds: 1.0 (100%), with all the available annotators agreeing that a sentence contains a belief, and 0.5 (50%), with at least half the annotators making that judgment. Same as with the belief identifier model trained on the quality-controlled MTurk data, we evaluate the models trained with original MTurk data in-domain (using cross validation) and out-of-domain (using the test partition for evaluating the model trained on all the training data available). The results appear in Table 2.

On the out-of-domain test set (*Test* in Table 2), neither of the two threshold conditions result in performance surpassing that of the model trained on quality controlled MTurk data, with the second best model (threshold of 0.5) still performing significantly[9] worse than the best model ($p < 0.001$).

We note that the 0.5 threshold model performs better than the other two during cross-validation (*CV* in Table 2). However, given its performance on the testing partition, we believe that the high cross-validation performance could be an indicator of consistent noise present in the training data. One way to address this, other than with manual quality control, is to work on improving the guidelines provided to MTurk workers before collecting additional data. In the meantime, we believe these results support the need for quality control.

---

[9] Statistical significance is calculated using bootstrap resampling with 10000 samples.

Model	P	R	F1
Unmarked trigger	$0.68_{\pm 0.05}$	$0.73_{\pm 0.07}$	$0.7_{\pm 0.02}$
Marked trigger	$0.72_{\pm 0.06}$	$0.72_{\pm 0.05}$	$0.72_{\pm 0.05}$

Table 3: Performance of the belief identifier during cross-validation ($k = 5$) over the training dataset—quality-controlled MTurk with two trigger marking conditions.

Model	P	R	F1
Unmarked trigger	$0.77_{\pm 0.03}$	$0.8_{\pm 0.03}$	$0.78_{\pm 0.02}$
Marked trigger	$0.81_{\pm 0.03}$	$0.74_{\pm 0.03}$	$0.77_{\pm 0.02}$

Table 4: Performance of the belief identifier on the team-annotated modeling-project-based partition with two trigger marking conditions. No statistical significance between the two configurations was observed.

4.2 Marked Trigger Experiment

After testing the efficacy of the model at predicting beliefs given a sentence, we tested whether using another piece of information available—the trigger—would improve the model performance. We test that by marking the trigger with special tokens **<t>** at the beginning of the trigger and **<\t>** at the end of the trigger (e.g., "... he **<t>**believes**<\t>** that…"). The data was formatted the same way in the marked trigger experiment as it was in the unmarked trigger experiment, with the exception of the special tokens marking the trigger.

The results of the experiment are in Tables 3 (cross-validation performance during training) and 4 (test set performance). In cross-validation, the marked trigger model demonstrates higher performance than the unmarked trigger model, with the difference especially prominent in terms of precision. This could mean that marking known triggers while training the belief identifier can be beneficial. However, the unmarked trigger model does slightly better on the test set, although the difference is not statistically significant.

We also experimented with using predicted instead of extracted triggers; however, the performance of the trigger classifier has not yet been high enough to test it in the belief identifier. See Appendix C for details of the experiment.

5 Application

We are in early stages of using the the belief identifier for the modeling project. We rank beliefs based on similarity to topics and provide them to modeling experts. While automatically identified beliefs, expressed in natural text, cannot be directly fed into models, they can inform modelers' decisions on what parameters to include in models and how to weigh them. So far, the work on belief identification has been met with enthusiasm since, at a minimum, we can save modelers time by surfacing the information that they would normally need to manually search for. We are working on ways to improve the quality of the belief identifier as well as to make the information regarding population beliefs that we provide more systematic.

6 Related Work

Handling noisy data. Noise in annotated data is a common issue discussed in literature, with recent work focusing on embracing it during modeling and evaluation (Davani et al., 2022, Fornaciari et al., 2021, Plank, 2022). Chen et al. (2022) describe a different approach—data cleaning, or targeted relabeling,—in which they use a large portion of the annotation budget to build a model and preserve the remaining budget to relabel the examples that the model gets wrong because those are more likely to be incorrect. For a comprehensive overview and recommendations on handling annotator disagreement, see Plank (2022).

Our approach of using quality control is more similar to that of Chen et al.: with a rather limited number of data points available for training (about a thousand) and between two and three annotations per data point, modeling uncertainty did not seem feasible. Additionally, we had reasons to believe that some annotation variation came from annotator- and guidelines-related issues (see Section 2.5.3) and not from the inherent subjectivity of the task, in which case uncertainty would need to be modeled.

Belief annotation. We are not aware of any datasets that handle beliefs the way we do; however, there exist datasets that focus on beliefs, but define and annotate them from a different perspective. Most recently, Tracey et al. (2022) released *BeSt*, the corpus of beliefs and sentiment, which is concerned with capturing agents' cognitive states. The authors equate belief with factuality and annotate data in terms of whether or not the author believes the described events to be true.

Tracey et al. (2022) provide a detailed summary on related datasets. Since their corpus shares many properties with other related datasets, we will use it as a point of comparison with our work.

While Tracey et al. focus on the authors evaluation of truthfulness of described events and distinguishes between committed and not committed beliefs (the author believes the events are true vs. the author thinks they are true, but is not certain), we target both committed and non-committed beliefs—the level of certainty of the agents does not impact how we annotate or use beliefs.

Tracey et al. are interested in author beliefs, while the main focus of our work is what Tracey et al. and Prabhakaran et al. (2015) refer to as *reported beliefs*—the beliefs reported by the author of the text but held by someone else. This type of beliefs is most likely to identify beliefs of some population—which is what we are interested in capturing—while author beliefs could be idiosyncratic and not representative of beliefs of a population. However, in certain cases we annotate author beliefs as well if the author identifies themself as being affiliated with some population:

> *We here in Germany think that we may have risked too much [...]*

Tracey et al. annotate full text, while we aim to locate reported beliefs. Due to sparsity of reported beliefs, annotating full text is not likely to result in the highest number of annotations of the type we are mainly interested in.

Our work is also related to work on opinion mining, or sentiment analysis (Wankhade et al., 2022), and stance detection (Mohammad et al., 2016). Both overlap with our work in how they target people's subjective views (*opinions* and *stance*, closely related to what we refer to as *attitudes*). However, while opinion mining and stance detection focus on subjective view gradation (*positive, negative, or neutral* in opinion mining and *against, neutral*, or *in favor* in stance detection), we are interested in the presence or absence of a subjective view in a given sentence without evaluating properties of the view, with what the view is being much more open and not forced into a Likert scale. Additionally, opinion mining and stance detection work on author views, such as, student feedback (Shaik et al., 2023), consumer product reviews (Kumar et al., 2016), and tweets (Glandt et al., 2021, Mohammad et al., 2016). We, on the other hand, are mainly interested in reported (third person) views with only occasional cases of first person narration included in the dataset.

7 Limitations and Future Work

While our dataset shows promise based on the models we train with it, at about 1000 annotated examples in the training partition, it is relatively small. Before working on increasing the size of the dataset, we need to work on improving the guidelines provided to Mechanical Turk workers and finding more robust ways of excluding bad faith annotators.

The dataset also currently misses some information that could be useful, e.g., polarity, beliefs involving out-of-sentence coreference resolution, as well as believer and belief span annotations. We plan to address all of these in future work.

8 Conclusion

In this paper, we create a dataset of subjective views of populations and test it by training and experimenting with a belief identifier model. We discuss the issues related to annotation and human label variation that we encountered during the annotation exercise such as the difficulty of creating guidelines for a subjective task and lack of certainty in annotators adhering to the annotation guidelines for various reasons (e.g., annotators focusing on different aspects of the guidelines or not annotating in good faith). We also compare two ways of managing human label variation—annotation quality control vs. majority voting with different thresholds—by evaluating a model performance under the two conditions. Finally, we provide support for the idea that human language variation should be embraced by doing an error analysis of the model predictions, which shows how language ambiguity as well as human factor and guidelines-related issues make it impossible to rely strictly on majority voting without qualitative analysis while evaluating systems working on subjective tasks.

9 Acknowledgments

The authors thank the anonymous reviewers for helpful discussion. This work was supported by the Defense Advanced Research Projects Agency (DARPA) under the Habitus program. Maria Alexeeva and Mihai Surdeanu declare a financial interest in lum.ai. This interest has been properly disclosed to the University of Arizona Institutional Review Committee and is managed in accordance with its conflict of interest policies. The annotation work was partially supported through Research and Project (ReaP) Grant from the University of Arizona Graduate and Professional Student Council (GPSC).

References

Derek Chen, Zhou Yu, and Samuel R. Bowman. 2022. Clean or annotate: How to spend a limited data collection budget. In *Proceedings of the Third Workshop on Deep Learning for Low-Resource Natural Language Processing*, pages 152–168, Hybrid. Association for Computational Linguistics.

Aida Mostafazadeh Davani, Mark Díaz, and Vinodkumar Prabhakaran. 2022. Dealing with disagreements: Looking beyond the majority vote in subjective annotations. *Transactions of the Association for Computational Linguistics*, 10:92–110.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Hubert L Dreyfus. 1990. *Being-in-the-world: A commentary on Heidegger's being in time, division I*. Mit Press.

Tommaso Fornaciari, Alexandra Uma, Silviu Paun, Barbara Plank, Dirk Hovy, and Massimo Poesio. 2021. Beyond black & white: Leveraging annotator disagreement via soft-label multi-task learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2591–2597, Online. Association for Computational Linguistics.

Kyle Glandt, Sarthak Khanal, Yingjie Li, Doina Caragea, and Cornelia Caragea. 2021. Stance detection in COVID-19 tweets. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1596–1611, Online. Association for Computational Linguistics.

K L Santhosh Kumar, Jayanti Desai, and Jharna Majumdar. 2016. Opinion mining and sentiment analysis on online customer review. In *2016 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, pages 1–4.

J. A. Meaney, Steven Wilson, Luis Chiruzzo, Adam Lopez, and Walid Magdy. 2021. SemEval 2021 task 7: HaHackathon, detecting and rating humor and offense. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 105–119, Online. Association for Computational Linguistics.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. Semeval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 31–41.

Barbara Plank. 2022. The "problem" of human label variation: On ground truth in data, modeling and evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10671–10682, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Dean Pomerleau and Delip Rao. 2017. Fake news challenge. *Exploring how artificial intelligence technologies could be leveraged to combat fake news. url: https://www. fakenewschallenge. org/(visited on 03/13/2020)*.

Vinodkumar Prabhakaran, Tomas By, Julia Hirschberg, Owen Rambow, Samira Shaikh, Tomek Strzalkowski, Jennifer Tracey, Michael Arrigo, Rupayan Basu, Micah Clark, Adam Dalton, Mona Diab, Louise Guthrie, Anna Prokofieva, Stephanie Strassel, Gregory Werner, Yorick Wilks, and Janyce Wiebe. 2015. A new dataset and evaluation for belief/factuality. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 82–91, Denver, Colorado. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144.

Thanveer Shaik, Xiaohui Tao, Christopher Dann, Haoran Xie, Yan Li, and Linda Galligan. 2023. Sentiment analysis and opinion mining on educational data: A survey. *Natural Language Processing Journal*, 2:100003.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Jennifer Tracey, Owen Rambow, Claire Cardie, Adam Dalton, Hoa Trang Dang, Mona Diab, Bonnie Dorr, Louise Guthrie, Magdalena Markowska, Smaranda Muresan, Vinodkumar Prabhakaran, Samira Shaikh, and Tomek Strzalkowski. 2022. BeSt: The belief and sentiment corpus. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2460–2467, Marseille, France. European Language Resources Association.

Mayur Wankhade, Annavarapu Chandra Sekhara Rao, and Chaitanya Kulkarni. 2022. A survey on sentiment analysis methods, applications, and challenges. *Artificial Intelligence Review*, 55(7):5731–5780.

A Dataset Statistics

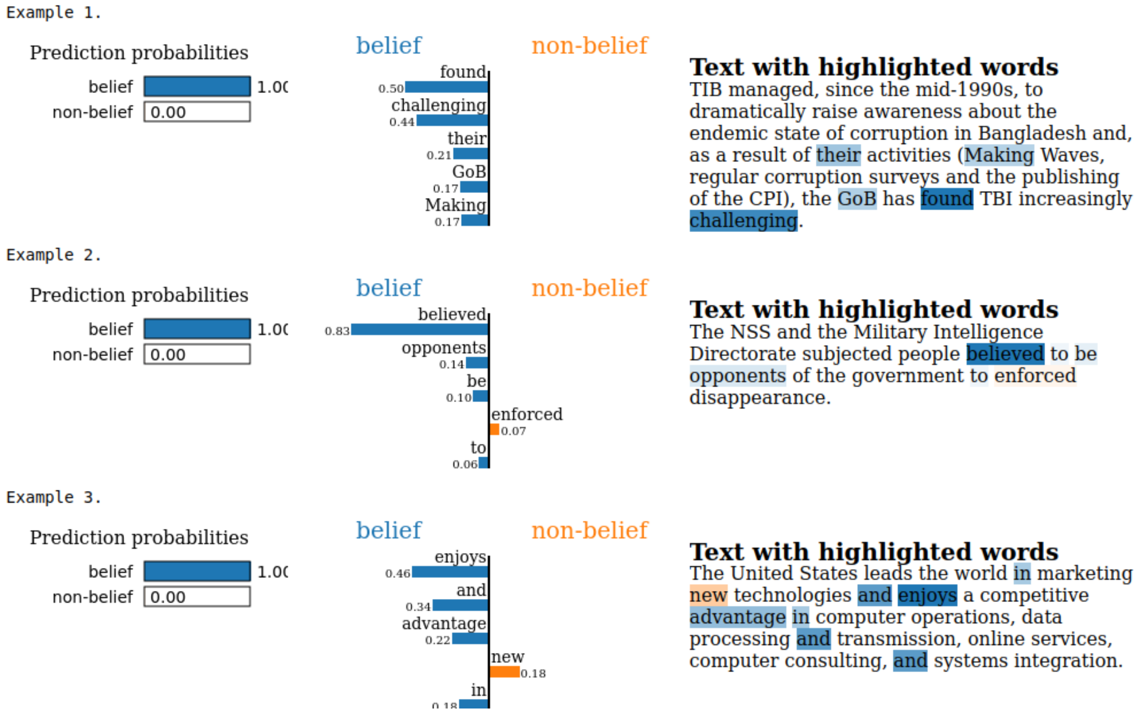
In Table 5, we report the details on the dataset composition, including the number of data points, documents, known triggers, etc.

B Error Analysis with `lime`

Figures 1–3 illustrate some common errors that were discovered during error analysis using the `lime` package (see Section 3.3).

Measure	Train			Test	
	known triggers	unk. triggers	unk. in training	known triggers	unk. triggers
N documents	59	65	65	50	43
N data points	1044	9769	1440	400	193
N positive class	360	0*	0*	202	12
% positive class	34%	0*	0*	50.5%	6%
Unique triggers	95	N/A	N/A	72	12

Table 5: Dataset statistics. For the training partition unknown trigger subset, we release all available data points as well as the subsample used for the experiments. Asterisk (*) indicates values assumed based on absence of known belief triggers in the sentence.

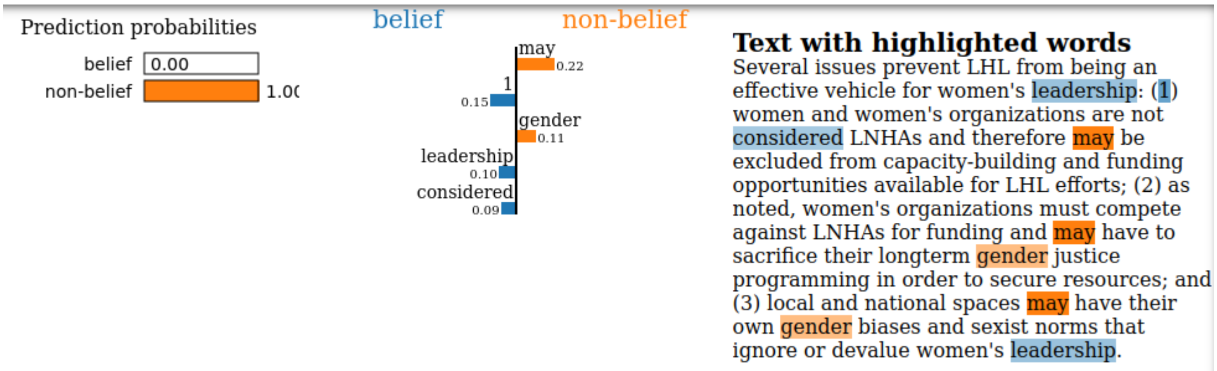


TIB managed, since the mid-1990s, to dramatically raise awareness about the endemic state of corruption in Bangladesh and, as a result of their activities (Making Waves, regular corruption surveys and the publishing of the CPI), the GoB has found TBI increasingly challenging.

The NSS and the Military Intelligence Directorate subjected people believed to be opponents of the government to enforced disappearance.

The United States leads the world in marketing new technologies and enjoys a competitive advantage in computer operations, data processing and transmission, online services, computer consulting, and systems integration.

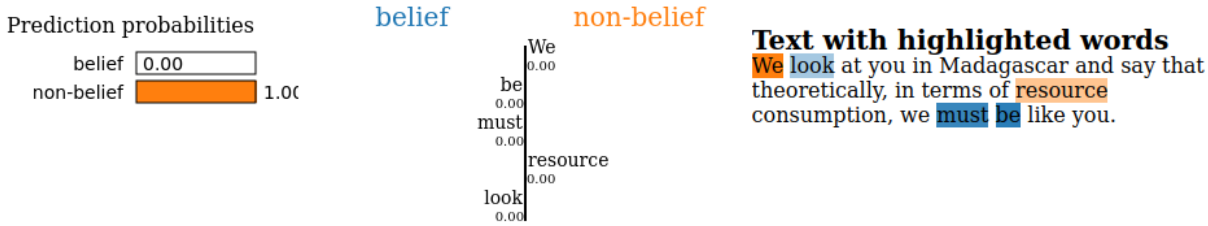
Figure 1: `lime` analysis of false positive examples from the belief identifier model trained on quality controlled MTurk data. The model learns and makes decisions on words that appear to be good quality belief triggers, but also includes some noise—the words that could occur in both beliefs and not beliefs, e.g., *their*, *and*, and *advantage*.



Text with highlighted words

Several issues prevent LHL from being an effective vehicle for women's leadership: (1) women and women's organizations are not considered LNHAs and therefore may be excluded from capacity-building and funding opportunities available for LHL efforts; (2) as noted, women's organizations must compete against LNHAs for funding and may have to sacrifice their longterm gender justice programming in order to secure resources; and (3) local and national spaces may have their own gender biases and sexist norms that ignore or devalue women's leadership.

Figure 2: `lime` analysis of a false negative example in which the model incorrectly judges a sentence with a large number of instances of the modal verb *may* as non-belief. The possible reason is that we aim to avoid hypothetical beliefs, which eliminates a lot of belief triggers accompanied by modal verbs. The sentence can also illustrate the anti-long sentence bias, where the model tends to not annotate long sentences as beliefs.



Text with highlighted words

We look at you in Madagascar and say that theoretically, in terms of resource consumption, we must be like you.

Figure 3: `lime` analysis of a false negative example in which the model incorrectly judges a sentence with the first person pronoun *we* as non-belief. This could be happening because the dataset the model was trained on (quality controlled MTurk) focuses on reported (non-author) beliefs, so the word *we* does not get associated with beliefs.

C Extracted Triggers vs. Predicted Triggers

We wanted to test if a classifier could be trained to predict the trigger words as opposed to the current approach, which searches text for a pre-selected list of trigger words before running the sentences through the belief identifier. The classifier could be helpful in multiple ways: it could help identify new triggers, avoid the need to extract triggers before running the marked belief version of the belief identifier, and potentially improve performance of the belief identifier by marking previously unknown triggers in sentences that do not have any known ones.

We trained a classifier to label each word in a sentence as either "n" for "not a trigger", "tb" ("trigger beginning") for the first token of the trigger, or "tc" ("trigger continued") for subsequent tokens in a multi-word trigger phrase. The predicted triggers were to be added to the dataframe used for training the belief identifier model. Based on the initial experiments, we judged the performance of the trigger classifier, with only about 25% of triggers correctly identified, not to be high enough for us to proceed with the predicted trigger belief classifier experiment. We will continue the work on improving the trigger prediction model.

Exploration of Contrastive Learning Strategies toward more Robust Stance Detection

Udhaya Kumar Rajendran
Lakehead University
Department of Computer Science
rajendranu@lakeheadu.ca

Amine Trabelsi
Université de Sherbrooke
Department of Computer Science
amine.trabelsi@usherbrooke.ca

Abstract

Stance Detection is the task of identifying the position of an author of a text towards an issue or a target. Previous studies on Stance Detection indicate that the existing systems are non-robust to the variations and errors in input sentences. Our proposed methodology uses Contrastive Learning to learn sentence representations by bringing semantically similar sentences and sentences implying the same stance closer to each other in the embedding space. We compare our approach to a pretrained transformer model directly finetuned with the stance datasets. We use char-level and word-level adversarial perturbation attacks to measure the resilience of the models and we show that our approach achieves better performances and is more robust to the different adversarial perturbations introduced to the test data. The results indicate that our approach performs better on small-sized and class-imbalanced stance datasets.

1 Introduction

A controversial topic divides people into two groups with different views (support/against) on the topic of discussion. Some popular, controversial topics include the Legalization of Abortion, Concern about Climate Change, Gay Marriage, Obama, the Legalization of Marijuana, Feminism, and Atheism. The existing Stance Detection models are non-robust, and even simple perturbations in the input sentences affect the model's performance (Schiller et al., 2021). For example, the input sentence 'Fetus is not human' has the stance label of 'support' for the topic of 'Legalization of Abortion.' However, when there is a variation to the same input sentence, such as 'A bunch of cells is not human,' it will confuse the model in reproducing the same stance label of 'support.' Also, spelling errors, missing words, repetition of words, and other commonly occurring errors in the text are the adversarial errors that make the Stance Detection models fall short in detecting the stance compared to humans. We aim to make the Stance Detection system more robust to adversarial perturbations by accommodating the variations and errors in the text when detecting the stance. We primarily concentrate on binary stances (e.g., support/against) in social media for English texts, such as tweets, news comments, and discussion forums. We use the Contrastive Learning (CL) approach to construct more robust sentence representations for the Stance Detection task. Given an example we call anchor, the CL technique brings the similar example closer to the anchor and drives the dissimilar example away from the anchor in the representation space. We build similar (positive) and dissimilar (negative) examples for CL by considering the stance label of the examples. We mainly explored different strategies for building positive and negative examples for an anchor example to learn the sentence representations in a contrastive fashion. Along with CL, we use Masked Language Modeling as a token-level objective to learn textual representations (see Figure 1). Our code is available in the GitHub repository [1]. We make the following contributions.

- We develop an approach using a CL framework with different positive and negative pairs selection strategies to learn more robust sentence representations to use in the Stance Detection task. To the best of our knowledge, this work is the first to employ a Contrastive Learning framework to learn robust sentence representations in the context of Stance Detection task.

- We conduct a comprehensive empirical investigation using various settings and datasets for stance detection, analyzing the results and pro-

---

[1] https://github.com/rajendranu4/stance-detection

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 431–440
July 14, 2023 ©2023 Association for Computational Linguistics

viding valuable insights into effective strategies for different contexts.

2 Related Work

Many approaches (Darwish et al., 2017; Matero et al., 2021; Zhang et al., 2021; Landwehr et al., 2005; Sobhani et al., 2017; Aldayel and Magdy, 2019; Rashed et al., 2020; Lai et al., 2020; Liang et al., 2022) were proposed to tackle different problems in the Stance Detection task. However, the existing Stance Detection models are sensitive to adversarial errors, and changes in the vocabulary of the input sentences (Schiller et al., 2021).

The adversarial robustness of the model is measured by making the model predict against the test set with char-level, and sequence-level modifications to the input as well as with the word substitutions (Dong et al., 2021; Zhang et al., 2022; Wang et al., 2020). Moradi and Samwald (2021) used various perturbations for Char-level such as Insertion, Deletion, Replacement, etc., and word-level perturbations such as Replacement with Synonyms, Negation, etc. Schiller et al. (2021) used the resilience score introduced by Thorne et al. (2019) to measure the robustness of the model.

CL is used to acquire better representations of text for many natural language tasks such as Question-Answering (Yue et al., 2021), multiple choice video questions, text-to-video retrieval (Xu et al., 2021), text summarization (Wu et al., 2020a; Du et al., 2021; Cao and Wang, 2021) etc. Wu et al. (2020b) used Contrastive Learning to learn noise invariant sentence representation with the help of different sentence-level augmentation strategies like span deletion, substitution, and reordering. Liang et al. (2022) introduced a hierarchical contrastive learning strategy to improve the Zero-shot Stance Detection (ZSSD) task by capturing the relationships not only between target-invariant and target-specific features but also among various stance labels.

In this study, our objective is to develop and explore a range of strategies encompassed within contrastive learning. Our aim is to enhance the quality of document representations specifically for the task of stance detection, consequently bolstering the robustness of stance detection classification models.

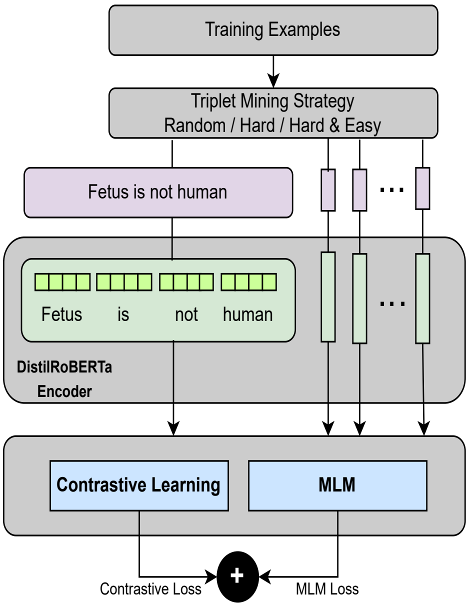


Figure 1: Architecture diagram for learning sentence representations using CL and MLM objectives to further use in the Stance Detection task.

3 Method

3.1 Contrastive Learning

Contrastive Learning maps the representations of 'similar' patterns closer to each other while pushing the representations of 'different' patterns farther away in the embedding space. CL learns from the examples that are hard to distinguish in the representation space from the anchor example (Ostendorff et al., 2022). The goal of the contrastive loss function ($\mathrm{loss_{CL}}$) given by Eqn. 1 is to minimize the distance between the anchor-positive pair ($\mathrm{d_A}$, $\mathrm{d_+}$) and to maximize the distance between the anchor-negative pair ($\mathrm{d_A}$, $\mathrm{d_-}$). We use pairwise Euclidean distance measure for calculating the contrastive loss in the Eqn. 1. m in the Equation 1 is the margin and is the desired difference between the anchor-positive and anchor-negative distances. CL makes similar examples have similar representations in the representation space, which makes the language model less sensitive (more robust) to adversarial errors, including changes in the text's vocabulary. For instance, the examples *'Fetus is not human'* and *'Bunch of cells in a woman's womb'* are having the same stance as *support* though the lexicons used in these examples are completely different. The example, *'Really? Fetus is not human?'* is a rhetorical question, having an opposite stance compared to the example *'Fetus is not human'*, however, both these examples are similar in

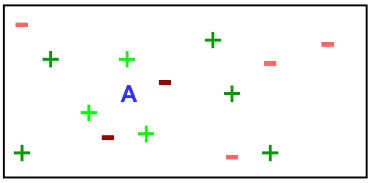


Figure 2: Illustration of Easy Positive + and Negative -, Hard Positive + and Negative - samples for an Anchor sample A in the representation space.

terms of lexicons. The contrastive learning method attempt to make the examples *'Fetus is not human'* and *'Bunch of cells in a woman's womb'* have similar representations by bringing the two examples closer to each other in the representation space. The final loss is the sum of CL loss and MLM loss.

$$loss_{\text{CL}} = max\{|d_{\text{A}} - d_{+}| - |d_{\text{A}} - d_{\text{-}}| + m, 0\} \quad (1)$$

3.2 Contrastive Learning Strategies

We use different strategies to select positives and negatives for an anchor for CL. The combination of anchor, positive and negative, is called a triplet.

Random Strategy The triplets are formed randomly, satisfying the anchor-positive and anchor-negative selections.
Hard Strategy Hard positive (same ground truth label as the anchor but far away from it) and hard negative (different ground truth label from the anchor but close to it) are chosen for an anchor.
H&E Strategy One Hard triplet similar to the Hard strategy and one Easy triplet (easy positive and easy negative) are chosen for an anchor (see Figure 2 for a graphical illustration of the hard and easy positives and negatives for an anchor in the representation space).

3.3 Robustness of Stance Detection Systems

We measure the robustness of the model with the resilience score Res in Equation 2 introduced by Thorne et al. (2019) by identifying the deviation between the performances of the model with the original test set $p(s,t)$, also called as ***non-perturbed test set*** and the adversarial ***perturbed test set*** $p(s,a)$ with adversarial attack a for a natural language system s.

We use three adversarial attacks *spelling errors*, *adding tautology*, and *synonym replacements* (see Table 1). The correctness ratio c_a of an adversarial attack a gives the total number of correctly transformed examples from the number of examples considered for perturbation.
Spelling error. We introduced spelling errors to perturb all the original sentences in the test set. We select two words randomly from a sentence to introduce misspellings by replacing a letter in one of the selected words and by swapping the position of two letters in the other word.
Adding tautology. All the input sentences in the test set are appended with '***False is not true and,***'.
Synonyms replacement. We consider 15 words that are frequent in the test dataset for the Synonyms replacement adversarial attack. We use WordNet (Miller, 1994), a Lexical Database for English, to select the synonyms for the 15 frequent words in the test dataset. We select a maximum of 2 words from a sentence (selected words fall under the frequent words) to replace with their synonymous words which do not change the meaning of the sentences. Since the frequent words are selected for the *synonyms replacement* attack, the words that are selected may or may not be in a given example. Hence not necessarily all the examples are perturbed for the *synonyms replacement* adversarial attack though all the examples are candidates for this attack.

$$Res = \left| \frac{\sum_{a \in A} c_a * (p(s,t) - p(s,a))}{\sum_{a \in A} c_a} \right| \quad (2)$$

3.4 Learning and Leveraging Robust Representations

Let F be the transformer model (DistilRoBERTa), for each of the input sequences $x^{(\text{i})}$ from batch j, the MLM objective masks a percentage of tokens, and the model predicts the masked token with the help of the surrounding tokens. Again, for the same input sequences from batch j, the Contrastive Learning framework identifies the triplets for each $x^{(\text{i})}$ (anchor) based on the strategies explained in Section 3.2. The combined loss (Contrastive Learning + MLM) is backpropagated to adjust the weights of the transformer model. Now the transformer model F trained with the Contrastive Learning and MLM objectives is added with a classification layer on top and finetuned with the stance datasets. Let $P^{(o)}$ be the model's performance after finetuning with the stance dataset D. The robustness of model F is identified by testing the finetuned model F against the perturbed test set D_{p}. Let $P_p^{(se)}$, $P_p^{(n)}$, and $P_p^{(sm)}$ be the performances of the model against the perturbed test sets generated with the

Adv. Attack	Original Sample	Perturbed Sample
Spelling Error	Green is the way **forward**	Green is the way **ferward**
Adding Tautology	The Olympics create a sense of national pride	**False is not True and** the Olympics create a sense of national pride
Synonyms	Golf is one of **independent** sports	Golf is one of **stand-alone** sports

Table 1: Illustration of the different types of adversarial attacks for perturbing the test set to measure the robustness and reliability of the model.

adversarial attacks spelling errors, tautology, and synonyms respectively.

4 Experiments

We have chosen seven Stance Detection datasets, DebateForum (DF) (Hasan and Ng, 2013), SemEval2016 (SE) (Mohammad et al., 2016), ARC (Habernal et al., 2018), Perspectrum (Chen et al., 2019), FNC-1 (Pomerleau and Rao), KSD-Biden and KSD-Trump (Kawintiranon and Singh, 2021) for the experiments. We have retained only the examples that have support/against equivalent labels in the datasets as we mainly focus on binary stances. Out of the seven chosen datasets, the Perspectrum dataset has more instances (11825), KSD-Biden has the least number of instances (766) and FNC-1 is the most imbalanced dataset (78/22). See Table 2 for more information on the statistics of these datasets. Table 4 describes the datasets, the domain of the corresponding datasets, and an example from the dataset to show the input and the stance output.

4.1 Setups

The setups below vary according to the level of information leveraged to train and evaluate the conceived models. To further validate our evaluation of resilience, we only perturbed the instances that were correctly classified (*Partial Perturbation*) by the models from the original test dataset and assessed its resilience in relation to those perturbations.

Mixed Topics. We consider the examples of all topics from a dataset as a whole for the experiments. The evaluation of models is carried out by perturbating all the examples in the test dataset while testing the model against an adversarial attack.

Mixed Topics + Partial Perturbation (PP). The models are constructed based on all topics similar to the *Mixed Topics* setup but the evaluation of models is carried out by perturbing with an adversarial attack only the examples that are correctly classified by the models from the original test dataset run.

Individual Topics. The models are constructed and evaluated based on individual topic-related sub-datasets. We consider topics from DF and SE datasets for this setup (see Table 3).

Individual Topics + Partial Perturbation (PP). The models are constructed based on individual topic-related sub-datasets similar to the *Individual Topics* setup but the evaluation of models is carried out by perturbing with an adversarial attack only the examples that are correctly classified by the models from the original test dataset run.

4.2 Models

We have used the DistilRoBERTa (Sanh et al., 2019) as the transformer model which is twice as fast as RoBERTa-base (Liu et al., 2019) for all our experiments. Inspired by the work of Giorgi et al. (Giorgi et al., 2020), we have used the code architecture and modified the loss objectives and the pipeline according to our experiment setup. The transformer model in our proposed methodology is not pre-trained from scratch. We use DistilRoBERTa pre-trained weights as the initial weights for the DistilRoBERTa model. We compare our proposed models described below with a baseline model.

Model$_{\textbf{Baseline}}$ is pretrained DistilRoBERTa model finetuned with stance datasets.

Model$_{\textbf{Random}}$. Randomly formed triplets from a batch are used in CL.

Model$_{\textbf{Random2}}$. Two random triplets from a batch are used in CL.

Model$_{\textbf{Hard}}$. One Hard triplet is used in CL.

Model$_{\textbf{H\&E}}$. One Hard and one Easy triplets are

Dataset	**# Examples**	**Classes**	**Splits**		
			Train	Dev	Test
DebateForum	4904	for(60%), against(40%)	3431	884	589
SemEval2016	3170	favor(35%), against(65%)	2149	205	816
ARC	3368	agree(47%), disagree(53%)	2660	283	425
Perspectrum	11825	support(52%), undermine(48%)	6979	2072	2774
FNC-1	7121	agree(78%), disagree(22%)	4519	1301	1301
KSD-Biden	766	favor(50%), against(50%)	546	110	110
KSD-Trump	843	favor(41%), against(59%)	591	126	126

Table 2: Statistics about the different datasets used for the experiments

Topic	**Class Ratio**	**# Examples**	**Splits**		
			Train	Dev	Test
Abortion$_{\text{DF}}$	56 / 44	1918	1341	288	289
GayRights$_{\text{DF}}$	64 / 36	1378	963	207	208
Marijuana$_{\text{DF}}$	71 / 29	629	439	95	95
Obama$_{\text{DF}}$	53 / 47	988	690	149	149
Abortion$_{\text{SE}}$	24 / 76	714	498	108	108
Atheism$_{\text{SE}}$	21 / 78	591	412	89	90
Climate$_{\text{SE}}$	90 / 10	364	253	55	56
Feminism$_{\text{SE}}$	35 / 65	782	546	118	118
HillaryClintion$_{\text{SE}}$	23 / 77	730	510	110	110

Table 3: The topicwise distribution of the datasets DebateForum and SemEval2016

Dataset	**Domain**	**Example**	**Topic**	**Stance Label**
DebateForum	Debating Forum	Passive smoking is harmful and secondhand smoke from the use of marijuana increases the chances of others suffering the damage by inhaling the smoke.	Marijuana	against
Arc		This is a great move by Wal-Mart. I hope they take out all the high fructose corn syrup out of their products as well. I avoid anything with high fructose corn syrup and as a result I have lost 37 pounds.	Wal-Mart can make us healthier	agree
Perspectrum		A game is less enjoyable if there is video replay.	There should be video replays for refs in football	undermine
SemEval2016	Social Media	Today Europe is breaking heat records, while Asia is breaking the lowest temperature records!! Should we not be concerned	Climate Change is a Real Concern	favor
KSD-Biden		i miss having a president that speaks eloquently. that has empathy and hope for a better tomorrow. fortunately, we will soon have that again with #bidenharris2020.	Biden	favor
KSD-Trump		not everyone in oklahoma is welcoming the president's visit	Trump	against
FNC-1	News	Tesla is reportedly choosing Nevada for its new battery factory.	Tesla to choose Nevada for Battery Factory	agree

Table 4: Illustrates the domain of the different datasets used for the experiments and an example from each of the datasets

used in CL.

Hyperparameter	Value
Batch Size	8
Epochs	20
Max. Seq. Length	100
Optimizer	Adam
Learning Rate	5e-5
Gradient Clipping	max norm: 1.0
Epsilon	1e-6
Weight Decay	0.1

Table 5: Hyperparameters for the training with CL

Objective	Hyperparameter	Value
MLM	% of tokens masked	15%
CL	Margin (m)	0.5

Table 6: Hyperparameters for the Objectives Contrastive Learning and Masked Language Modeling

Hyperparameter	Value
Batch Size	16
Epochs	4
Optimizer	Adam
Learning Rate	5e-5

Table 7: Hyperparameters for finetuning the DistilRoBERTa model with stance dataset

4.3 Settings

The number of characters and words used in social media posts is usually restricted to cut out the fluff. For example, currently, Twitter (Twitter, 2022) has a character limit of 280 characters per post to express the user's thoughts. In all our experiments, we use a word limit of 100 to capture the valuable meaning of the user's post. To allow maximum participation of different examples in CL, the training batch size is reduced from 16 to 8 as the strategies Hard and H&E mine one and two triplets, respectively, from a batch of examples for CL. All the other hyperparameters for the models are as per the transformer model's predefined values. We train the DistilRoBERTa model using CL (0.5 as margin, m) and MLM objectives (15% tokens masked) for 20 epochs to learn the sentence representations. We then finetune the model with stance datasets for 4 epochs. See Tables 5, 6 and 7 for more details on hyperparameters for pretraining and finetuning.

The Correctness Ratio for the adversarial attack 'adding tautology' is 1 as the data is perturbed by prefixing the example sentence with the words ***False is not True and*** which does not change the truth value of the sentence, hence the stance labels for the sentence remains the same. The Correctness Ratio for the adversarial attack 'synonyms replacement' is also 1 as the words in a sentence are replaced with their synonyms which does not change the sentence's truth value and hence the stance labels for the sentences remain the same. We use Flesch–Kincaid grade level (Kincaid et al., 1975) to check if the transformed sentence with the adversarial attack 'spelling error' is readable. We consider the example after perturbation which has the same readability grade level as the original example as a correctly perturbed example. The Correctness Ratio of adversarial attack 'spelling error' is 1 as all the examples used in the experiments are correctly perturbed for all the datasets.

The resilience of models is measured by perturbing **all** the examples in the test dataset with the adversarial attacks individually for the experiment setups **Mixed Topics** and **Individual Topics**, see under Section 4.1. For the experiment setups **Mixed Topics + PP** and **Individual Topics + PP** , the resilience of the model is measured by making the model predict on the test set in which the perturbations are introduced on the examples that are correctly classified in the original non-perturbed test. For example, the model$_{\text{Hard}}$ is evaluated on the original non-perturbed dataset initially, then a dataset is prepared by perturbing (with an adversarial attack, e.g., spelling attack) only the correctly classified examples from the original non-perturbed test run and finally, the model is evaluated on the prepared dataset to measure the resilience of the model. We consider only the spelling and negation adversarial attacks for the experiments **Mixed Topics + PP** and **Individual Topics + PP** since not all the examples in a given set of examples are perturbed in *synonyms replacement* adversarial attack. The difference in the performance of the models between the original non-perturbed test set and the adversarial test sets is measured to identify the robustness of the model. The percentage of examples perturbed from a given set of examples needs to be consistent across the different adversarial attacks as well as the different models. For example, from the original non-perturbed test set, if Model 1 predicts 60% of the examples correctly

Dataset	Model$_{\text{Baseline}}$	Model$_{\text{Random}}$	Model$_{\text{Random2}}$	Model$_{\text{Hard}}$	Model$_{\text{H\&E}}$
DebateForum	93.24 (64.06)	98.33 (**68.68**)	98.12 (65.73)	98.42 (62.22)	**98.53** (62.97)
SemEval2016	98.31 (**74.04**)	99.24 (72.21)	**99.66** (73.31)	99.5 (71.18)	99.49 (71.27)
ARC	**99.71** (60.94)	98.19 (61.77)	99.02 (**62.97**)	95.92 (62.21)	99.35 (62.25)
Perspectrum	92.91 (65.5)	95.16 (**66.05**)	96.54 (65.81)	95.55 (64.75)	**98.82** (63.15)
FNC-1	93.77 (48.86)	97.61 (**52.87**)	96.58 (52.22)	**99.06** (52.63)	97.95 (52.2)
KSD-Biden	93.32 (82.08)	98.38 (**88.77**)	98.25 (87.87)	**98.47** (85.22)	98.16 (84.21)
KSD-Trump	98.97 (86.95)	**99.72** (**88.81**)	99.19 (82.86)	98.97 (85.97)	98.84 (83.58)
Average	95.74 (68.91)	98.09 (**71.30**)	98.19 (70.11)	97.98 (69.16)	**98.73** (68.51)

Table 8: Resilience and F1-score (within parenthesis) of all the models for all the datasets in ***Mixed Topic*** setup. The F1-scores are reported in % on all the original, non-perturbated datasets. Bold numbers in **Purple** and **Blue** colors indicate the model with the best Resilience score and F1-score respectively

Dataset	Model$_{\text{Baseline}}$	Model$_{\text{Random}}$	Model$_{\text{Random2}}$	Model$_{\text{Hard}}$	Model$_{\text{H\&E}}$
Abortion$_{DF}$	97.22 (67.01)	98.24 (68.39)	**98.92** (65.66)	98.54 (**68.78**)	98.58 (66.29)
Marijuana$_{DF}$	**98.65** (40.14)	97.56 (45.31)	98.55 (42.29)	95.79 (50.94)	96.99 (**53.19**)
Gay Rights$_{DF}$	96.61 (67.14)	95.66 (60.75)	94.1 (60.06)	**98.74** (58.51)	96.85 (**67.75**)
Obama$_{DF}$	98.91 (64.07)	99.10 (**68.2**)	98.59 (68.17)	98.65 (61.48)	**99.64** (64.8)
Abortion$_{SE}$	97.33 (71.39)	**98.89** (74.3)	97.39 (74.59)	96.31 (**81.19**)	96.63 (78.68)
Atheism$_{SE}$	96.23 (77.14)	95.70 (78.18)	95.22 (79.54)	**97.56** (**80.43**)	96.07 (77.14)
Climate$_{SE}$	93.89 (61.81)	79.54 (68.57)	**94.12** (68.57)	91.89 (**82.37**)	90.60 (72.97)
Feminism$_{SE}$	**99.60** (64.32)	93.72 (**65.06**)	99.36 (60.82)	95.51 (62.97)	85.12 (63.97)
Hillary Clinton$_{SE}$	86.17 (**84.63**)	92.49 (82.37)	94.56 (80.3)	**98.19** (71.52)	96.01 (73.46)
Average	96.06 (66.40)	94.54 (67.90)	96.75 (66.67)	**96.79** (**68.69**)	95.16 (**68.69**)

Table 9: Resilience and F1-score (within parenthesis) of all the models for all the datasets in ***Individiual Topic*** setup. The F1-scores are reported in % on all the original, non-perturbated datasets. Bold numbers in **Purple** and **Blue** colors indicate the model with the best Resilience score and F1-score respectively

Dataset	Model$_{\text{Baseline}}$	Model$_{\text{Random}}$	Model$_{\text{Hard}}$	Model$_{\text{H\&E}}$
DebateForum	82.05	90.68	**95.15**	93.37
SemEval2016	88.98	**91.69**	91.16	91
ARC	**96.96**	95.98	95.84	95.86
Perspectrum	95.80	96.26	**96.47**	**96.47**
FNC-1	75.15	79.36	81.62	**86.08**
KSD-Biden	98.19	95.29	**98.62**	97.76
KSD-Trump	**98.96**	97.49	92.97	95.88
Average	90.87 ± 9.2	92.39 ± 6.26	93.12 ± 5.61	**93.77** ± 4.06

Table 10: Reslience of all the models for all the datasets in ***Mixed Topic + Partial Perturbation*** setup. Bold numbers in **Purple** color indicate the model with the best Resilience score. The last row shows the models' average resilience over all datasets including standard deviation.

Dataset	Model$_{\text{Baseline}}$	Model$_{\text{Random}}$	Model$_{\text{Hard}}$	Model$_{\text{H\&E}}$
Abortion$_{\text{DF}}$	90.26	93.32	**95.34**	94.96
Marijuana$_{\text{DF}}$	**98.77**	95.06	93.57	96.32
GayRights$_{\text{DF}}$	88.25	**92.97**	90.19	80.15
Obama$_{\text{DF}}$	92.9	**95.64**	94.92	94.24
Abortion$_{\text{SE}}$	79.08	88.64	**90.03**	87.53
Atheism$_{\text{SE}}$	85.7	**93.59**	90.97	90.96
Climate$_{\text{SE}}$	86.3	96.17	**97.37**	92.08
Feminism$_{\text{SE}}$	**87.15**	79.64	84.84	80.47
Hillary Clinton$_{\text{SE}}$	74.24	80.10	**92.55**	90.23
Average	86.96 ± 7.18	90.57 ± 6.44	**92.2** ± 3.71	89.66 ± 5.92

Table 11: Reslience of all the models for all the datasets in ***Individual Topic + Partial Perturbation*** setup. Bold numbers in **Purple** color indicate the model with the best Resilience score. The last row shows the models' average resilience over all datasets including standard deviation.

and Model 2 predicts 70% of the examples correctly, then all the 60% of the examples for Model 1 and 70% of the examples for Model 2 need to be perturbed with an adversarial attack to maintain the consistency in measuring the difference in the performance of the models Model 1 and Model 2 against the corresponding adversarial attack. The models are pre-trained on NVIDIA 8GB GPUs.

4.4 Results

Mixed Topics Our proposed method outperforms the Model$_{\text{Baseline}}$, in terms of F1-score in 6 out of 7 original, non-perturbed datasets (see Table 8). All of our models achieve a higher or comparable average F1-score than the baseline. In addition, our models consistently outperform the baseline on the highly unbalanced FNC-1 dataset. When comparing our proposed models, Model$_{\text{Random}}$ achieved the best overall classification performance by learning from multiple randomly selected examples, while Model$_{\text{Random2}}$, which selects only two random triplets that may belong to different topics, performed worse. However, Model$_{\text{Random2}}$ still outperformed models **Hard** and **H&E**, which use only a few contrastive examples (one or two triplets) based on their label and similarity or dissimilarity to the anchor. This approach makes it less likely for them to cover a wider range of mixed topic examples.

In terms of resilience to perturbations, all of our models show a higher average resilience compared to the baseline (see Table 8). Model$_{\text{H\&E}}$ achieves a better average resilience score compared to all other models while maintaining a comparable average F1-score to the baseline. Indeed, the results suggest that using contrastive learning with only extreme or unorthodox "hard" examples, or a combination of both "hard" and standard "easy" examples, leads to more robust models when training examples belong to different topics (see Tables 8 and 10). On the other hand, although the baseline has a slightly better resilience score for the ARC dataset, all of our contrastive models perform better for highly unbalanced datasets like FNC-1, as well as for slightly less unbalanced datasets such as DebateForum and SemEval2016.

Mixed Topics + Partial Perturbation To validate previous results, we performed experiments where we only perturbed instances that were correctly classified by the models in the original test dataset. We observed similar results, with our proposed contrastive models exhibiting better resilience than the baseline overall (see Table 10). There was a significant increase of more than 10% for unbalanced datasets FNC-1 and DebateForum. Training with Model$_{\text{H\&E}}$ and Model$_{\text{Hard}}$ produced more robust models in general.

Individual Topics In this setting where the training data consists of examples from the same topic and dataset, our proposed models demonstrate comparable or superior F1-scores compared to the Model$_{\text{Baseline}}$ on average, and outperform it in eight out of nine non-perturbed test sets (refer to Table 9). Model$_{\text{H\&E}}$ and Model$_{\text{Hard}}$, achieved better performance compared to the Random models in the mixed topics settings. Specifically, the "hard" contrastive training strategy, which selects a dissimilar example with the same stance and a similar example with an opposite stance from the "same topic" in this case, appears to give the model a better ability not only to generalize but also to exhibit better stability, as evidenced by the resilience score of $\text{Model}_{\text{Hard}}$ (see Table 9). This is particularly evident when we only perturb correctly classified instances (see Table 11). For the smallest and most unbalanced topic dataset, $\text{Climate}_{\text{SE}}$, all our models outperform the baseline, with $\text{Model}_{\text{Hard}}$ achieving more than 20% increase in classification performance. Similarly, a notable increase in F1-score is observed with our models, specifically $\text{Model}_{\text{Hard}}$, for Marijuana_{DF}, Abortion_{SE}, and Atheism_{SE}. These datasets are highly imbalanced and relatively small, containing less than 750 examples.

Our proposed models exhibit better resilience scores than $\text{Model}_{\text{Baseline}}$ in 7 out of 9 datasets and also perform better in terms of resilience for the smaller and more imbalanced SE datasets, such as Abortion, Atheism, and Hillary Clinton. While the average resilience score of $\text{Model}_{\text{Random2}}$ and $\text{Model}_{\text{Hard}}$ is comparable, $\text{Model}_{\text{Hard}}$ achieves the best average F1-score among all the models on the original, non-perturbed test set.

Individual Topics + Partial Perturbation When perturbing only the correctly classified examples of a model, as in the previous setting, we observe a significant increase in the resilience score for our proposed models compared to the $\text{Model}_{\text{Baseline}}$ for the small and unbalanced topic datasets, namely Abortion, Atheism, Climate, and Hillary Clinton, as well as on average (see Table 11). Once again, $\text{Model}_{\text{Hard}}$ appears to be the most robust among the proposed models.

5 Conclusion

In this work, we have adopted the combination of CL + MLM method and explored different triplet strategies to learn more robust sentence representations to use in the Stance Detection task. Experiment results show that our proposed methodology is more resilient to errors and variations. Also, the experiments with different setups show that our proposed methodology is effective for small-sized as well as class-imbalanced datasets.

Limitations

We considered the binary stances examples topics mainly i.e. for/against, support/refute, or agree/disagree. The proposed methodology leverages the Contrastive Learning framework which is conditioned to work with two stance labels examples to identify whether the author of the text is in favor of or against the topic of discussion. However, social media such as Twitter and online forums like Reddit will have threads discussing topics having more than two stances such as for/against/neither, or support/refute/comment.

References

Abeer Aldayel and Walid Magdy. 2019. Your stance is exposed! analysing possible factors for stance detection on social media. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–20.

Shuyang Cao and Lu Wang. 2021. Cliff: Contrastive learning for improving faithfulness and factuality in abstractive summarization.

Sihao Chen, Daniel Khashabi, Wenpeng Yin, Chris Callison-Burch, and Dan Roth. 2019. Seeing things from a different angle: Discovering diverse perspectives about claims.

Kareem Darwish, Walid Magdy, and Tahar Zanouda. 2017. Improved stance prediction in a user similarity feature space.

Xinshuai Dong, Anh Tuan Luu, Min Lin, Shuicheng Yan, and Hanwang Zhang. 2021. How should pre-trained language models be fine-tuned towards adversarial robustness? In *Advances in Neural Information Processing Systems*, volume 34, page 4356–4369. Curran Associates, Inc.

Yangkai Du, Tengfei Ma, Lingfei Wu, Fangli Xu, Xuhong Zhang, Bo Long, and Shouling Ji. 2021. Constructing contrastive samples via summarization for text classification with limited annotations.

John Giorgi, Osvald Nitski, Bo Wang, and Gary Bader. 2020. Declutr: Deep contrastive learning for unsupervised textual representations.

Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2018. The argument reasoning comprehension task: Identification and reconstruction of implicit warrants. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1930–1940.

Kazi Saidul Hasan and Vincent Ng. 2013. Stance classification of ideological debates: Data, models, features, and constraints. In *Proceedings of the sixth international joint conference on natural language processing*, pages 1348–1356.

Kornraphop Kawintiranon and Lisa Singh. 2021. Knowledge enhanced masked language model for stance detection. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4725–4735, Online. Association for Computational Linguistics.

J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, Naval Technical Training Command Millington TN Research Branch.

Mirko Lai, Alessandra Teresa Cignarella, Delia Irazú Hernández Farías, Cristina Bosco, Viviana Patti, and Paolo Rosso. 2020. Multilingual stance detection in social media political debates. *Computer Speech & Language*, 63:101075.

Niels Landwehr, Mark Hall, and Eibe Frank. 2005. Logistic model trees. *Machine Learning*, 59(1):161–205.

Bin Liang, Zixiao Chen, Lin Gui, Yulan He, Min Yang, and Ruifeng Xu. 2022. Zero-shot stance detection via contrastive learning. In *Proceedings of the ACM Web Conference 2022*, pages 2738–2747.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Matthew Matero, Nikita Soni, Niranjan Balasubramanian, and H. Andrew Schwartz. 2021. Melt: Message-level transformer with masked document representations as pre-training for stance detection.

George A. Miller. 1994. WordNet: A lexical database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. Semeval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 31–41.

Milad Moradi and Matthias Samwald. 2021. Evaluating the robustness of neural language models to input perturbations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1558–1570, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Malte Ostendorff, Nils Rethmeier, Isabelle Augenstein, Bela Gipp, and Georg Rehm. 2022. Neighborhood contrastive learning for scientific document representations with citation embeddings.

Dean Pomerleau and Delip Rao. Exploring how artificial intelligence technologies could be leveraged to combat fake news.

Ammar Rashed, Mucahid Kutlu, Kareem Darwish, Tamer Elsayed, and Cansın Bayrak. 2020. Embeddings-based clustering for target specific stances: The case of a polarized turkey.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Benjamin Schiller, Johannes Daxenberger, and Iryna Gurevych. 2021. Stance detection benchmark: How robust is your stance detection? *KI - Künstliche Intelligenz*, 35(3–4):329–341.

Parinaz Sobhani, Diana Inkpen, and Xiaodan Zhu. 2017. A dataset for multi-target stance detection.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2019. Evaluating adversarial attacks against multiple fact verification systems. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2944–2953, Hong Kong, China. Association for Computational Linguistics.

Twitter. 2022. Twitter. it's what's happening.

Boxin Wang, Shuohang Wang, Yu Cheng, Zhe Gan, Ruoxi Jia, Bo Li, and Jingjing Liu. 2020. Infobert: Improving robustness of language models from an information theoretic perspective.

Hanlu Wu, Tengfei Ma, Lingfei Wu, Tariro Manyumwa, and Shouling Ji. 2020a. Unsupervised reference-free summary quality evaluation via contrastive learning.

Zhuofeng Wu, Sinong Wang, Jiatao Gu, Madian Khabsa, Fei Sun, and Hao Ma. 2020b. Clear: Contrastive learning for sentence representation. *arXiv preprint arXiv:2012.15466*.

Hu Xu, Gargi Ghosh, Po-Yao Huang, Dmytro Okhonko, Armen Aghajanyan, Florian Metze, Luke Zettlemoyer, and Christoph Feichtenhofer. 2021. Videoclip: Contrastive pre-training for zero-shot video-text understanding.

Zhenrui Yue, Bernhard Kratzwald, and Stefan Feuerriegel. 2021. Contrastive domain adaptation for question answering using limited text corpora.

Cenyuan Zhang, Xiang Zhou, Yixin Wan, Xiaoqing Zheng, Kai-Wei Chang, and Cho-Jui Hsieh. 2022. Improving the adversarial robustness of NLP models by information bottleneck. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3588–3598, Dublin, Ireland. Association for Computational Linguistics.

Zhiwei Zhang, Jiyi Li, Fumiyo Fukumoto, and Yanming Ye. 2021. Abstract, rationale, stance: A joint model for scientific claim verification.

Adapting Emotion Detection to Analyze Influence Campaigns on Social Media

Ankita Bhaumik, Andy Bernhardt, Gregorios A Katsios,
Ning Sa, Tomek Strzalkowski
Rensselaer Polytechnic Institute, Troy, New York

Abstract

Social media is an extremely potent tool for influencing public opinion, particularly during important events such as elections, pandemics, and national conflicts. Emotions are a crucial aspect of this influence, but detecting them accurately in the political domain is a significant challenge due to the lack of suitable emotion labels and training datasets. In this paper, we present a generalized approach to emotion detection that can be adapted to the political domain with minimal performance sacrifice. Our approach is designed to be easily integrated into existing models without the need for additional training or fine-tuning. We demonstrate the zero-shot and few-shot performance of our model on the 2017 French presidential elections and propose efficient emotion groupings that would aid in effectively analyzing influence campaigns and agendas on social media.

1 Introduction

Digital environments, such as social media, are powerful launching platforms for wide-reaching influence campaigns surrounding important events such as elections, pandemics, and armed conflicts, as well as commercial interests (Karlsen and Enjolras, 2016; Raudeliūnienė et al., 2018; Badawy et al., 2019). These campaigns aim to manipulate public opinion in a particular way: to favor or oppose a political candidate, to accept or resist vaccination, to justify an aggression, etc. This is achieved by disseminating messages that advance a specific agenda, using language, imagery, and topics that are likely to resonate with the target audience.

Presidential elections offer a substantial context for examining influence campaigns on social media platforms and is the focus of this study. Various indicators, such as agenda, stance, concern, belief, emotion, and imageability, have been identified for measuring the influence of social media messages within this context (Mather et al., 2022).

Emotion is deeply integrated in political discourse and is used as a rhetorical tool in persuading the audience (Cislaru, 2012). Emotionally charged messages can significantly sway public opinion regarding specific agendas or candidates (Weber, 2013; Mohammad et al., 2015) and several studies have documented the effect of emotional language in disseminating polarizing content via social media platforms (Brady et al., 2017).

Existing social media datasets, especially those focused on election-related messages posted on Twitter, are labeled using traditional emotion categories derived from Ekman or Plutchik labels (Ekman, 1999; Plutchik, 1984). These datasets facilitate the development of emotion analysis tools and apply them on diverse applications ranging from healthcare (Tivatansakul et al., 2014) and education (Karan et al., 2022) to stock market (Aslam et al., 2022) and political opinion mining (Cabot et al., 2020). However, each new application domain presents its own set of challenges that existing systems are unable to handle. Therefore, when a new emotion detection problem emerges in a specialized domain, researchers engage in an exhaustive annotation process to build relevant datasets. This highlights the necessity for enhancing the flexibility and robustness of existing models in order to accommodate new scenarios.

Potential solutions involve using semi-supervised, unsupervised, zero-shot, or few-shot techniques (Yin et al., 2019; Chen et al., 2022; Zhang et al., 2019). Nevertheless, solely relying on emotion labels and their definitions from external resources, such as WordNet (Strapparava et al., 2004), are insufficient to capture the intricate concepts and subtleties associated with each emotion label when viewed through the lens of the application domain. Psychological theories suggest that emotion definitions are not universally applicable across domains or individuals, rather, they are profoundly shaped by the socio-cultural

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 441–451
July 14, 2023 ©2023 Association for Computational Linguistics

context and specific events (Averill, 1980; Mohammad and Kiritchenko, 2018), emphasizing the need to incorporate domain-specific knowledge and emotion inter dependencies for effective zero-shot systems.

But which emotions matter in an influence campaign? Do the same emotions arise when discussing a new electronic gadget on the market as when comparing political candidates ahead of an election? In this paper, we present a novel zero-shot approach to detect emotions in text, adaptable to unexplored domains or target label sets. Our method incorporates interpretations of emotion labels and their inter dependencies for improved results in the target domain. We investigate tweets around the 2017 French Presidential Elections part of which is publicly available on Kaggle (Daignan, 2017) and thoroughly evaluate our method to demonstrate that it addresses the shortfalls of existing zero-shot approaches. This is an important step towards providing valuable insights on the emotions of the audience towards political campaigns and agendas.

2 Background

2.1 Emotions in Political Discourse

Extensive research has been conducted on the strategic employment of emotions to sway voting behaviors and public opinion during political campaigns. Campaigns often utilize specific emotional appeals, such as positive emotions (e.g., enthusiasm and pride) to foster support, while leveraging negative emotions (e.g., fear and anger) to incite negative emotions towards the opposition (Ridout and Searles, 2011; Fridkin and Kenney, 2012; Grüning and Schubert, 2022). Some studies contend that only certain emotions, namely anxiety and enthusiasm, are particularly influential in political contexts (Marcus and MacKuen, 1993), with anger and other negative emotions frequently employed by political leaders (Cislaru, 2012).

Prior studies have also reported that negative campaign emotions, such as anger, contempt, disgust, and fear often co-occur and are difficult to distinguish (Fridkin and Kenney, 2012; Mohammad and Kiritchenko, 2018). Consequently, the selection of emotion labels is heavily reliant on the specific influence patterns under examination, which presents the challenge of developing a versatile emotion model capable of adapting to various emotion label sets.

2.2 Emotion Detection Models

Emotion detection in text is a long-standing research challenge due to the ever-changing nature of textual content across applications and platforms. Large pretrained language models, such as GPT (Radford et al., 2018) and BERT (Devlin et al., 2019), have emerged as powerful tools for this task (Cai and Hao, 2018; Huang et al., 2019; Polignano et al., 2019; Ma et al., 2019; Chiorrini et al., 2021). Our approach employs popular Twitter-specific language models, which provide a robust baseline for core NLP tasks in social media analysis (Barbieri et al., 2020).

Zero-shot learning techniques are frequently employed for emotion detection when training data is unavailable in the target domain. Recent studies in zero-shot emotion detection use text entailment approaches, wherein target labels generate hypotheses for the model (Yin et al., 2019; Basile et al., 2021). Prompt engineering techniques also facilitate emotion label inference from pretrained NLI models (Plaza-del Arco et al., 2022). Additionally, some zero-shot methods leverage sentence embeddings for unsupervised or semi-supervised predictions on unlabeled datasets (Chen et al., 2022; Zhang et al., 2019; Olah et al., 2021). The drawback of these techniques stem from their generalized design, enabling them to function across multiple domains, while only excel when target emotion labels align with standard definitions. They lack integration of domain knowledge or comprehension of emotion concepts that may arise in specialized domains.

3 Methodology

Upon completing a preliminary investigation of the 2017 French election dataset, our annotation team observed that assigning a distinct emotion label to each tweet is a challenging and a complex task. Following a more practical approach, we label tweets using groups of emotions that frequently co-occur or overlap (Mohammad and Kiritchenko, 2018; Cislaru, 2012). These groups of emotions are combinations of the traditional emotion labels and are difficult to isolate from short informal tweets. In instances where a message cannot be classified into any of the emotion groups but still conveys a strong positive or negative sentiment, it is assigned a "Positive-other" or "Negative-other" label. The following is the final set E of grouped emotion labels:

1. Anger, hate, contempt, disgust
2. Embarrassment, guilt, shame, sadness
3. Admiration, love
4. Optimism, hope
5. Joy, happiness
6. Pride, including national pride
7. Fear, pessimism
8. Sarcasm, amusement
9. Positive-other
10. Negative-other

3.1 Problem Statement

The goal is to automatically tag a text message x with scores between 0 and 1 for each emotion label in E. The score for each label $e \in E$ should reflect the confidence that the emotion e is expressed by the author of x.

3.2 Approach

Our approach combines traditional sentiment analysis and emotion detection results, obtained by hierarchical grouping of standard emotions. The grouped emotion labels form the basis for our ensemble model, which can be readily adapted to the political domain without the need for additional training. The performance of this ensemble can be further optimized with the availability of some in-domain annotated data. We illustrate our emotion model ensemble in Fig.1.

Given the text of a tweet as input, our model produces scores over three sentiment categories, six Ekman emotions, and their fine-grained subcategories defined in (Demszky et al., 2020). To obtain confidence scores over label set E, we design a many-to-one mapping based on the emotion groups and their corresponding definitions in the political domain.

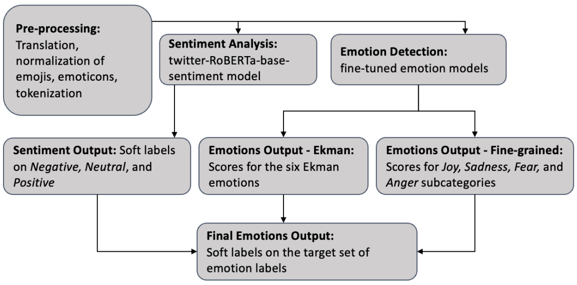

Pre-processing: Translation, normalization of emojis, emoticons, tokenization
Sentiment Analysis: twitter-RoBERTa-base-sentiment model
Emotion Detection: fine-tuned emotion models
Sentiment Output: Soft labels on *Negative, Neutral*, and *Positive*
Emotions Output - Ekman: Scores for the six Ekman emotions
Emotions Output - Fine-grained: Scores for *Joy, Sadness, Fear,* and *Anger* subcategories
Final Emotions Output: Soft labels on the target set of emotion labels

Figure 1: Ensemble Emotion Detection Architecture

3.3 Datasets & Preprocessing

We have identified two social media datasets that can be utilized to train the emotion models in our ensemble, providing us with the broadest possible coverage of all standard and fine-grained emotion labels:

Cleaned Balanced Emotional Tweets (CBET) (Shahraki and Zaiane, 2017) is a collection of 81k English tweets that have been collected using a set of hashtags corresponding to the nine emotion labels (*anger, fear, joy, love, sadness, surprise, thankfulness, disgust*, and *guilt*). We use this dataset to train a model to predict scores over the six Ekman emotions, removing the annotations for *thankfulness, love* and *guilt*. The 56,281 remaining tweets that have at least one emotion label are split randomly into training (81%), validation (9%), and testing (10%) sets.

GoEmotions (Demszky et al., 2020) is a corpus of 58k English Reddit comments manually annotated with 27 emotion labels or *neutral*. The large number of fine-grained emotion labels in this dataset makes it an ideal choice for creating a base emotion model suitable specialized emotion tasks. We use GoEmotions to train a model to predict scores over the six Ekman emotions, and for the emotions of *joy, sadness, fear* and *anger*, we identify their lower level emotions in the hierarchy of the dataset to produce the training, validation, and testing sets (Table 1) to train specialized emotion models.

Model	Training	Validation	Test
joy	17,410	2,219	2,104
sadness	3,263	390	379
fear	726	105	98
anger	5,579	717	726

Table 1: Distribution of training, validation, and test sets for emotion subcategory models derived from GoEmotions

Given an input tweet, our system first translates it to English [1] and applies basic text preprocessing techniques (Tiedemann and Thottingal, 2020). The preprocessing pipeline is used as a social tokenizer (Baziotis et al., 2017) to remove any usernames, tweet IDs, hyperlinks, emails, phone numbers, times, dates, and percentages, normalize money

---

[1] `https://huggingface.co/Helsinki-NLP/opus-mt-fr-en`

values and numbers, annotate any censored or elongated words, and convert emoticons to plain text.

3.4 Training and Fine-tuning

For the task of sentiment analysis, we use the twitter-XLM-RoBERTa-base-sentiment [2] model to produce normalized values on the three sentiment categories *negative, neutral*, and *positive* (Barbieri et al., 2020).

For emotion detection, we further fine-tune six models as components of the hierarchical mapping system. Each model in the ensemble is built using the twitter-RoBERTa-base-emotion [3] (Barbieri et al., 2020) checkpoint, but we append a new linear layer on top of the last hidden state of RoBERTa's [CLS] token. The purpose of the linear layer is to convert the final hidden state vector into a vector related to the distinct emotion labels in the corresponding dataset. Subsequently, this vector can be converted into probabilities via the Softmax function. The labels of each model are listed in Table 2.

In the first step, two models are fine-tuned to output normalized scores on the six Ekman emotions using the CBET Twitter data and GoEmotions Reddit data. We choose to train separate models on Twitter and Reddit data to be able to weigh them in the next step based on the target domain. The remaining four models are then fine-tuned to output scores on the subcategories of *joy, sadness, fear, and anger*. The fine-tuning setup and metrics for each model are described in Appendix A.

3.5 Hierarchical Label Transfer

The fine-grained emotion scores are used downstream to adapt the model to a new domain. We map the scores from the model outputs to scores over a desired label set E using a weighted linear combination derived by considering the relatedness of emotions in the Plutchik's wheel of emotions (Plutchik, 1984) and the co-occurrence of emotions in the target domain. A general set of rules to determine the mapping from the hierarchical emotion model outputs to other emotions $e \in E$ is outlined below:

1. Determine which sentiment category (positive/negative) $S \subseteq Sent$ corresponds to emotion e. (e.g. *e: optimisms* $\rightarrow$ *s: positive*). For ambiguous emotions, we choose the sentiment category with a higher score.

Model	**Output Labels**
Sentiment(Sent)	positive, neutral, negative
CBET-Ekman	joy, sadness, fear, anger, disgust, surprise
GE-Ekman	joy, sadness, fear, anger, disgust, surprise
Joy(J)	joy, amusement, approval, excitement, gratitude, love, optimism, relief, pride, admiration, desire, caring
Sadness(S)	sadness, disappointment, embarrassment, grief, remorse
Fear(F)	fear, nervousness
Anger(A)	anger, annoyance, disapproval

Table 2: Set of output labels for each component model.

2. To remove any bias caused by a specific dataset, calculate one output score EK for each Ekman label using a linear combination of the scores from the CBET-Ekman and GE-Ekman models.

3. For each sentiment $s \in S$, determine which high-level Ekman emotions corresponding to s, $EK_s \subseteq EK$ have subcategories relevant to emotion e. As mentioned above, the sentiment of *e: optimism* is *positive*, and *joy* is the EK_s which corresponds to *s: positive* and has subcategories relevant to *e: optimism*.

4. For each high-level Ekman emotion $ek \in EK_s$, if ek has subcategories, determine which subcategories $sub_{ek} \subseteq Sub_{ek}$ are relevant to emotion e. Continuing with the example of *optimism*, out of all the *Joy* subcategories, the only relevant subcategory is *optimism*.

5. Then, the score of e is

$$\sum_{s\in S}\sum_{ek\in EK_s}\sum_{sub_{ek}\in Sub_{ek}} \alpha\left(Sent[s]*EK[ek]*Sub_{ek}[sub_{ek}]\right)$$

where α is a weight that can be set to 1, or fine-tuned to maximize a performance metric on a target-domain validation set (if one exists). In other words, the final score for e is a weighted sum of terms, where each term

---

[2] `https://huggingface.co/cardiffnlp/twitter-xlm-roberta-base-sentiment`

[3] `https://huggingface.co/cardiffnlp/twitter-roberta-base-emotion`

Mapping	Output Label
((EK[anger] * A[anger]) + EK[disgust]) * Sent[negative]	anger, contempt, disgust
(EK[sadness] * (S[sadness] + S[embarrassment] + Sent[grief])) * Sent[negative]	embarrassment, guilt
(EK[joy] * (J[admiration] + J[love])) * Sent[positive]	admiration, love
(EK[joy] * (J[optimism])) * Sent[positive]	optimism, hope
(EK[joy] * (J[joy])) * Sent[positive]	joy, happiness
(EK[joy] * (J[pride])) * Sent[positive]	pride
(EK[fear] * (F[fear])) * Sent[negative]	fear, pessimism
(EK[joy] * (J[amusement])) * Sent[positive]	amusement, sarcasm
(EK[joy] * (J[approval] + J[excitement] + J[gratitude] + J[relief] + J[desire] + J[caring])) * Sent[positive]	positive-other
((EK[sadness] * (S[disappointment] + S[remorse])) + (EK[fear] * (F[nervousness])) + (EK[anger] * (A[annoyance] + A[disapproval]))) * Sent[negative]	negative-other

Table 3: Mapping of model outputs to French election labels

is the product of scores for a sentiment, Ekman emotion, and low-level emotion subcategory triple that is relevant to e. For example, for the output emotion *optimism*, we may have the term $(Sent[positive] * EK[joy] * Joy[optimism])$.

3.6 In-domain Optimization

The availability of any in-domain data can be used as a validation set to boost the model performance in two ways: 1) finding a set of optimal classification thresholds for each emotion label, and 2) fine-tuning the weights of the linear mapping of the emotion scores for a target metric. The classification thresholds are fine-tuned by choosing a threshold for each target class to maximize the F1 score on that class over the validation dataset.

The mapping weights are optimized by successively applying differential evolution to each individual target label mapping to maximize the F1 score on that label over the validation dataset (Storn and Price, 1997). We use a subset of the manually annotated French election dataset to fine-tune both the mapping weights and the classification thresholds by first optimizing the weights, and subsequently choosing the thresholds for each label. More details on the label-wise classification thresholds and mapping weights parameters have been listed in Appendix B.

3.7 Data

Our annotation team utilized the emotion label set E, as detailed in Section 3, to annotate a subset of the 2017 French Presidential Election dataset. Three raters independently assigned one or more emotions to each tweet, with a label considered ground truth if confirmed by at least two annotators. The inter-rater reliability (IRR) across all emotion labels for the three raters was determined to be 17%, calculated by macro-averaging kappa scores (Carletta, 1996) between each rater pair. This low IRR highlights the task's complexity and the challenge of obtaining consistent emotion labels in this domain. Factors such as political background familiarity, cross-cultural dynamics, and multilingualism contribute to this complexity (Shaikh et al., 2015).

In addition, the annotators assigned agenda labels as a second influence indicator to the dataset. An agenda can be defined as the indicator that influences the target audience to believe in something or to perform a certain task (e.g., vote for a candidate, engage in a demonstration). We perform a set of experiments that utilize these agenda labels to examine the emotional patterns in relation to different agendas in a campaign. We show that the use of emotional language tends to align strongly with the type of agenda being promoted.

4 Experiments

In this section, we compare our approach against popular semi-supervised and zero-shot techniques. All experiments have been carried out on the French election dataset in the below configurations:

- *Zero-shot mode:* Emotion classification on the test set by adapting the model ensemble to the target domain without any fine-tuning. We also repeat this experiment without the sentiment component in the ensemble to demonstrate its contribution.

- *In-domain optimization mode:* Use a small subset of available in-domain data to optimize the classification thresholds and mapping weights.

4.1 Baselines

We evaluate our approach against the following baselines:

- *Zero-shot textual entailment (TE):* Following the work of Yin et al., 2019, we convert each emotion label into the hypothesis: "This text expresses <label>." We use the BART MNLI [4] model to generate entailment and contradiction scores and threshold them to produce binary outputs for each label.
- *Zero-shot sentence embeddings (SB):* We use SBERT (Reimers and Gurevych, 2019) to obtain the embeddings for the input texts and class labels [5]. The texts are then labeled based on their closeness to the labels in the embedding space using cosine similarity.
- *Semi-supervised models (EK):* We use existing emotion datasets (CBET and GoEmotions) to fine-tune twitter-RoBERTa-base-emotion pretrained models (Barbieri et al., 2020) on the six Ekman labels, and test these models over the label set in the target domain. Many of the target labels are absent in these Ekman datasets and thus their outputs are set to 0.

4.2 Results

The mapping of the model ensemble outputs to the French election emotion groupings (Table 3) follows the general rules outlined in Section 3.4. For example, each label in *anger, hate, contempt, disgust* is associated with a *negative* sentiment. Further, for the Ekman emotions *anger* and *disgust*, the only relevant subcategory is *anger*, which results in the final mapping *((EK[anger] * Anger[anger]) + EK[disgust]) * Sentiment[negative]*. The label *positive-other* is associated with a *positive* sentiment and the only positive Ekman emotion, *joy*. Additionally, from the label definition, it accumulates scores of all the positive fine-grained emotions that have not been recorded by any other label. Figure 2 shows an example tweet from the test dataset with its output emotion scores.

The evaluation metrics in Table 4 highlight the poor performance of existing zero-shot methods on the French Election dataset. This is because these models do not incorporate any domain knowledge and rely solely on the standard emotion definitions to classify text. The specialized label groups in the French election labels are tailored to the influence detection task, which makes them difficult for traditional emotion detection models to handle. For example, the labels *love* and *admiration* can be synonymous in a political influence campaign but not in a general emotion taxonomy. This further emphasizes the need for domain-specific knowledge in emotion detection models which is incorporated by our label transfer step.

	EK	TE	SB	Ours
anger/cont/disgust	0.17	0.13	0.13	0.23
embarrass/guilt	0.05	0.03	0.04	0.19
admiration/love	0	0.04	0.04	0.15
optimism/hope	0	0.22	0.16	0.30
joy/happiness	0.04	0.04	0.03	0.16
pride	0	0.07	0.07	0.17
fear/pessimism	0.10	0.07	0.06	0.18
amusement	0	0.14	0.14	0.14
positive-other	0	0.56	0.43	0.50
negative-other	0	0.53	0.41	0.50

Table 4: F1 scores across all emotion labels in the French Election dataset; (Ours: zero-shot performance of deploying our emotion model ensemble over this dataset)

For the few-shot mode (Table 5), the optimization of the classification thresholds and label mapping weights do not aid in improving the performance of the model due to inconsistencies in annotation between the validation and test datasets. We believe that more consistent annotations or sampling fine-tuning data from the same dataset would result in a performance boost as observed in other datasets performing the same task.

We also perform an ablation study to understand the effect of adding a sentiment component to the ensemble (Table 5). The improvement of scores across all experiments ascertain that the influence of sentiment is crucial for this emotion detection task.

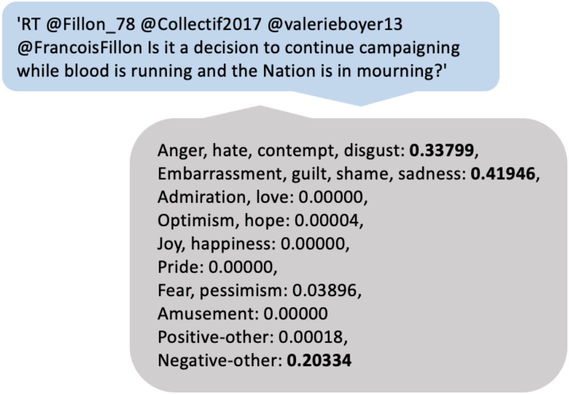
---

[4] `https://huggingface.co/facebook/bart-large-mnli`

[5] We use the SBERT all-MiniLM-L6-v2 pretrained model to obtain the embeddings.

	P	R	F1
Semi-supervised			
CBET	0.05	0.07	0.06
GoEmotions	0.05	0.08	0.06
CBET + GoEmotions	0.06	0.09	0.07
Zero-shot			
BART MNLI (TE)	0.13	0.86	0.23
SBERT (SB)	0.10	0.65	0.17
Ours	0.32	0.44	0.37
Ours + Sentiment	0.34	0.48	**0.40**
Few-shot			
Optimize mapping	0.34	0.48	**0.39**
Optimize threshold	0.29	0.29	0.29

Table 5: Evaluation results against baselines. Ours: Our emotion model ensemble without the sentiment module; Ours + Sentiment: Our emotion model ensemble with sentiment module. *Few-shot* section lists results of optimizing our label transfer step with the availability of some in-domain data.



'RT @Fillon_78 @Collectif2017 @valerieboyer13 @FrancoisFillon Is it a decision to continue campaigning while blood is running and the Nation is in mourning?'

Anger, hate, contempt, disgust: **0.33799**,
Embarrassment, guilt, shame, sadness: **0.41946**,
Admiration, love: 0.00000,
Optimism, hope: 0.00004,
Joy, happiness: 0.00000,
Pride: 0.00000,
Fear, pessimism: 0.03896,
Amusement: 0.00000
Positive-other: 0.00018,
Negative-other: **0.20334**

Figure 2: Example tweet from the French election dataset.

5 Emotions as an Influence Indicator

We use our emotion detection approach to understand how emotions correlate with other influence indicators during political campaigns. We select a subset of the tweets that are associated with specific agendas in the election. Figure 5a shows the emotion distribution across tweets mentioning popular candidates. As expected, the predominant emotions are *anger, contempt, disgust* and *optimism, hope* signifying that political campaigns either influence the audience by expressing hope/optimism for a brighter future or by expressing hatred towards the opposing candidate or political party. Interestingly, although a large portion of the tweets express some strong positive/negative emotion, they cannot be accurately tagged with a specific emotion label (Fig 3). This leads us to hypothesize that a large number of emotion labels may not be required to effectively analyze the emotional influence of political campaigns.

Figures 4 and 5b illustrates the results of emotion detection on the agenda annotated tweets. In this paper, we focus on the following agendas: 1) believe that an entity (E) or group (G) is immoral or harmful; 2) believe that E/G is moral or beneficial; 3) believe that your group are at risk; 4) believe that your actions can lead to a good outcome or hope; 5) call to share information; 6) call to vote for a E/G; 7) call to vote against a E/G; and 8) call to participate in demonstration/protest or attend a rally/campaign.

In Figure 4, higher anger and negative-other scores are observed in the agenda of 'Entity is immoral'. In contrast, the 'Entity is moral' agenda shows higher *admiration* and *positive-other* scores. By comparing the emotions of 'group at risk' and 'belief for good outcome', we find higher *anger, fear*, and *negative-other* scores in the former, and higher *optimism* and *positive-other* scores in the latter. Similarly, the agendas 'vote for entity' and 'vote against entity' have higher positive and negative emotions, respectively. These differences can be clearly seen in Fig. 5b that shows the proportion of each emotion in an agenda. We can conclude that emotions play an important role in understanding patterns in a campaign and the impact of political agendas on the audience.

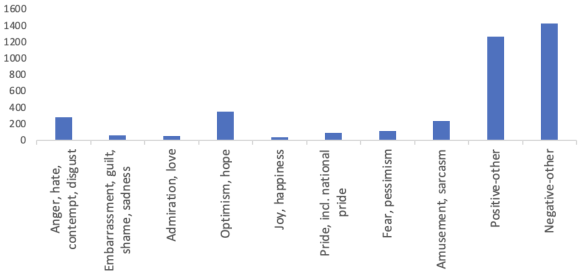


Figure 3: Distribution of emotion labels in the test dataset

6 Conclusion

Our paper presents a generalized approach to emotion detection wherein existing emotion detection datasets and models can be quickly adapted to specialized emotion labels to effectively analyze influence campaigns in the political domain. Our experiments demonstrate the efficacy of this zero-shot approach on tweets from the 2017 French presidential election.

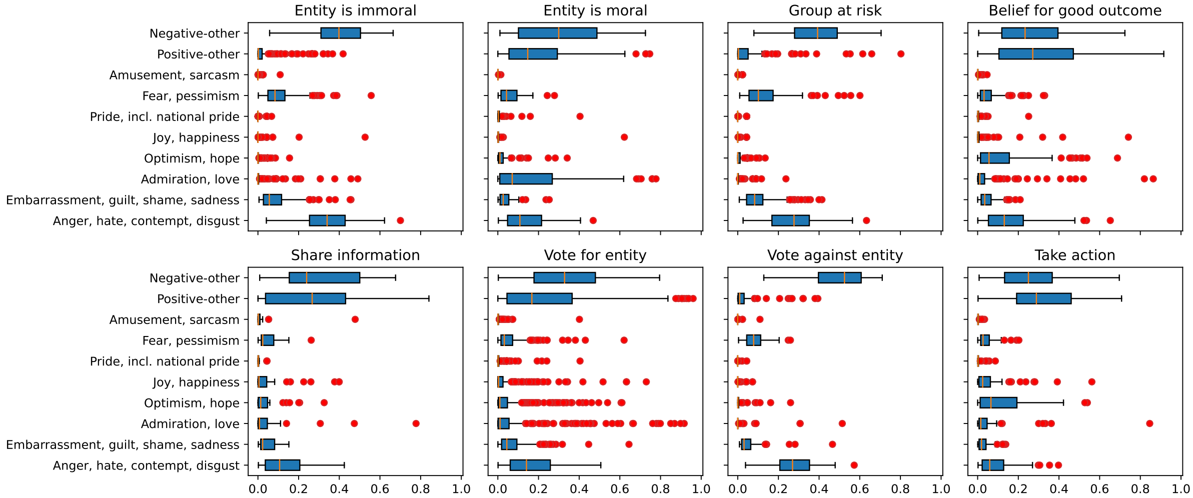
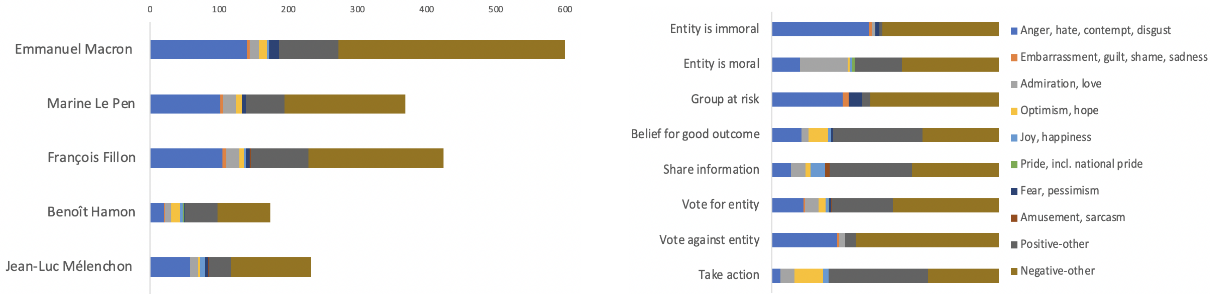


Figure 4: Boxplots showing summary of emotion scores across different agendas in the campaign. The box from the first quartile to the third quartile, the line depicting the median score for that label. The whiskers are shown from the box by 1.5x the inter-quartile range. Anything past the whiskers are shown as outliers in red.



(a) Distribution of emotion labels across tweets mentioning popular candidates

(b) Distribution of emotion labels across different agendas in the campaign

Figure 5: Distribution of emotions during campaigns for political figures or agendas

We further utilize our inference results to get insights on the use of emotional language along with other influence indicators like agenda. This work could be utilized in multiple downstream applications to forecast election outcomes or understand public opinions on specific agendas or issues. Our results signify the importance of certain emotion groups in political campaigns and provides a path for future work integrating multiple influence indicators in social media and understanding interdependencies between different emotions.

7 Limitations

Currently our approach relies on translation to analyze multilingual tweets. Future work would include using multilingual pre-trained models like XLM-RoBERTa and the use of non-English training data to build a language agnostic emotion model ensemble.

We carry out our in-domain optimization on a small validation dataset that was annotated by a different set of raters than the one used for the test dataset, which results in a performance drop in the few-shot mode. Ideally, the availability of a high quality validation dataset would boost the zero-shot performance and further adapt the label mappings to the target domain. We also aim to carry out in house annotations by experts to release a publicly available dataset annotated with emotions in the political domain which would pave the way for further analysis in this domain.

Acknowledgements

This paper is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR001121C0186. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of DARPA or the U.S. Government.

Ethics Statement

We use multiple Twitter and Reddit datasets to fine-tune our emotion model ensemble. Both these datasets have been cleaned to remove any toxicity, biases and offensive language. The annotated French election dataset cannot be publicly released following the terms and conditions of the project. The data available to us for fine-tuning and evaluation does not contain any personally identifiable data and we do not have any knowledge of the annotators behind creating this dataset. We also utilize multiple pre-trained models which reduces the carbon footprint of training models from scratch. Further, utilization of this transfer learning method for any new domain would not incur any training costs as minimal fine-tuning may be required. However, the results obtained in an unknown domain should be human evaluated before using it for any downstream analytics task.

References

Naila Aslam, Furqan Rustam, Ernesto Lee, Patrick Bernard Washington, and Imran Ashraf. 2022. Sentiment analysis and emotion detection on cryptocurrency related Tweets using ensemble LSTM-GRU Model. *IEEE Access*, 10:39313–39324.

James R Averill. 1980. A constructivist view of emotion. In *Theories of emotion*, pages 305–339. Elsevier.

Adam Badawy, Aseel Addawood, Kristina Lerman, and Emilio Ferrara. 2019. Characterizing the 2016 russian ira influence campaign. *Social Network Analysis and Mining*, 9:1–11.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. TweetEval: Unified Benchmark and Comparative Evaluation for Tweet Classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650.

Angelo Basile, Guillermo Pérez-Torró, and Marc Franco-Salvador. 2021. Probabilistic ensembles of zero-and few-shot learning models for emotion classification. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, pages 128–137.

Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. 2017. DataStories at SemEval-2017 task 4: Deep LSTM with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 747–754.

William J Brady, Julian A Wills, John T Jost, Joshua A Tucker, and Jay J Van Bavel. 2017. Emotion shapes the diffusion of moralized content in social networks. *Proceedings of the National Academy of Sciences*, 114(28):7313–7318.

Pere-Lluís Huguet Cabot, Verna Dankers, David Abadi, Agneta Fischer, and Ekaterina Shutova. 2020. The pragmatics behind politics: Modelling metaphor, framing and emotion in political discourse. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4479–4488.

Xiaofeng Cai and Zhifeng Hao. 2018. Multi-view and attention-based bi-LSTM for Weibo emotion recognition. In *2018 International Conference on Network, Communication, Computer Engineering*, pages 772–779.

Jean Carletta. 1996. Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 22(2):249–254.

Qi Chen, Wei Wang, Kaizhu Huang, and Frans Coenen. 2022. Zero-shot text classification via knowledge graph embedding for social media data. *IEEE Internet of Things Journal*, 9(12):9205–9213.

Andrea Chiorrini, Claudia Diamantini, Alex Mircoli, and Domenico Potena. 2021. Emotion and sentiment analysis of tweets using BERT. In *EDBT/ICDT Workshops*.

Georgeta Cislaru. 2012. Emotions as a rhetorical tool in political discourse. In Maria Zaleska, editor, *Rhetoric and Politics*, pages 107–126. Cambrige Scholar Press.

Jean-Michel Daignan. 2017. French presidential election: Extract from twitter about the french election.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A dataset of fine-grained emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4040–4054.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Paul Ekman. 1999. Basic emotions. *Handbook of cognition and emotion*, 98(45-60):16.

Kim L Fridkin and Patrick J Kenney. 2012. The impact of negative campaigning on citizens' actions and attitudes. *The SAGE handbook of political communication*, pages 173–185.

David J Grüning and Thomas W Schubert. 2022. Emotional campaigning in politics: Being moved and anger in political ads motivate to support candidate and party. *Frontiers in Psychology*, 12:6337.

Yen-Hao Huang, Ssu-Rui Lee, Mau-Yun Ma, Yi-Hsin Chen, Ya-Wen Yu, and Yi-Shin Chen. 2019. EmotionX-IDEA: Emotion BERT- an Affectional Model for Conversation. *arXiv preprint arXiv:1908.06264*.

KV Karan, Vedant Bahel, R Ranjana, and T Subha. 2022. Transfer learning approach for analyzing attentiveness of students in an online classroom environment with emotion detection. In *Innovations in Computational Intelligence and Computer Vision: Proceedings of ICICV 2021*, pages 253–261. Springer.

Rune Karlsen and Bernard Enjolras. 2016. Styles of social media campaigning and influence in a hybrid political communication system: Linking candidate survey data with twitter data. *The International Journal of Press/Politics*, 21(3):338–357.

Luyao Ma, Long Zhang, Wei Ye, and Wenhui Hu. 2019. PKUSE at SemEval-2019 task 3: emotion detection with emotion-oriented neural attention network. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 287–291.

George E Marcus and Michael B MacKuen. 1993. Anxiety, enthusiasm, and the vote: The emotional underpinnings of learning and involvement during presidential campaigns. *American Political Science Review*, 87(3):672–685.

Brodie Mather, Bonnie Dorr, Adam Dalton, William de Beaumont, Owen Rambow, and Sonja Schmer-Galunder. 2022. From stance to concern: Adaptation of propositional analysis to new tasks and domains. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3354–3367.

Saif Mohammad and Svetlana Kiritchenko. 2018. Understanding emotions: A dataset of tweets to study interactions between affect categories. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Saif M Mohammad, Xiaodan Zhu, Svetlana Kiritchenko, and Joel Martin. 2015. Sentiment, emotion, purpose, and style in electoral tweets. *Information Processing & Management*, 51(4):480–499.

Justin Olah, Sabyasachee Baruah, Digbalay Bose, and Shrikanth Narayanan. 2021. Cross domain emotion recognition using few shot knowledge transfer. *arXiv preprint arXiv:2110.05021*.

Flor Miriam Plaza-del Arco, María-Teresa Martín-Valdivia, and Roman Klinger. 2022. Natural language inference prompts for zero-shot emotion classification in text across corpora. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 6805–6817.

Robert Plutchik. 1984. Emotions: A general psychoevolutionary theory. *Approaches to emotion*, 1984(197-219):2–4.

Marco Polignano, Pierpaolo Basile, Marco de Gemmis, and Giovanni Semeraro. 2019. A comparison of word-embeddings in emotion detection from text using biLSTM, CNN and Self-attention. In *Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization*, pages 63–68.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning.

Jurgita Raudeliūnienė, Vida Davidavičienė, Manuela Tvaronavičienė, and Laimonas Jonuška. 2018. Evaluation of advertising campaigns on social media networks. *Sustainability*, 10(4):973.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Travis N Ridout and Kathleen Searles. 2011. It's my campaign i'll cry if i want to: How and when campaigns use emotional appeals. *Political Psychology*, 32(3):439–458.

Ameneh Gholipour Shahraki and Osmar R Zaiane. 2017. Lexical and learning-based emotion mining from text. In *Proceedings of the international conference on computational linguistics and intelligent text processing*, volume 9, pages 24–55.

Samira Shaikh, Tomek Strzalkowski, Sarah Taylor, John Lien, Ting Liu, George Aaron Broadwell, Laurie Feldman, Boris Yamrom, Kit Cho, and Yuliya Peshkova. 2015. Understanding cultural conflicts using metaphors and sociolinguistic measures of influence. In *Proceedings of the Third Workshop on Metaphor in NLP*, pages 67–76.

Rainer Storn and Kenneth Price. 1997. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11:341–359.

Carlo Strapparava, Alessandro Valitutti, et al. 2004. WordNet Affect: An affective extension of wordnet. In *Lrec*, volume 4, page 40. Lisbon, Portugal.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT — Building open translation services for the World. In *Proceedings of the 22nd Annual Conferenec of the European Association for Machine Translation (EAMT)*, Lisbon, Portugal.

Somchanok Tivatansakul, Michiko Ohkura, Supadchaya Puangpontip, and Tiranee Achalakul. 2014. Emotional healthcare system: Emotion detection by facial expressions using japanese database. In *2014 6th computer science and electronic engineering conference (CEEC)*, pages 41–46. IEEE.

Christopher Weber. 2013. Emotions, campaigns, and political participation. *Political Research Quarterly*, 66(2):414–428.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 3914–3923.

Jingqing Zhang, Piyawat Lertvittayakumjorn, and Yike Guo. 2019. Integrating semantic knowledge to tackle zero-shot text classification. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1031–1040.

A Hyperparameters

To fine-tune the pretrained twitter-RoBERTa-base-emotion models on each of the six training and validation datasets, we use the following settings, chosen in order to stay close to the pretrained weights and also alleviate overfitting to the target domains. We use a binary cross-entropy loss for the task of multi-label classification, an Adam optimizer, an initial learning rate of 1e-6, and a batch size of 16. During each training procedure, we apply early stopping on the validation loss with a patience of 10 epochs to alleviate overfitting by stopping fine-tuning when the validation performance no longer improves. In each case, we choose the model that achieves the lowest validation loss as our final model. We train for 72 epochs on the CBET dataset over the six Ekman emotions, 90 epochs on the GoEmotions dataset over the six Ekman emotions, 66 epochs on the GoEmotions *joy* subcategory dataset, 13 epochs on the GoEmotions *sadness* subcategory dataset, 18 epochs on the GoEmotions *fear* subcategory dataset, and 8 epochs on the GoEmotions *anger* subcategory dataset, in order to achieve these best results in Table 6. Across the six models, the total training procedure converged after approximately 5.5 hours on a single GPU.

B Fine-Tuning Thresholds and Weights

In the hierarchical label mappings in Tables 3, the weights for each term in the linear combinations for each target emotion are by default set to 1. Without any fine-tuning data in the target domain, we let each emotion subcategory have equal weight in determining the value of the target emotion. Additionally, in the evaluation, we let the thresholds for classification of each emotion all be equal to 0.3. However, with the availability of a small in-domain validation dataset, we can improve the classification thresholds as well as the mapping weights. We fine-tune the classification thresholds by choosing a threshold for each target class to maximize the F1 score on that class over the validation dataset.

We fine-tune the mapping weights by successively applying differential evolution to each individual target label mapping to maximize the F1 score on that label over the validation dataset (Storn and Price, 1997). The implementation of the differential evolution algorithm for fine-tuning the mapping weights is provided by Scipy[6]. For each target label mapping, we constrain each weight in $[0, 2]$ in the optimization process, and continue iteratively until the improvements in the label-wise F1 scores are sufficiently small.

Model	**Validation Accuracy**	**Test Accuracy**
CBET-Ekman	0.6558	0.6483
GoEmo-Ekman	0.6966	0.6914
Joy	0.7386	0.7519
Sadness	0.7205	0.7625
Fear	0.9048	0.8878
Anger	0.6541	0.6501

Table 6: Final validation accuracy and final testing accuracy for each of the six fine-tuned twitter-RoBERTa-base-emotion models in our model ensemble

---

[6] `https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.differential_evolution.html`

Not Just Iconic: Emoji Interpretation is Shaped by Use

Brianna O'Boyle
Dept. of Linguistics
San Diego State University
San Diego, CA, USA, 92182
boboyle@sdsu.edu

Gabriel Doyle
Dept. of Linguistics
San Diego State University
San Diego, CA, USA, 92182
gdoyle@sdsu.edu

Abstract

Where do the meaning of emoji come from? Though it is often assumed that emoji are fully iconic, with meanings derived from their visual forms, we argue that this is only one component of their meaning. We surveyed users and non-users of the Chinese social media platform WeChat for their interpretations of emoji specific to WeChat. We find that some emoji show significant differences in their interpretations between users and non-users, as well as how familiar a person is with the specific emoji's use. We argue that this reflects a more complex process for building the meaning of emoji on a platform than pure iconicity.

1 Introduction

How do you know what an emoji means? The answer may seem obvious: what it looks like is what it means. The Unicode emoji U+1F971, for instance, is a yawning face. It corresponds to a universal human symbol for tiredness. In this sense, emoji are different from words, whose forms generally provide little information about their meaning. *Chat* means very different things in English and French, for instance. As a result, emoji may seem to be a fairly trivial part of sentiment analysis, easier than dealing with the complex meanings of a text.

But this simplified division into semantically-transparent emoji and semantically-opaque text is not the whole story. Some words' forms do convey information about their meanings. Onomatopoeia, like *quack* or *meow*, are attempts to transcribe a sound, and aim for semantic transparency. While onomatopoeic words seem like very special cases, recent work has increasingly found evidence of sound symbolism even in words whose wordforms seem completely unrelated to their meaning (see Svantesson, 2017 for a review). Wordforms also correlate with the perceived complexity of the ideas they express (Lewis and Frank, 2016). These subtle links between words' forms and meanings are even used by children as part of language acquisition (Imai et al., 2008).

These results suggest that words are not as semantically opaque as they seem. In a similar vein, then, we might ask if emoji are as semantically transparent as they seem. We examined people's interpretation of emoji from the Chinese social media platform WeChat and found significant deviations in meanings for some emoji between people familiar and unfamiliar with the emoji.

This suggests that emoji interpretation is substantially less transparent than it seems. The meanings of emoji get built up through use, like words' meanings do, and may drift away from their purely iconic interpretations as they amass platform-specific meanings.

Our key finding is that some, but not all, emoji show significantly different interpretations by regular users of a platform than the first impression these emoji convey to novices. These differences are idiosyncratic, and appear to arise from a combination of increasing familiarity with the platform and its cultural context. In some cases, the difference is sufficient to flip the valence of the emoji's interpretation between novices and experienced users. We also propose some possible approaches to closing this gap—or at least mitigating the impact of such misinterpretations.

2 Background

2.1 Emoji meaning and interpretation

Emoji were first created in the 1990s for Japanese cellphones, and have become a prominent component of computer-mediated communication since then. Emoji have a variety of communicative uses, including conveying the tone of a message (Lo, 2008; Hu et al., 2017), adding situational meaning (Kaye et al., 2016), and giving a sense of personality from the author (Sugiyama, 2015). They pro-

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 452–457
July 14, 2023 ©2023 Association for Computational Linguistics

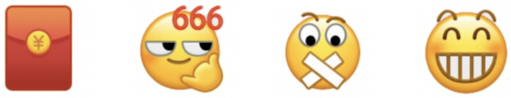


Figure 1: Some of the WeChat-specific emoji.

vide paralinguistic information which the reader then combines with the linguistic information of a message to determine the sentiment of the message, though the result is not always a straightforward combination of the text and emoji components (Tian et al., 2017).

Most importantly for this research, some have proposed that emoji represent a cross-linguistic, cross-cultural communication avenue, due to their iconicity and semantic transparency (Alshenqeeti, 2016; Lu et al., 2016). However, for this to be the case, we need to establish that emoji are actually interpretable across the divides of language, culture, and even social media platforms.

There is some evidence that emoji do not seamlessly straddle these divides. Tigwell and Flatla (2016), for instance, collected people's sentiment ratings on Unicode emoji that were rendered differently on Android and iOS devices. They found significant deviations in how people reacted to these emoji due to their different forms on different devices. Similarly, the intended interpretation of an emoji may rely on idiosyncratic interpretations, as with inside jokes or call-backs (Wiseman and Gould, 2018). Thus, while we can develop emoji sentiment lexicons that give approximate sentiments for an emoji (Kralj Novak et al., 2015), there will be a variety of dimensions in which the actual interpretation will vary from that baseline.

In the present study, we examine how platform familiarity can affect emoji interpretation in a more stripped-down setting. Participants all see the same representation of an emoji, so cross-platform rendering differences will not exist. The emoji are evaluated apart from their text so that there will not be complex interactions between the two. Despite this, we find that familiarity with the emoji and the WeChat platform affect the interpretation of an emoji's sentiment.

2.2 WeChat and its emoji

WeChat is a Chinese messaging, social-media, and payment app with more than one billion users. It is extremely prominent within China, and is used both by Chinese nationals and people of other nationalities within China (e.g., Szurawitzki, 2022 studied German speakers' use of WeChat).

WeChat represents an exciting test case because it maintains its own lexicon of 105 emoji, which range from minor variants of the familiar Unicode emoji to completely unique forms[1]. Some work exists looking at the usage of WeChat emoji by WeChat users. Li and Yang (2018) used a corpus study of WeChat to examine the motivations behind the use of emoji, and found approximately half of the emoji use was to express or intensify an emotion. Liu et al. (2020) studied feelings of embarrassment from sending an emoji that had the opposite meaning of what was intended, reflecting the dangers of emoji misinterpretation. Both of these studies point to the importance of getting accurate understandings of emoji sentiment for understanding the meaning of a message, as well as hinting at the danger of misinterpretation.

3 Experiment

3.1 Experimental design

Survey and participants To determine how well the meaning of emoji were conveyed by their images and how much their use on the platform contributed, we asked 57 participants to rate the sentiment of the 105 WeChat emoji. For each emoji, participants were shown the emoji and asked to place it on a two-dimensional plot with valence on the x-axis and energy on the y-axis. Participants went through a four-round familiarization phase at the start of the survey to make sure they understood the task, placing four words (*happy*, *sad*, *peaceful*, and *angry*) as examples of each quadrant of the plot. Participants also rated their familiarity with the emoji on a five-point scale. At the end of the survey, participants self-reported information about their frequency and length of WeChat use, proficiency in Chinese, length of residency in China, and familiarity with Chinese culture. The survey was reviewed by our institution's IRB process.

Our participants ranged in age from 20 to 72 years old, with a mean age of 32.3. 36 of the participants were WeChat users (mean age 34.3), and 21 of the participants were non-users (mean age 29.0). Unsurprisingly, WeChat use correlated both with participants' self-reported familiarity with Chinese culture and language proficiency. The mean Chinese cultural familiarity, on a six-point scale, was

---

[1] We exclude three legacy emoji imported from QQ, WeChat's predecessor, from our analysis.

4.7 for WeChat users and 2.0 for non-users. The mean Chinese language proficiency, again on a six-point scale, was 3.7 for users and 1.9 for non-users.

Sentiment dimensions The sentiment dimensions of valence and energy (also called “arousal”) were chosen based on previous usefulness in word sentiment (Mohammad, 2018).[2]

Statistical testing We analyzed the data in two ways. For our initial analysis, we divided the participants into 21 users and 36 non-users of WeChat. The visualizations in Figure 2 show this division, with cross-hairs showing the means and standard deviations for each emoji’s rating. This two-way division provides a simple representation of the different interpretations of the emoji between users and non-users. We performed Mann-Whitney U-tests (with Holm corrections for multiple comparisons) on these categorical splits.

To increase our statistical power, we followed up the above analysis with Kendall rank correlation tests between sentiment ratings and participants’ familiarity with each emoji (again, with Holm corrections). Since not all users are equally familiar with all emoji, and non-users may have some familiarity with some emoji (such as those that match Unicode emoji), this gives us a more detailed sense of how much the interpretation is shaped by experience rather than the strict user/non-user dichotomy.

3.2 Results

Different interpretations Of the 105 emoji we tested, eight had significant correlations between familiarity with the emoji and the valence ratings of the emoji after a Holm correction for multiple comparisons. Figure 2 shows the difference between user and non-user means for the emoji with significant valence correlations. The top left emoji (clapping) had a significant energy difference in addition to its valence difference.

One clear pattern is that familiarity with an emoji generally led to more extreme ratings; participants gave more neutral ratings to emoji that they were less familiar with. Interestingly and importantly, we see that non-users occasionally misunderstand the sign of the valence. The most extreme example is the top-right emoji in Figure 2, which shows strong but nearly opposite valence ratings between users and non-users of WeChat. This emoji is used

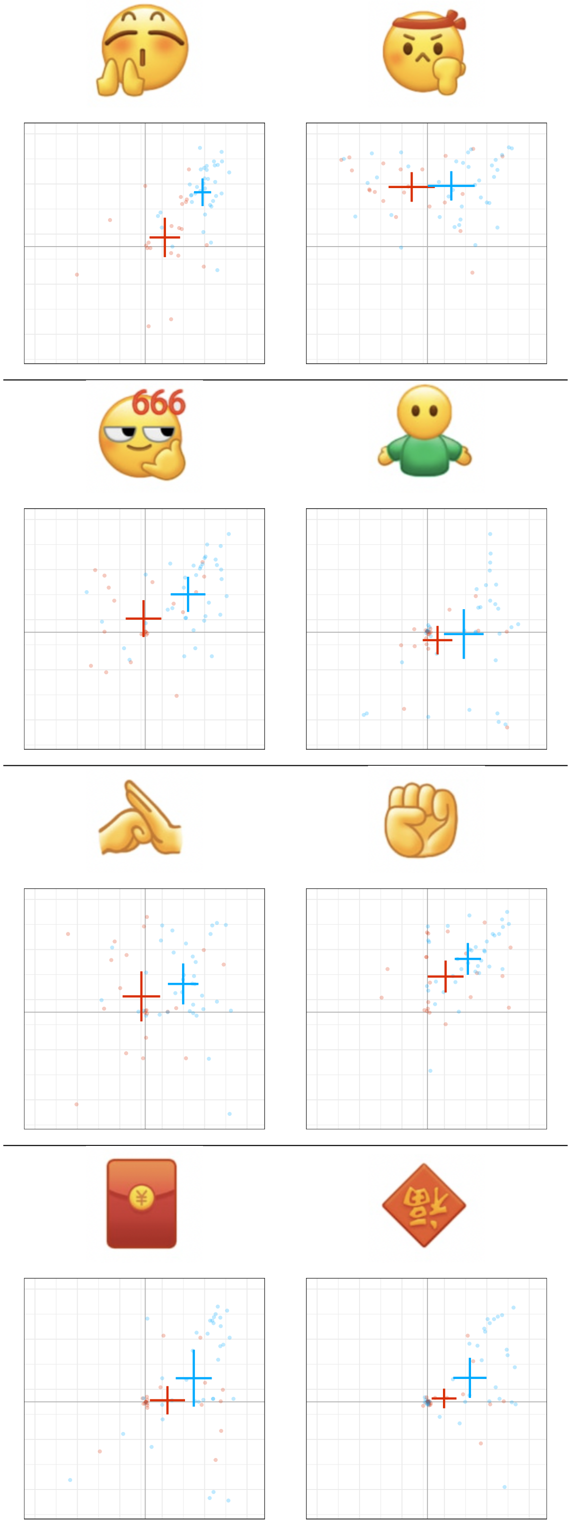


Figure 2: Emoji with significant valence-familiarity correlations. Blue dots are ratings from WeChat users, Red dots from non-users. The crosshairs show the mean and standard deviations for each group’s ratings. The x-axis shows valence ratings; the y-axis shows energy. All emoji had Holm-corrected significances at $p < .01$ except for the 666-emoji, which had $p < .05$.

---

[2] Sentiment work often includes a third dimension, dominance, but we omitted this factor due to its correlations with energy, and to keep the task visually simple for participants.

to express encouragement and likely has relation to the Chinese expression *jīa yóu*, which literally translates to "add oil" but is used as a cheer like in sporting events, or to mark a concerted effort. However, to non-users, the emoji appears to be raising a fist to fight the interlocutor.

Some of these emoji, like *jīa yóu*, require knowledge of Chinese culture to properly assess their meanings. For instance, the "666" emoji in Western cultures is likely to call to mind the Christian devil and negative emotions; however, it also represents Chinese internet slang for "awesome", explaining some of the interpretive gap.

But not all the emoji misinterpretations can be explained through cultural knowledge. For instance, the green-shirted man is used to signal a hug on WeChat, while non-users find its gesture incomprehensible, regardless of their general Chinese cultural knowledge. Similarly, the clapping face (top-left) and fist (third row, right) lack obvious cultural explanations for the different interpretations; these appear to have platform-specific interpretations that deviate from the information available in their visual forms. Kendall tests found significant correlations between valence and participants' self-reported Chinese cultural familiarity for only four of these eight emoji (clapping, *jīa yóu*, 666, and fist-in-hand), so cultural knowledge alone is not enough to account for the differences.

Similiar interpretations The emoji with significant interpretative differences generally show stronger, more positive valence ratings for users than non-users. As a result, one might argue that these differences are nothing more than users being more confident in their ratings than non-users. However, if this were the case, we would expect to see a similar pattern on all the emoji (though not necessarily to statistically-significant levels). To the contrary, we found that users and non-users agree very closely in their interpretation of a number of emoji. This suggests that the deviations in Figure 2 are not just a result of WeChat users growing more confident in their sentiment ratings, but actually developing a richer meaning for some emoji based on their use on WeChat. Figure 3 shows the seven emoji with the smallest distance between the users' and non-users' estimates of valence and energy, and illustrates the close agreement between two of them; users and non-users align closely on both mean and standard deviation.

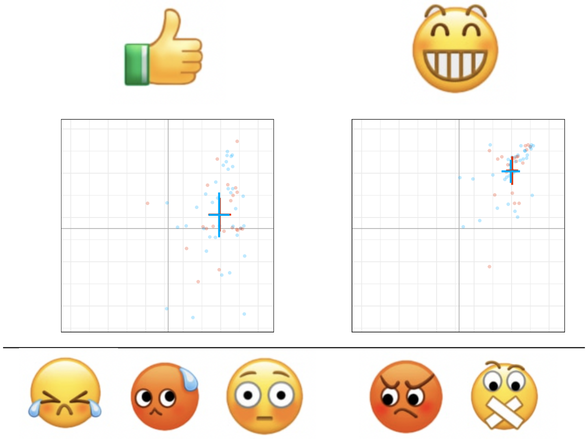


Figure 3: Emoji with the smallest differences between users and non-users, with differences plotted for two of them. The x-axis shows valence ratings; the y-axis shows energy.

4 Discussion

Our key finding was that familiarity with an emoji can, in some but not all cases, lead to significant differences in the interpretation of that emoji. Such differences have been found in previous studies of emoji, but the differences were generally tied to changes in the emoji's form (Tigwell and Flatla, 2016), multiple meanings for the emoji (Miller et al., 2016), or personal experience (Wiseman and Gould, 2018). We found differences even with the same visual forms, and separated from any textual content that might influence the emoji's meanings.

This suggests that emoji interpretation is not strictly based on form and iconicity, but that emoji develop richer meanings through use on a platform. They are interpreted through a mixture of form and use, similar to words or other linguistic elements, albeit with a much heavier impact of iconic form. As such, the study of emoji interpretation may gain from treating them through a more linguistic lens.

4.1 Avoiding misinterpretation

Our results also suggest that the danger of emoji misinterpretation is most acute for new users of a platform, whether because they do not have experience with the range of available emoji, do not know the "dialect" differences between platforms, or simply misinterpret the images the emoji convey.

Social media platforms may wish to acclimate new users to the established meanings of emoji on a platform, to help them avoid misinterpretations. While both Unicode and WeChat emoji have textual labels, these do not adequately convey nuanced meanings, and alternative methods of conveying richer senses of the emoji sentiments are required.

One possibility is to show users a visualization of the baseline sentiments of an emoji, whether assessed automatically through sentiment analysis of messages containing these emoji or by asking users to participate in a rating experiment like this one.

Another possibility is to provide a "glossary", with samples of the emoji in use, as dictionaries do with words. This could be curated, with emoji lexicographers selecting sample messages that illustrate the range of meanings, or automatically generated based on recent messages that use the emoji on the platform.

4.2 Limitations

The current study was based on primarily American respondents, though approximately half had lived in China for at least a year (including 75% of WeChat users). As a result, our data does not fully reflect the wide range of users or non-users of WeChat. We encountered correlations between our participants' familiarity with WeChat, the individual WeChat emoji, Chinese culture, and the Chinese language. WeChat use and emoji familiarity had the strongest impacts on emoji interpretation differences, but further work is needed to control for the impacts of these other factors.

Additionally, we limited our analysis to emoji in isolation, as our goal was to assess whether the most basic interpretation of the emoji still relied on experience with the emoji. Emoji are sometimes used by themselves without text, so these results will apply to some real-world usage. But of course, in their general use, emoji tend to appear in richer conversational contexts and are accompanied by other linguistic information. As such, it is not certain that the observed user/non-user sentiment differences will persist for emoji used in conjunction with texts. That said, the assessment of these emoji in isolation can serve as a baseline for future research examining how their sentiment differs in real-world conversations, especially in cases of sarcasm, hyperbole, or irony.

References

Hamza Alshenqeeti. 2016. Are emojis creating a new or old visual language for new generations? A socio-semiotic study. *Advances in Language and Literary Studies*, 7(6).

Tianran Hu, Han Guo, Hao Sun, Thuy-vy Nguyen, and Jiebo Luo. 2017. Spice Up Your Chat: The Intentions and Sentiment Effects of Using Emojis. *Proceedings of the International AAAI Conference on Web and Social Media*, 11(1):102–111.

Mutsumi Imai, Sotaro Kita, Miho Nagumo, and Hiroyuki Okada. 2008. Sound symbolism facilitates early verb learning. *Cognition*, 109(1):54–65.

Linda K. Kaye, Helen J. Wall, and Stephanie A. Malone. 2016. "Turn that frown upside-down": A contextual account of emoticon usage on different virtual platforms. *Computers in Human Behavior*, 60:463–467.

Petra Kralj Novak, Jasmina Smailović, Borut Sluban, and Igor Mozetič. 2015. Sentiment of Emojis. *PLOS ONE*, 10(12):e0144296.

Molly L Lewis and Michael C Frank. 2016. The length of words reflects their conceptual complexity. *Cognition*, 153:182–195.

Li Li and Yue Yang. 2018. Pragmatic functions of emoji in internet-based communication—a corpus-based study. *Asian-Pacific Journal of Second and Foreign Language Education*, 3(1):16.

Liyuan Liu, Yen Hsu, Jie Zhang, and Qianling Jiang. 2020. A study on the embarrassment of senders who missend emojis with opposite meanings on social apps: Taking WeChat as an example. *Psicologia: Reflexão e Crítica*, 33(1):20.

Shao-Kang Lo. 2008. The Nonverbal Communication Functions of Emoticons in Computer-Mediated Communication. *CyberPsychology & Behavior*, 11(5):595–597.

Xuan Lu, Wei Ai, Xuanzhe Liu, Qian Li, Ning Wang, Gang Huang, and Qiaozhu Mei. 2016. Learning from the ubiquitous language: An empirical analysis of emoji usage of smartphone users. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 770–780, Heidelberg Germany. ACM.

Hannah Miller, Jacob Thebault-Spieker, Shuo Chang, Isaac Johnson, Loren Terveen, and Brent Hecht. 2016. "Blissfully Happy" or "Ready to Fight": Varying Interpretations of Emoji. In *Proceedings of the 10th International Conference on Web and Social Media, ICWSM 2016*, pages 259–268.

Saif Mohammad. 2018. Obtaining Reliable Human Ratings of Valence, Arousal, and Dominance for 20,000 English Words. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 174–184, Melbourne, Australia. Association for Computational Linguistics.

Satomi Sugiyama. 2015. Kawaii meiru and Maroyaka neko: Mobile emoji for relationship maintenance and aesthetic expressions among Japanese teens. *First Monday*.

Jan-Olof Svantesson. 2017. Sound symbolism: The role of word sound in meaning. *WIREs Cognitive Science*, 8(5).

Michael Szurawitzki. 2022. The Chinese Messaging Application WeChat as Used by German Speakers in China: Usage Practices, Multilingual Environments, Emojis, and Beyond. *Linguistik Online*, 113(1):111–152.

Ye Tian, Thiago Galery, Giulio Dulcinati, Emilia Molimpakis, and Chao Sun. 2017. Facebook sentiment: Reactions and emojis. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 11–16, Valencia, Spain. Association for Computational Linguistics.

Garreth W. Tigwell and David R. Flatla. 2016. Oh that's what you meant!: Reducing emoji misunderstanding. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*, pages 859–866, Florence Italy. ACM.

Sarah Wiseman and Sandy J. J. Gould. 2018. Repurposing Emoji for Personalised Communication: Why [Pizza] means "I love you". In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–10, Montreal QC Canada. ACM.

The Paradox of Multilingual Emotion Detection

Luna De Bruyne
LT^3, Language and Translation Technology Team
Ghent University
`luna.debruyne@ugent.be`

Abstract

The dominance of English is a well-known issue in NLP research. In this position paper, I turn to state-of-the-art psychological insights to explain why this problem is especially persistent in research on automatic emotion detection, and why the seemingly promising approach of using multilingual models to include lower-resourced languages might not be the desired solution. Instead, I campaign for the use of models that acknowledge linguistic and cultural differences in emotion conceptualization and verbalization. Moreover, I see much potential in NLP to better understand emotions and emotional language use across different languages.

1 Introduction

Variation and diversity are inherent to human life, not least to human language. Yet, machine learning approaches used in natural language processing (NLP) usually ignore this variation and are biased to a (consciously or subconsciously imposed) norm. Mohammad (2022) stressed that current NLP applications therefore often amplify societal inequalities and "lead to more adverse outcomes for those that are already marginalized". Indeed, we have known for some time that NLP applications show several biases, e.g., racial bias in conversational agents (Sap et al., 2019) or gender bias in machine translation (Savoldi et al., 2021). Mohammad (2022) therefore campaigns for introducing ethics sheets for AI tasks, in which diversity should be one point to be addressed, including a discussion of the design choices that impact diverse groups of people.

Besides inequality across social groups, Søgaard (2022) recently pointed at the inequality across languages as an unwanted bias in NLP (around two thirds of NLP research at top venues would be devoted exclusively to English, which has not changed over the last 10 years). However, instead of merely acknowledging these biases, he argues that it is simple to mitigate inequality amplification, for which he proposes three strategies inspired by policies for reducing carbon emissions: (i) an NLP Cap and Trade, (ii) an NLP Carbon Tax, and (iii) NLP Car-Free Sundays. As the language bias is generally towards English, this would in practice mean to (i) distribute quota for publications on English, (ii) impose a cost on researchers submitting papers on English or (iii) a one-year ban on English models. These measures should encourage groups to work on NLP systems and resources for other languages than English.

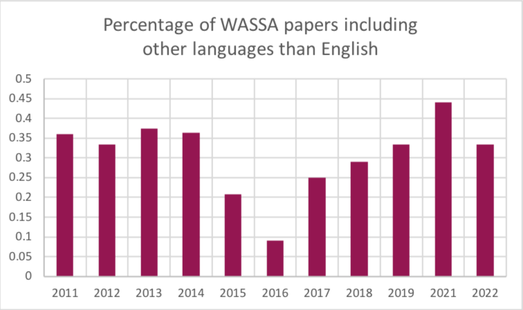


Figure 1: Proportion of papers (presented at WASSA between 2011 and 2022) including other languages than English.

For work on emotion detection (also referred to as automatic emotion recognition or AER)[1] as well, we observe a huge bias towards English resources and systems. In order to get some insight in the variety of languages addressed in research on emotion detection and related research fields, I analyzed the papers that were presented at the Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis (WASSA) since 2011[2], and counted the number of papers that included other languages than English.[3] My findings,

---

[1] I will use these terms interchangeably.

[2] Except for the first WASSA edition in 2010, the contributions of all editions are found on https://aclanthology.org/venues/wassa/.

[3] I manually scanned all WASSA publications, except ab-

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 458–466
July 14, 2023 ©2023 Association for Computational Linguistics

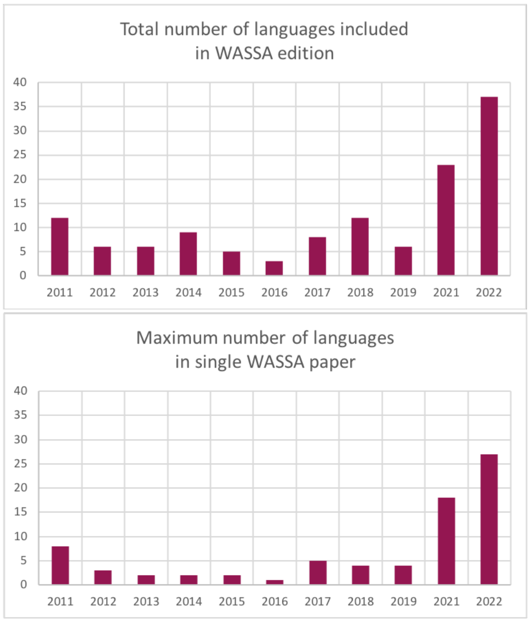


Figure 2: Number of languages that were included in the papers presented at WASSA between 2011 and 2022.

shown in Figure 1, are in line with the numbers reported by Søgaard (2022): the proportion of papers that not exclusively focused on English fluctuates around one third and did not increase over the years. However, I did observe that there was a remarkable increase in the past two years concerning the total number of languages that were included and the maximum number of languages across papers (see Figure 2).

I believe that this positive trend is fueled by large language models like multilingual BERT (Devlin et al., 2019), which lend themselves perfectly to (zero or few-shot) transfer learning. Indeed, the high number of languages at WASSA 2021 and 2022 is largely due to just three papers that include many different languages: Lamprinidis et al. (2021), Bianchi et al. (2022) and Rajda et al. (2022) respectively include 18, 19 and 27 languages, each of them performing experiments using multilingual models.

However, even though multilingual models might seem promising for tackling NLP tasks for other (and lower-resourced) languages than English, the use of multilingual models result in a paradox in the case of emotion detection: using multilingual emotion detection models inherently assumes that different languages deal with emotions the same way. This idea may be in line with classical views on emotion analysis, but goes completely against state-of-the-art evidence in psychology showing that emotions are not universal, but rather culture (and language) dependent (Barrett, 2017; Mesquita et al., 2016).

Therefore, I believe the real challenge does not lie in attracting more research on a larger number or greater variety of languages, but in studying emotion detection without falling into universalist ideas and instead acknowledging differences in emotional conceptualization and verbalization across languages. This can be achieved by creating valid datasets (original data written and annotated by native speakers), with label sets that are adjusted to the target language (using native emotion words or emotion representations that go beyond the anglocentric basic emotions).

In this position paper, I will discuss state-of-the-art psychological findings and their implications for emotion detection in NLP (Section 2). I will then expand on these implications by discussing them in the light of current papers in AER research (Section 3). Next, I will propose some research directions that can be taken in AER to better align with psychological evidence (Section 4). Finally, my viewpoints are summarized in some concluding thoughts (Section 5).

2 What psychology teaches us and what it means for AER

For a very long time, a universalist view on emotions prevailed. In such a view, it is believed that the way emotions are conceptualized and experienced is the same across different cultures and that emotions are biologically hard-wired. Especially the work on facial expressions by Ekman and his colleagues, in which participants from different cultures made similar decisions when asked to match emotion words or emotional stories with facial expressions of *anger*, *disgust*, *fear*, *joy*, *sadness* and *surprise*, consolidated the idea of universal emotions (Ekman et al., 1969; Ekman and Friesen, 1971).

However, experiments like Ekman's are biased by the Western perspective of the researcher and influenced by the used research methods (e.g., the

---

stracts of invited talks and submissions for shared tasks (but, for each shared task, I did include the task description paper). If the paper included other languages than English, it was usually mentioned in the paper title or in the abstract, although sometimes I had to read the dataset description to find out. For a remarkable number of papers, no language was mentioned at all. In almost all of these cases, the papers used English data. Plots for the distribution of languages in all WASSA discussions are shown in Figure 4 in the Appendix.

choice of emotion words to match with; posed instead of spontaneous emotion expressions). More recent experiments using a greater diversity in research methods and contexts, reveal diversity instead of universality (Gendron et al., 2014).

Moreover, a critical review of literature about the biological basis of emotion categories by Barrett (2006) indicates that evidence for the existence of such universal basic emotions is inconclusive. Rather, there is "cultural variation in the experience of emotion that is intrinsically driven by cultural differences in emotion categories and concepts" (Barrett, 2006, p.38). In a similar vein, Mesquita et al. (2016) claim that emotional experience is culturally constructed, which shows itself in cultural differences regarding how people communicate and talk about their emotions.

Variation in emotion conceptualization and experience on the hand, and concomitantly, emotion expression and verbalization on the other hand, both have consequences for automatic emotion recognition in NLP.

Diversity in emotion conceptualization

Not all cultures dispose of the same emotion concepts. There exist many examples of emotion concepts in specific languages that do not seem to have a translation in other languages, e.g., *toska* in Russian (described as spiritual anguish without a specific cause), *saudade* in Portuguese (described as a somewhat melancholic feeling of incompleteness), *lítost* in Czech (a state of agony and torment created by the sudden sight of one's own misery) or *fago* in Ifaluk (which has characteristics shared with the English concepts *love*, *compassion* and *sadness*). Even for concepts that are claimed to be 'basic emotions', not all languages have a word – e.g., there exists no word for *sadness* in Tahiti (Levy, 1984).[4]

Besides the untranslatability of some emotion words, there are also many differences in the connotations and meanings of emotion terms across languages (Mesquita et al., 1997; Pavlenko, 2008; Wierzbicka, 2009). The concept of *anger*, for example, is hardly the same as *gnev*, although they are usually glossed as translation equivalents in English-Russian dictionaries (Wierzbicka, 1998).

That there is variation between emotion concepts across languages, is not just because the emotion words we use to refer to them are not perfect translation equivalents (which is an inherent problem related to translation in general, and not only to the translation of emotion words), but because emotions are culturally constructed (Mesquita et al., 2016). For example, in studies comparing the emotion conceptualization between inhabitants from the United states and Japan, it was found that emotion concepts arise from the *individual* in the perspective of American respondents, while they arise from the *relationships* between individuals in Japanese respondents (Uchida et al., 2009). This reflects how in some cultures (e.g., in Japan), processes at the level of the collective are more important for constructing emotions, while in other cultures (e.g., the U.S.) individual-level processes prevail (Mesquita et al., 2016).

Implications for AER: As emotion concepts are dependent on the culture we live in and the language we speak, we should design our datasets and models accordingly. Native speakers should label texts, with emotion labels that make sense to them. We should not use the Ekman emotion taxonomy as the basis of AER without motivation, as the claim that these emotions would be universal has been disproved. Moreover, there is no reason to believe that these emotions have the same meaning as they have for speakers of English.

Diversity in emotion verbalization

The way we conceptualize and experience our emotions has of course a huge impact on how we express and verbalize them. Again, the distinction between individualistic and collectivist cultures is important. People in individualistic cultures seem to be more openly conveying emotional feelings and use a more expressive style than people from collectivist cultures, which is illustrated by the reticence of verbal and non-verbal expression of the emotion *love* by Chinese people compared to Americans (Caldwell-Harris et al., 2013). Moreover, it was found that there are several linguistic differences in the emotional expression between people from individualistic cultures – where emotion terms are related to the self and the use of nouns and adjec-

---

[4] The fact that a language does not have a word for specific emotion concepts, does not necessarily mean that people speaking that language cannot *conceptualize* such an emotion. However, according to Barrett (2017), conceptualization is a prerequisite for emotional *experience*. Whether Tahitians can experience *sadness* thus depends on whether they can conceptualize it. We are not sure whether having a word for a concept is necessary for having the concept, but it seems the case that having a word makes conceptualization easier (Barrett, 2017).

tives is more prominent – and collectivist cultures – where emotion terms are more often used to refer to relationships intead of the individual, and more interpersonal verbs are used (Semin et al., 2002; Mesquita et al., 2016).

However, there are even more subtle differences that have nothing to do with the individualistic-collectivist dichotomy. Languages can have very characteristic strategies for emotion verbalization, e.g., using diminutive, augmentative and pejorative suffixes in Spanish or Portuguese (Rudolph, 1990), or emotion verbalization that is focused on the human body in Russian (Wierzbicka, 1999). Also emojis, which are a common strategy to convey emotions in informal writing, show much divergence between languages and countries, sometimes even between countries that are geographically close to each other and in which the same language is spoken, like Mexico and Columbia (Kejriwal et al., 2021).

Implications for AER: As emotion verbalization is dependent on the culture we live in and the language we speak, models should be trained on texts that are written in the language for which we want to use the developed emotion detection system. Both training and evaluation data should be written and labeled by native speakers, as only native speakers might pick up on language-specific emotion verbalization strategies.

3 What we are really detecting in NLP

I will expand on the implications mentioned in the previous section by discussing recent papers dealing with multilingual AER. I will zoom in on three important aspects of automatic emotion detection: the data, the labels, and the models.

I selected three WASSA submissions from 2021 and 2022, namely those that included the highest number of languages: the papers of Rajda et al. (2022), Bianchi et al. (2022), and Lamprinidis et al. (2021). The first one includes an assessment of sentiment analysis in 27 languages (I will refer to this work as MSA, standing for multilingual sentiment analysis); the second one presents XLM-EMO, a multilingual emotion detection model evaluated on 19 languages; and the last one presents Universal Joy, an emotion detection dataset including 18 languages. Additionally, I will also discuss the work by Öhman et al. (2020), who present the multilingual emotion detection dataset XED, including 32 languages.

The data: Both in MSA (Rajda et al., 2022) and Universal Joy (Lamprinidis et al., 2021), data is used that was originally written in the target languages. While existing sentiment datasets are used in MSA, Universal Joy is created by scraping Facebook posts based on the Facebook-specific feelings tags. Also in XLM-EMO (Bianchi et al., 2022), original data from existing emotion datasets is used, although the data for some languages (French, German and Hindi) was machine-translated (from Spanish to French and German, and from English to Hindi, respectively).

I believe the use of (machine) translations is problematic, as it neglects language-specific characteristics of emotion verbalization. Moreover, as shown by Troiano et al. (2020), emotional connotations are partly lost in the machine translation process. Also for XED (Öhman et al., 2020), translated (although human-translated) data was used, namely in the form of subtitles. Although the use of translated subtitles allows for the creation of a parallel corpus – which is in itself a compelling idea – it is far from ideal to use non-original data, as such data – even if it is translated by humans – might be biased towards the source language in terms of emotion characteristics.

The labels: I will not focus on MSA here, as it uses sentiment labels instead of fine-grained emotion labels. Judging from the three other papers, there is still work to be done regarding the handling of emotion labels in multilingual datasets. It seems to be common to treat the labels across languages as one and the same category. In XLM-EMO, for example, datasets from various languages with different label sets are merged by removing instances that did not fit the labels *anger*, *fear*, *joy* or *sadness*. However, based on the literature cited in Section 2, it is hardly likely that *anger* in one language has a perfectly overlapping meaning with its translation in a different language. Moreover, this approach results in a huge loss of data and information. Universal Joy relies on the simple but nice idea of employing Facebook feelings-tags as labels. As Facebook users attach these tags themselves when posting messages, it is ensured that the labels correspond to the feelings of the writers of the posts. However, the original tags (27 different tags initially), were mapped to the five categories *anger*, *anticipation*, *fear*, *joy*, and *sadness*. The mapping happened in the same way for all languages, but again, it is not certain that these mappings make sense at the level of the individual languages. In XED, an even more risky approach is used, consisting of projecting labels that were manually annotated for the English instances to the translations of those instances in the other languages. Apart from the fact that emotion labels might not be comparable across languages, this approach assumes that utterances have the same emotional connotation, irrespective of in which language or culture it is uttered. In each of these papers, the classical view on emotion prevails, assuming that emotion categories are universal.

The models: In MSA, XLM-EMO, and Universal Joy, a pre-trained multilingual model (e.g., mBERT (Devlin et al., 2019) or XML-R (Conneau et al., 2020)) is fine-tuned on the multilingual datasets. In each of these papers, the default setting is to fine-tune on all languages at the same time. This neglects the fact that emotions are verbalized differently in different languages, and moreover enhances the classical emotion view by modeling emotion concepts as if they were one and the same category shared by all languages. Instead of acknowledging variation across languages, multilingual systems are modelling artificial universal emotion categories.

Moreover, the multilingual models in the discussed papers show a bias towards English: in each of the three datasets, there is more fine-tuning data for English than for the other languages. In fact, pre-trained models itself are already biased towards English anyway: in mBERT, for example, 21% of the training data is English.[5] This English bias is also evidenced by the zero-shot experiments described in XLM-EMO: Bianchi et al. (2022) fine-tuned mBERT on all languages except on a target language, which is respectively English, Arabic and Vietnamese. In contrast to Arabic and Vietnamese (where a language-specific model outperforms zero-shot experiments), there is almost no difference between the zero-shot performance and a language-specific model in the case of English as target language, indicating that the pre-trained model already contains information on English.

In further experiments on Universal Joy, cross-lingual fine-tuning effects are investigated: Lamprinidis et al. (2021) compare multilingual fine-tuning of mBERT (i.e., fine-tuning on all languages of the dataset) with monolingual fine-tuning (fine-tuning on only the target language). They observe positive cross-lingual effects, meaning that performance increases when fine-tuning data from other languages is added (especially when there are syntactic and typological similarities between fine-tuning and target languages). However, these positive effects were only found when the size of the target language dataset was small. For large target language datasets, including multiple languages for fine-tuning did not result in an improvement.

The experiments in the discussed papers show that multilingual models, fine-tuned on a variety of languages, can improve performance. But performance on what? The answer is: performance on classifying texts – which are sometimes not even originally written in the target language – into artificial emotion categories – that are modeled across different languages at the same time and might not make sense according to the emotion conceptualization in that target language. Moreover, the increased performance seems to be just as easy to reach by gathering more data in the target language itself.

I am not claiming that these multilingual models are by definition useless for AER. They can be a compromise in real low-resourced situations. However, we should face that, although such multilingual models are driven by a very inclusive idea, they might not be inclusive at all, and may disadvantage languages that are verbalizing and conceptualizing emotions in a different way than it is done in English.

Therefore, rather than investing in multilingual models, we should invest in better monolingual resources that are not created from an Anglocentric (or by extension: Western) perspective. Qualitative monolingual resources (respecting language-specific ways of emotion conceptualization and verbalization), are moreover needed to investigate how multilingual models really deal with the language-specificity of emotions.

4 The real challenge in AER

Similarly to Søgaard (2022), I believe we should act against the dominance of English in NLP and more specifically in emotion detection. However, nudging researchers to publish papers on other languages than English or to create multilingual datasets is not sufficient, or at least not if we do not let go of the Anglocentric perspective on emotion (Wierzbicka, 2009). Instead, we should be

---

[5] https://github.com/google-research/bert/blob/master/multilingual.md

aware of the Anglocentric bias that multilingual models have, and find out how language-specific emotion verbalization and conceptualization affect multilingual emotion detection. Maybe, language-specific information is or can be employed by multilingual models, but at this point, our knowledge about that is too limited. I therefore see a promising research line in using NLP to investigate how emotion verbalization exactly differs across languages, and how emotion detection models deal with such language-specific information.

Although the first question is mainly a question for psychology, computational methods can help to solve this puzzle, as illustrated by the work of Jackson et al. (2019), who performed a network analysis on emotion words in 2,474 languages to get more insight in how emotion concepts vary across languages, and Markov et al. (2018), who found evidence that emotional features depend on someone's native language by analyzing over 1,000 essays written in English by non-native speakers.

However, to investigate emotion verbalization across languages, and to create models that can deal with those differences, the prime concern is to create valid data, that is, data that is originally created in that language and annotated by native speakers. Such annotations (and thus label sets) should be adjusted to the target language, which is impossible if we keep on using English words and theories like Ekman's set of basic emotions.

Nonetheless, I do see the perks in using annotations that are comparable across languages. I therefore want to break a lance for using dimensional emotion representations like the circumplex model of affect (Russell, 1980; Barrett, 2017), instead of emotion categories. The circumplex model consists of two axes, namely *pleasure* and *arousal*, representing *core affect*. Core affect feelings lie at the heart of emotional episodes, which makes that specific emotion words are associated with specific states of core affect. The English emotion concept *anger*, for example, is prototypically associated with low valence and high arousal, but translations of emotion words might have other associations with core affect in other cultural contexts. Therefore, they can be a compelling approach in comparing emotional states (and performances of emotion detection models) across languages. In the work of Preoţiuc-Pietro et al. (2016), Buechel and Hahn (2016) and De Bruyne et al. (2021), such dimensional emotion representations were already successfully used in the context of emotion detection. In line with the ambition of making emotion detection more language inclusive, I therefore believe that combining core affect with language-specific emotion labels might be the way ahead.

5 Conclusion

In this position paper, I addressed a known issue in NLP, namely the dominance of English. I discussed this issue in the light of automatic emotion detection and argued that this dominance is not limited to the small number of papers that includes other languages than English, but is also reflected in the way current datasets and models are used. It is thus not enough to encourage research on other languages than English, but to address these languages the right way, without assuming that emotions are conceptualized and verbalized in a universal way. Therefore, it is crucial to create valid data, i.e., original data from the target language (not translated) and annotated by native speakers. Label sets should be adjusted to the target language, using native emotion words and preferably combined with labels for core affect.

Moreover, I see much potential in NLP to better understand how emotional language use differs across languages. That information can subsequently help to reveal how current multilingual models deal with such differences, or even to make them more language inclusive.

Limitations

As this is a position paper, I mainly provide thoughts here, and do not include any experiments or actions myself.

Although the goal of this paper is to combat biases in AER, it is limited to discussing the dominance of English. Other biases, like the bias towards social media texts, or the tendency to ignore neurodiversity and conditions like alexithymia and autism spectrum disorder, are not addressed in this paper.

The counting study I performed to demonstrate that the number of papers dealing with other languages than English does not increase – contrary to the number of languages that are addressed, which does show an upward trend – is only based on papers presented at the Workshop on Computational Approaches to Subjectivity and Sentiment Analysis. Maybe other patterns could be discovered when analyzing the papers of other venues.

References

Lisa Feldman Barrett. 2006. Solving the emotion paradox: Categorization and the experience of emotion. *Personality and social psychology review*, 10(1):20–46.

Lisa Feldman Barrett. 2017. The theory of constructed emotion: An active inference account of interoception and categorization. *Social Cognitive and Affective Neuroscience*, 12(1):1–23.

Federico Bianchi, Debora Nozza, and Dirk Hovy. 2022. XLM-EMO: Multilingual emotion prediction in social media text. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 195–203, Dublin, Ireland. Association for Computational Linguistics.

Sven Buechel and Udo Hahn. 2016. Emotion analysis as a regression problem – Dimensional models and their implications on emotion representation and metrical evaluation. *Proceedings of the Twenty-second European Conference on Artificial Intelligence*, pages 1114–1122.

Catherine Caldwell-Harris, Ann Kronrod, and Joyce Yang. 2013. Do more, say less: Saying “I love you” in Chinese and American cultures. *Intercultural Pragmatics*, 10(1):41–69.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Luna De Bruyne, Orphee De Clercq, and Veronique Hoste. 2021. Emotional RobBERT and insensitive BERTje: Combining transformers and affect lexica for Dutch emotion detection. In *Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 257–263, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Paul Ekman and Wallace V Friesen. 1971. Constants across cultures in the face and emotion. *Journal of personality and social psychology*, 17(2):124.

Paul Ekman, E Richard Sorenson, and Wallace V Friesen. 1969. Pan-cultural elements in facial displays of emotion. *Science*, 164(3875):86–88.

Maria Gendron, Debi Roberson, Jacoba Marietta van der Vyver, and Lisa Feldman Barrett. 2014. Perceptions of emotion from facial expressions are not culturally universal: Evidence from a remote culture. *Emotion*, 14(2):251.

Joshua Conrad Jackson, Joseph Watts, Teague R Henry, Johann-Mattis List, Robert Forkel, Peter J Mucha, Simon J Greenhill, Russell D Gray, and Kristen A. Lindquist. 2019. Emotion semantics show both cultural variation and universal structure. *Science*, 366(6472):1517–1522.

Mayank Kejriwal, Qile Wang, Hongyu Li, and Lu Wang. 2021. An empirical study of emoji usage on Twitter in linguistic and national contexts. *Online Social Networks and Media*, 24:100149.

Sotiris Lamprinidis, Federico Bianchi, Daniel Hardt, and Dirk Hovy. 2021. Universal joy a data set and results for classifying emotions across languages. In *Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 62–75, Online. Association for Computational Linguistics.

Robert I Levy. 1984. The emotions in comparative perspective. *Approaches to emotion*, pages 397–412.

Ilia Markov, Vivi Nastase, Carlo Strapparava, and Grigori Sidorov. 2018. The role of emotions in native language identification. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 123–129, Brussels, Belgium. Association for Computational Linguistics.

Batja Mesquita, Michael Boiger, and Jozefien De Leersnyder. 2016. The cultural construction of emotions. *Current Opinion in Psychology*, 8:31–36. Culture.

Batja Mesquita, Nico H Frijda, and Klaus R Scherer. 1997. Culture and emotion. *Handbook of cross-cultural psychology*, 2:255–297.

Saif Mohammad. 2022. Ethics sheets for AI tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8368–8379, Dublin, Ireland. Association for Computational Linguistics.

Emily Öhman, Marc Pàmies, Kaisla Kajava, and Jörg Tiedemann. 2020. XED: A multilingual dataset for sentiment analysis and emotion detection. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6542–6552, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Aneta Pavlenko. 2008. Emotion and emotion-laden words in the bilingual lexicon. *Bilingualism: Language and cognition*, 11(2):147–164.

Daniel Preoţiuc-Pietro, H. Andrew Schwartz, Gregory Park, Johannes Eichstaedt, Margaret Kern, Lyle Ungar, and Elisabeth Shulman. 2016. Modelling valence and arousal in Facebook posts. In *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 9–15, San Diego, California. Association for Computational Linguistics.

Krzysztof Rajda, Lukasz Augustyniak, Piotr Gramacki, Marcin Gruza, Szymon Woźniak, and Tomasz Kajdanowicz. 2022. Assessment of massively multilingual sentiment classifiers. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 125–140, Dublin, Ireland. Association for Computational Linguistics.

Elisabeth Rudolph. 1990. Portuguese diminutives as special indicators of emotions. *Grazer Linguistische Studien*.

James A Russell. 1980. A circumplex model of affect. *Journal of Personality and Social Psychology*, 39(6):1161–1178.

Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A. Smith. 2019. The risk of racial bias in hate speech detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1668–1678, Florence, Italy. Association for Computational Linguistics.

Beatrice Savoldi, Marco Gaido, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2021. Gender bias in machine translation. *Transactions of the Association for Computational Linguistics*, 9:845–874.

Gün R Semin, Carien A Görts, Sharda Nandram, and Astrid Semin-Goossens. 2002. Cultural perspectives on the linguistic representation of emotion and emotion events. *Cognition & Emotion*, 16(1):11–28.

Anders Søgaard. 2022. Should we ban English NLP for a year? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5254–5260, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Enrica Troiano, Roman Klinger, and Sebastian Padó. 2020. Lost in back-translation: Emotion preservation in neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4340–4354, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yukiko Uchida, Sarah S. M. Townsend, Hazel Rose Markus, and Hilary B. Bergsieker. 2009. Emotions as within or between people? Cultural variation in lay theories of emotion expression and inference. *Personality and Social Psychology Bulletin*, 35(11):1427–1439.

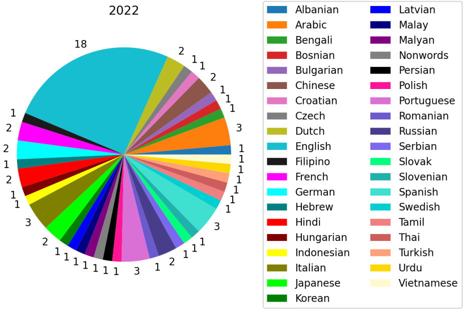
Anna Wierzbicka. 1998. “Sadness” and “anger” in Russian: The non-universality of the so-called “basic human emotions”. In Angeliki Athanasiadou and Elzbieta Tabakowska, editors, *Speaking of Emotions*, pages 3–28. De Gruyter Mouton, Berlin, New York.

Anna Wierzbicka. 1999. *Emotions across languages and cultures: Diversity and universals*. Cambridge University Press.

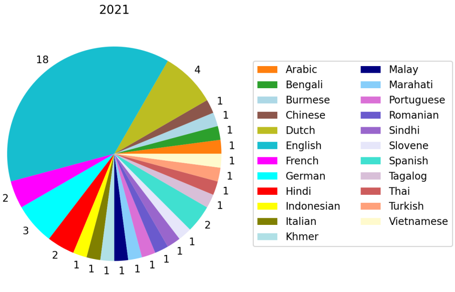
Anna Wierzbicka. 2009. Overcoming anglocentrism in emotion research. *Emotion Review*, 1(1):21–23.

Appendix

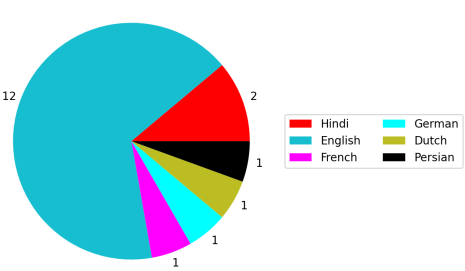
In the following pie charts, the distribution of languages treated in the WASSA editions between 2011 and 2022 are shown. The papers used for obtaining these distributions are the same as the papers used in Figure 2.



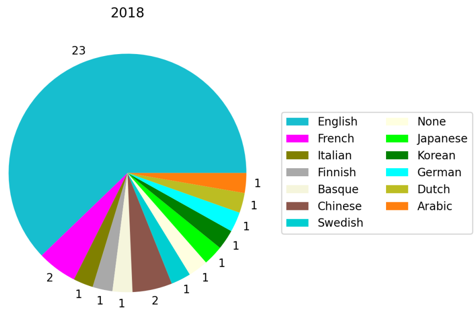
Languages included in the papers at WASSA-2022.



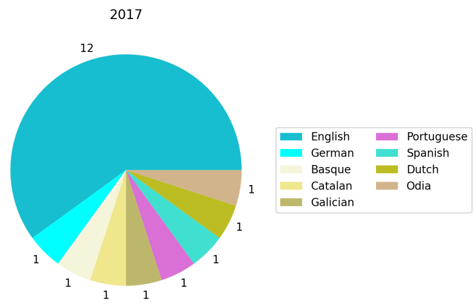
Languages included in the papers at WASSA-2021.



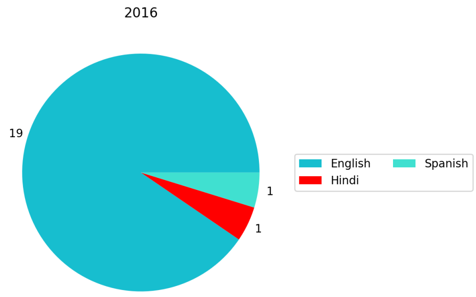
Languages included in the papers at WASSA-2019.



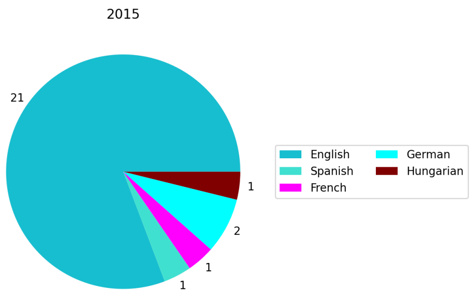
Languages included in the papers at WASSA-2018.



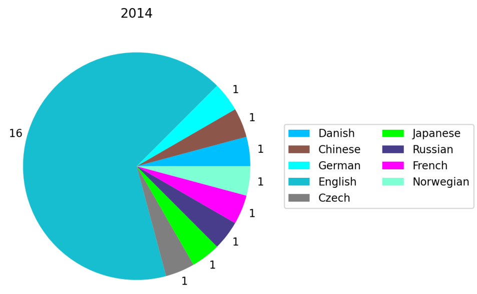
Languages included in the papers at WASSA-2017.



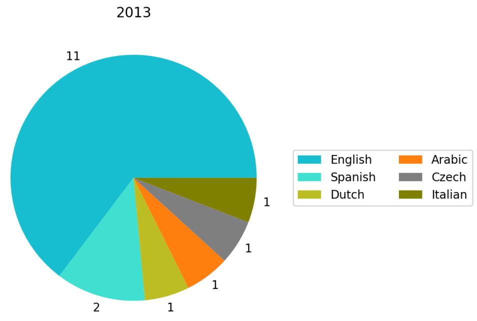
Languages included in the papers at WASSA-2016.



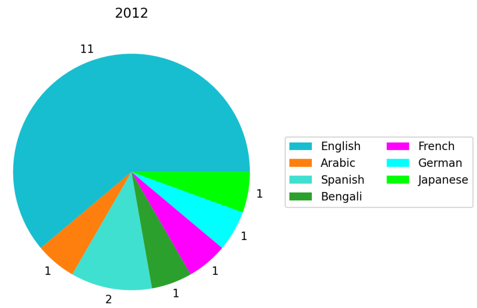
Languages included in the papers at WASSA-2015.



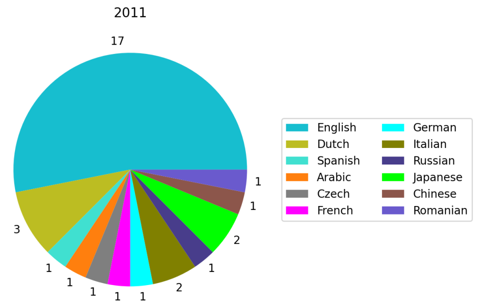
Languages included in the papers at WASSA-2014.



Languages included in the papers at WASSA-2013.



Languages included in the papers at WASSA-2012.



Languages included in the papers at WASSA-2011.

Figure 4: Distribution of languages in the WASSA contribitions between 2011 and 2022.

Sadness and Anxiety Language in Reddit Messages Before and After Quitting a Job

Molly E. Ireland **Micah Iserman** **Kiki Adams**
Receptiviti
`{mireland, miserman, kadams} @receptiviti.com`

Abstract

People globally quit their jobs at high rates during the COVID-19 pandemic, yet there is scant research about emotional trajectories surrounding voluntary resignations before or during that era. To explore long-term emotional language patterns before and after quitting a job, we amassed a Reddit sample of people who indicated resigning on a specific day (n = 7,436), each of whom was paired with a comparison user matched on posting history. After excluding people on the basis of low posting frequency and word count, we analyzed 150.3 million words (53.1% from 5,134 target users who indicated quitting) using SALLEE, a dictionary-based syntax-aware tool, and Linguistic Inquiry and Word Count (LIWC) dictionaries. Based on posts in the year before and after quitting, people who had quit their jobs used more sadness and anxiety language than matched comparison users. Lower rates of "I" pronouns and cognitive processing language were associated with less sadness and anxiety surrounding quitting. Emotional trajectories during and before the pandemic were parallel, though pandemic messages were more negative. The results have relevance for strategic self-distancing as a means of regulating negative emotions around major life changes.

1 Introduction

Leaving a job is a major crossroads in adult life. Though in hindsight most people view their choice positively (e.g., with respect to improved career opportunities, pay, and job satisfaction; Parker and Horowitz, 2022), voluntarily resigning is nevertheless a stressful and consequential life transition. Like other major life changes, the social turmoil and financial uncertainty that come with quitting a job are universal, but there are better and worse ways to cope with the emotional fallout.

In the last few years, the psychology of quitting has been complicated by what some have called the Great Resignation. In the United States, voluntary resignation rates have been increasing linearly since the mid-2000s, peaking in 2021, the second year of the pandemic (Gittleman, 2022); indeed, the main difference between pre-pandemic and pandemic-era quitting may be increased likelihood of leaving the labor market entirely (Ferguson, 2022). Whether pandemic-era quitting is the culmination of a decades-long trend or a phenomenon triggered by the pandemic is still being debated (Fuller and Kerr, 2022). What is clearer is that people who chose to leave their jobs in the last few years did so against the backdrop of ongoing social, emotional, and economic upheavals related to the COVID-19 pandemic. A large part of the Great Resignation narrative centers on debates about work culture and work-life balance that are occurring on social media platforms like Reddit (Medlar et al., 2022).

Despite plenty of media dialogue on the subject, there is scarce psychological research on the Great Resignation specifically or the emotions involved in quitting more broadly. Research on quitting or voluntary turnover has historically focused more on antecedents than a broader window considering individuals' psychology before and following resignation (Rubenstein et al., 2018). Earlier research found that quitting results from low or declining job satisfaction (Chen et al., 2011), burnout related to overwork or high-stress responsibilities at work, a lack of work-life balance (especially difficulty disengaging from work when away from the workplace; Sonnentag and Bayer, 2005), and more attractive job opportunities (Rainayee, 2013). Pandemic-era quitting seems to follow a similar course, though burnout related to employers who are perceived as exploitative and COVID-related burnout have become more salient as the pandemic has increased work-related stress for many employees and employers (Jiskrova, 2022).

The following paper introduces the Reddit Job

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 467–478
July 14, 2023 ©2023 Association for Computational Linguistics

Change Corpus, a sample that includes all Reddit submissions and comments from individuals who discussed leaving or being fired from a job on a specific day in addition to the same complete history from matched comparison users (i.e., people with similar posting histories as the target users who had never discussed leaving a job). The present paper focuses on a subsample of users who indicated quitting (voluntarily leaving) their jobs. As an initial illustration of how the corpus can be used to answer questions about emotional language over time, we explored how rates of sadness and anxiety words vary as a function of two baseline individual differences relevant to emotion regulation: self-focus and cognitive processing. Finally, we compared sad and anxious language between pandemic-era and earlier quitting.

2 Background

Pronouns are intimately linked with affective language and emotion regulation. In the context of negative affect, first-person singular pronouns (e.g., *I*, *me*, *my*; sometimes called "I"-words) tend to be more closely linked with avoidant emotions such as fear and sadness (Tackman et al., 2019) than approach emotions such as anger (Simmons et al., 2005). In the view that emotions prepare people for pragmatic actions, avoidant emotions such as disgust, fear, and sadness compel people to withdraw from noxious or harmful stimuli, whereas approach emotions such as desire, joy, and anger impel people to engage with the emotional stimulus (Carver and Harmon-Jones, 2009; Corr, 2013). People who use more first-person singular pronouns in everyday conversations and writing tend to be more vulnerable to stress and more prone to affect regulation disorders, such as depression (Baddeley et al., 2013; Tackman et al., 2019). Shifting from a first-person to more distant perspective seems to be a healthy coping strategy when experiencing negative emotions or recounting distressing events. For example, in the expressive writing paradigm, people benefit more from writing about traumatic events when they shift perspectives rather than remaining fixed in a first-person mindset (Holtzman et al., 2017; Pennebaker and Chung, 2007). Self-distancing research similarly shows that people naturally use less "I" when writing or talking about distressing memories, a strategy that reduces negative emotions in the moment in both naturalistic and experimental studies (Park et al., 2016).

Whether self-focus is a product or cause of distress or mental health conditions such as depression has been rigorously debated in psychology. Early theories on depressive realism and aversive self-focus argued that viewing the world and especially the self realistically was distressing; in that view, anyone, regardless of trait negative affectivity or neuroticism, would be disturbed by heightened self-awareness (Wicklund, 1975). Later research qualified those findings, showing that the aversiveness of self-awareness and the tendency towards self-focus after failure but not success were specific to people predisposed to negative affect regulation conditions such as depression (Greenberg and Pyszczynski, 1986; Pyszczynski et al., 1987). In summary, self-focus appears to be harmful when distressed but not otherwise (Pyszczynski et al., 1987), is correlated with trait negative affectivity (Schwartz-Mette and Rose, 2016; Tackman et al., 2019), and can be strategically decreased to help downregulate negative emotions (Kross and Ayduk, 2011).

Cognitive processing language (e.g., *think*, *know*), like self-focus, is not altogether harmful or helpful but can become risky in the context of stress and negative affect. Talking through thought processes can help make sense of emotionally complex issues (e.g., in expressive writing; Kacewicz et al., 2007), and cognitive reappraisal can be a valuable tool for regulating emotions (Riepenhausen et al., 2022). However, chronically high cognitive processing language in conversations (such as in letters or social media messages)—especially in conjunction with negative emotional language—may reflect the kind of rumination (i.e., repetitive, intrusive, inward-focused negative thoughts; Watkins and Roberts, 2020) that characterizes affect regulation disorders such as depression (Dean and Boyd, 2020; Eichstaedt et al., 2018).

Reddit is an increasingly popular resource for social-behavioral scientists interested in analyzing publicly accessible language use surrounding major life events such as romantic breakups (Seraj et al., 2021), community crises such as the COVID-19 pandemic (Ashokkumar and Pennebaker, 2021), and mental health conditions including depression, anxiety, and suicidality (e.g., Matero et al., 2022; Shing et al., 2018). Though Reddit's active user base continues to skew young, American, and male, it is more diverse in terms of ethnic backgrounds, nationalities, and age than typical convenience samples in psychology, such as undergraduate students (Henrich et al., 2010; Sattelberg, 2021). More importantly for collecting conversations about risky and distressing topics, such as ending a relationship or quitting a job, Reddit usernames are typically anonymous, which enables people to discuss negative experiences frankly with few concerns about social or legal risks. Social media analyses also facilitate real-time tracking of changes in social movements. For example, topic analyses of Reddit messages showed that conversations about leaving one's job became more focused on mental health and negative experiences at work after the start of the pandemic (del Rio-Chanona et al., 2022).

Like many social media platforms, Reddit's popularity increased globally during the COVID-19 pandemic. Reddit has also become a hub for social movements related to what's become known as the Great Resignation (e.g., r/antiwork; Medlar et al., 2022). Though, to many, mass resignations appeared to be a zeitgeist triggered by the socioeconomic conditions of COVID, employment data suggest that COVID-era resignations are not unique but are a continuation of linearly increasing voluntary turnover rates dating back to at least a decade before COVID (Gittleman, 2022). Thus, in addition to the primary aim of exploring negative emotional language before and after quitting, a secondary goal of this project was to examine whether negative emotion trajectories differed as a function of quitting era (pre-pandemic or during the COVID-19 pandemic).

3 Method

The sections below first summarize the methods used to assemble the Reddit Job Change Corpus and then discuss the narrower subsample that we focused on in the present analyses. Last, language measures and analytic strategies are described.

3.1 Dataset

In an early phase of this project, we used pushshift.io (which had a searchable archive of Reddit data that is no longer available) to identify a set of users who may have quit their job in two steps: First, we searched for submissions or comments with the query *quit job*, then refined those results by searching for variants of the phrase *I just quit my job*. Second, we collected all submissions and comments from authors in the refined results from between January 2015 and July 2022. This resulted in a sample of 11,391 unique users.

In the current phase of this project (between October 2022 and April 2023), we used the original sample to collect new data, including (a) the full submission and commenting history from 8,797 users with active accounts from the original sample, and (b) any users who were recently active (author of or commenter on up to 102 "hot" submissions at the time of collection) in any of the 1,200 most common subreddits within the original sample. This resulted in a sample of 1,389,763 unique users, which constituted a pool of possible target users (those who may have quit their job) and potential comparison users.

This research analyzes only publicly observable behavior and thus qualifies as exempt under the revised Common Rule in the United States' Federal Policy for the Protection of Human Subjects (Department of Health and Human Services, 2017). In compliance with the Reddit API terms of use, all data analyzed in this research are publicly available and will not be used for commercial purposes.

Target Sample From the new pool of users, we searched for people who may have recently quit their job in two rounds: In the first, we loosely searched for messages (submission title plus body, or comment body) that included (a) a job-related word, such as *job* or *boss*, and (b) a word relating to either quitting or being fired (e.g., *quit*, *resigned*, *fired*, *furloughed*), or (c) a phrase such as *lost my job* or *let me go*. To match, phrases could stand on their own, but quit terms had to be preceded within a sentence by *i*, and fired terms had to be preceded by *i* and *got*, *was*, *was given*, or *have been*. This resulted in a set of 485,005 messages from 271,839 users. In the second round of searching, we lightly cleaned matched messages to remove curly quotes and HTML, then processed them with a dependency tagger (Wijffels, 2023). Once parsed, we used a simple set of dependency-based rules to refine target messages: Each message had to have a self reference (exclusively *i*, *me*, or *my*), job reference (exclusively *job*, *career*, or *position*), and target reference (associated with quitting or being fired). If a message contained all required references, a series of dependency-chain checks were applied to attempt to ensure that the author was talking about their own job, and the target reference applied to that job. If a message passed all dependency checks, it was considered a target message but was additionally checked for hypothetical references (such as *if*, *should*, or *might*) or quotations, which would mark the target message as hypothetical or quoted. See the code for specific checks and criteria: osf.io/p2rt7.

After a refined set of target messages was defined, time references were searched for in the extracted target phrase. If the target phrase included a day reference (such as *yesterday* or *on Monday*), the target sentence included a reference to *minutes*, *hours*, or *days* followed by *ago*, or the target message had no specific time reference but included *just* or *recently*, the message was considered recent. To develop and manually spot-check these criteria, we extracted sentences from target messages along with target phrases and time references: osf.io/xahrc.

The final set of target users were those with messages assigned a target type (quit or fired), not marked as hypothetical or quoted, and marked as recent, and that were not posted in subreddits with names containing the words *meme*, *joke*, *funny*, or *humor*. These criteria resulted in a set of 7,436 users, of whom, 5,357 had only quit messages, 2,016 had only fired messages, and 63 had both.

Comparison Sample To construct a comparison sample, we first removed any users who (a) appeared in the first round of target message identification [i.e., users with any message containing terms loosely relating to a job, and quitting or firing], or (b) made any submissions or comments in subreddits appearing in the second round of target message identification more than once, that also contained the words *work*, *job*, or *career* (such as *r/antiwork* and *r/byebyejob*; which included 71 subreddits). This left 830,960 users to make up the possible comparison pool. To find comparison users, each target user was compared with each user in the comparison pool. The similarity between each user was calculated from inverse Canberra distance between three sets of features: (1) Counts of messages per subreddit [$count_s$; submissions or comments] in which the target user had any messages [$subreddits_t$, where t is the target user; Equation 1], (2) counts of comments and submissions separately across all subreddits [replacing $subreddits_t$ with $\{comments, submissions\}$ for each user in Equation 1], and (3) counts of characters within comments and submissions separately across all subreddits [replacing $subreddits_t$ with $\{nchar(comments), nchar(submissions)\}$ for each user in Equation 1].

$$sim_c = \frac{\sum_{s \in subreddits_t} 1 - \frac{|count_{sc} - count_{st}|}{count_{sc} + count_{st}}}{len(subreddits_t)} \quad (1)$$

These were combined into a weighted average, with subreddit similarity getting 50% weight, message count similarity 30% weight, and message length similarity 20% weight (which helped adjust for differences in similarity distributions between each feature set). These were further weighted by difference in time of first activity (where only users within 1% quantile of the target user were considered) and availability (so each comparison user was only assigned to one target user). After weighting, the comparison user with the highest similarity score was assigned to the given target user.

The comparison users had no target message, so we defined target messages using the paired target user's target message(s): We calculated the position of the target user's target message(s) within their complete message history, then assigned the same target type to the nearest message(s) in similar positions within the comparison user's history. In this way, each comparison user has the same number and type of target message(s).

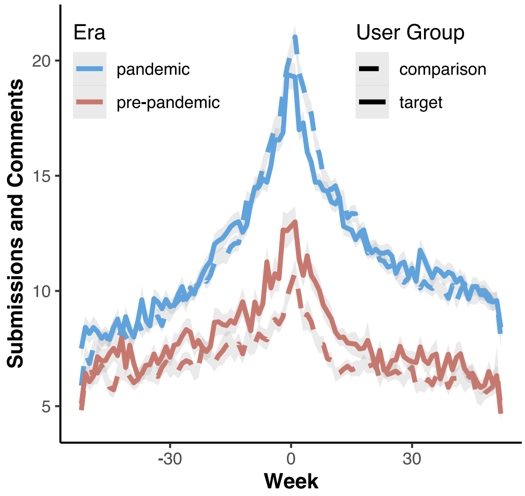


Figure 1: Average message count per week between user groups and eras.

Figure 1 shows the average number of messages (submissions or comments) across users within each week, between the target and comparison samples and between eras. The method of selecting comparison users and assigning them target messages ensured that these distributions would look similar between samples. One remaining difference is the slightly lower number of pre-pandemic messages from comparison users, which is due to the way users were initially sampled: The target sample is more informed by the initial seed sample of users, which includes many users who are no longer active in the most common subreddits within that sample, whereas the comparison sample is primarily made up of users who were recently active at the time of collection.

The fact that there are fewer pre-pandemic messages overall may make comparisons between eras more challenging, as anything that varies by number of messages will also appear to vary by era. A broader challenge comes from the general distribution of messages: All users in this sample must have a 0th week, but they can freely vary in how many surrounding weeks they have. This results in the tent-like distribution of messages around 0, which may make it difficult to identify trends in messages over time, as anything that varies by number of messages will appear to have a strong time association as well.

The scripts used to collect and prepare these samples, along with raw and scored versions of the resulting datasets are available on the Open Science Framework: osf.io/gxbts.

3.2 SALLEE and LIWC

We chose to use SALLEE (Syntax-Aware LexicaL Emotion Engine; Adams, 2022) for measuring emotions. Beyond measuring overall sentiment (i.e., positive or negative emotional tone), SALLEE provides measures of granular emotions that underlie sentiment, such as *fear*, *excitement*, and *gratitude*. SALLEE's lists of emotion words are derived in part from LIWC's affect categories but are applied in a syntax-aware architecture which helps it to perform well on short texts, such as many found on Reddit, and behave reliably in noisy and diverse linguistic contexts. The syntax-aware architecture includes provisions for structures such as intensifiers (e.g., *very*, *so much*), softeners (e.g., *kinda*, *a bit*), negations (e.g., *not*, *never*), punctuation, capitalization, idioms, and words that express emotion and sentiment flexibly depending on context (e.g., swear words). For example, the phrases *pretty darned happy*, *not really happy*, and *NOT happy!* would be scored as moderately positive, somewhat negative, and very negative, respectively. SALLEE outputs a weighted percentage for fourteen specific emotions, three sentiment valences (*goodfeel*, *badfeel*, *ambifeel*), a combined sentiment score reflecting *goodfeel* minus *badfeel*, and combined *emotionality* and *non-emotion* scores.

SALLEE's design makes it particularly valuable for use outside the field of computer science. While more complex models can also offer syntax awareness and a rich array of emotions as output features, they typically do not offer transparency or explainability, do not perform well on casual language, or do not perform well on short texts. Like many other researchers in fields such as sociology, psychology, sociolinguistics, and communication, we found the ability to dissect and explain the way that our data was scored to be vital.

We additionally used the Linguistic Inquiry and Word Count (LIWC; Pennebaker et al., 2015) anxiety category to measure anxiety; although SALLEE has a fear category that captures worry, anxiety, and fright broadly, it does not have an anxiety-specific category. LIWC is SALLEE's conceptual progenitor and the most commonly used dictionary-based emotional language measure in use in the social-behavioral sciences today (see Eichstaedt et al., 2021). LIWC has fewer words in each of its emotion dictionaries and does not use syntactic context to qualify the weight of individual words, yet it performs well across many social contexts and modalities (Boyd and Schwartz, 2021; Vine et al., 2020). For both tools, we focused on negative emotion words (LIWC anxiety and SALLEE sadness) as the outcomes and first-person singular pronouns ("I"-words) and cognitive processing language, both from LIWC-2015, as moderators.

3.3 Analytic Strategies

Our aims were to explore (1) how individual differences in self-focus (based on first-person singular pronouns, e.g., *I*, *me*, *my*) and cognitive language (based on cognitive words, e.g., *idea*, *think*, *realize*) relate to anxiety and sadness language in social media messages before and after quitting, and (2) how the long-term emotional trajectories associated with quitting a job compare between pre-pandemic and pandemic-era resignations. We focused specifically on sadness and anxiety words as relatively common avoidant emotions that may present barriers to actively coping with major life stressors.

To address these questions, we first concatenated messages by week and analyzed weeks with SALLEE and LIWC (rather than scoring and averaging across individual messages, most of which were too brief for traditional dictionary-based text analysis). Weeks contained about 458 words total on average (*median* = 223) and were highly variable across users (*SD* = 790).

We measured emotional trajectories (linear and quadratic) surrounding quitting as a function of linguistic moderators ("I" pronouns and cognitive processing language), quitting era (resigning before or during the COVID-19 pandemic), and user set (target users who wrote about quitting or comparison users) using linear mixed effects models in R (Bates et al., 2015; R Core Team, 2023). All models included random intercepts for authors nested within dyads (target users and matched pairs). For test statistics, we report *F* with Kenward-Roger approximated degrees of freedom (Halekoh and Højsgaard, 2014).

Though the full corpus covers individuals' entire submission and comment histories, in some cases for several years, we focused on posts within 52 weeks before or after users' quit messages. We excluded the week centered on a quit message (the quit week) from the dataset used for visualization and statistical tests in order to focus on how people communicate in general—outside of submissions or comments specifically about resigning—and avoid artifacts related to atypical quit weeks at the center of the distribution. Weeks with outliers (>3.5 standard deviations from the mean) for language variables (anxiety, sadness, "I" pronouns, and cognitive processing language) were removed (2.7% of rows from the original sample); we also removed users who were unmatched (lacking a comparison or target users; *n* = 277) after the word count and posting frequency exclusions.

4 Results

We first regressed emotional language on user set (target versus comparison users) and week in a main effects model. Individuals who discussed quitting their jobs on Reddit used more sadness [*F*(1, 3966.2) = 32.93] and anxiety language [*F*(1, 3935.2) = 261.1] than did matched comparisons across the 2-year time span, and quadratic effects were the best fit for both anxiety [*F*(2, 289357) = 32.22] and sadness [*F*(2, 290720.1) = 51.7], all *p* < .0001.

4.1 First-Person Singular Pronouns

In main effects-only models including user set and week as covariates, baseline "I" rates correlated with higher rates of sadness language, *F*(1, 8410.4) = 55.50, and anxiety words, *F*(1, 8369.3) = 501.1.

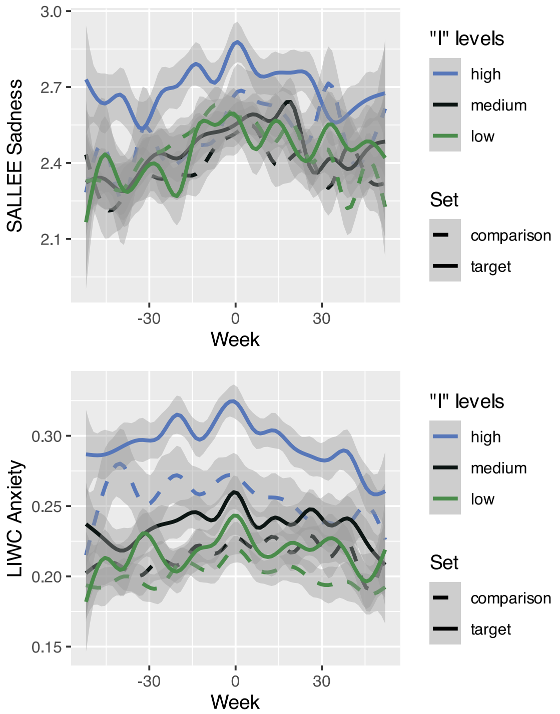


Figure 2: Sadness and anxiety language (% of total words; unweighted for LIWC, weighted for SALLEE) by baseline "I" pronouns. Error bands show 95% confidence intervals. Quit weeks are omitted.

For the moderator models, we were most interested in interactions with user set, indicating whether moderation by "I" differed between people who discussed quitting on Reddit and comparison users. The two-way interaction between "I"-words and user set, controlling for week as a covariate, was significant for both sadness and anxiety language, both *F* > 31, both *p* < .0001. Figure 2 suggests that the largest differences between target and comparison users occurred before quitting for sadness and after quitting for anxiety.

Simple effect models showed that target users who used high rates of first-person singular pronouns at baseline used more sadness language than comparison users, *F*(1, 1905.1) = 28.46; user set effects were nonsignificant for moderate and low baseline "I"-word usage, both *F* < 1. Users who reported quitting their jobs used more overall anxiety language, relative to comparison users, at all levels of baseline "I"-words, though effects were strongest for people using the most baseline "I" [High *F*(1, 1697.90) = 55.53, Medium *F*(1, 1782.12) = 32.48, Low *F*(1, 1768.8) = 18.02].

The differences between comparison and target users' anxiety and sadness language appeared to

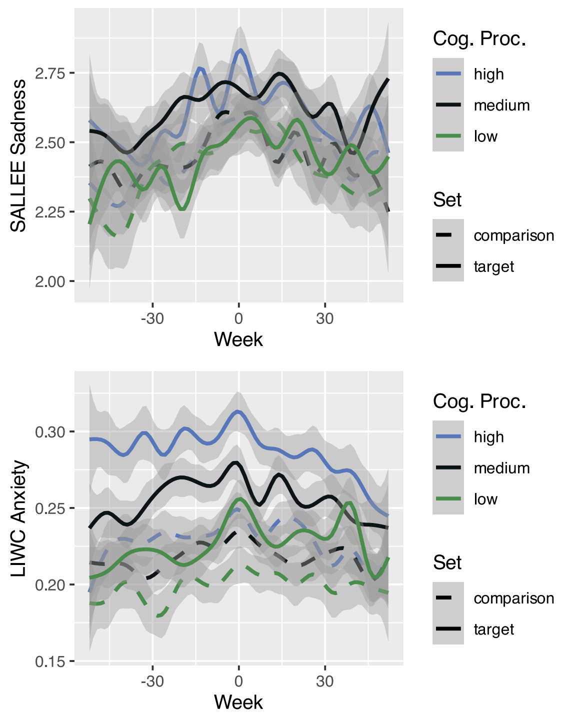


Figure 3: Sadness and anxiety language by baseline cognitive processing language. Error bands show 95% confidence intervals. Quit weeks are omitted.

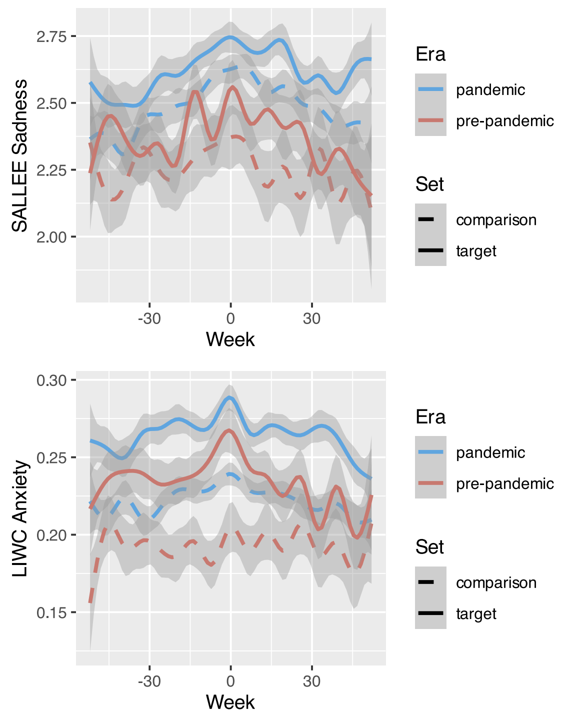


Figure 4: Sadness and anxiety language as a function of era. Error bands show 95% confidence intervals.

be independent of whether they were also talking about work, based on models controlling for work-related language (LIWC's work category, e.g., *boss*, *salary*) as a covariate. Indeed, reading posts from people in the target sample with high baseline "I" and high rates of sadness in the few months before quitting shows that their messages focused primarily on personal grief separate from work (e.g., mourning romantic partners and pets, sadness over family members' serious illnesses). However, anxiety language for high "I" users was more clearly a mix of personal distress and work-related worries, especially after quitting. For example, some messages in the months after quitting reflect the daily life stressors associated with starting a new job (e.g., "My biggest fear is money ... I just started this job so I have no PTO to fall back on.").

4.2 Cognitive Processing

Main effects models including user set and week as covariates showed that baseline cognitive language correlated with higher rates of anxiety [$F(1, 8742.9) = 214.5$] and sadness language [$F(1, 8752.3) = 22.64$] overall, both $p < .0001$.

In the full model regressing anxiety and sadness on the interaction of baseline cognitive processing language, user set, and quadratic effects of week, the strongest effects for both outcome variables were the cognitive processing-by-set interactions. For both, effects were driven by user set effects (i.e., people who quit their jobs using more sadness and anxiety language than comparison users) being greater for those using more baseline cognitive processing language. For anxiety, the effect was significant for all levels [High $F(1, 1735.8) = 146.91$, Moderate $F(1, 1878.1) = 72.79$, low $F(1, 1837.2) = 39.54$, all $p < .0001$]. For sadness, effects were significant only for high and medium cognitive language [High $F(1, 1782.7) = 20.76$, $F(1,1868.8) = 26.199$, Low $F(1,1760.9) = 0.06$]. All models controlled for week as a covariate.

Conclusions were identical when controlling for the degree to which people talked about their job on Reddit by including work-related language as a covariate. Messages with high rates of anxiety written by high baseline cognitive language users tended to focus on general anxiety more than work-related worries. For example, people expressed anxieties about belongings and hobbies ("I recently put a ton of thought into getting either the moto x pure edition ... I am currently using the Galaxy S6 edge, and I swear, I have a panic attack at least once per day worrying about dropping and breaking it") and their own mental health ("So I hope this makes sense. When I am really anxious, [I need] time alone away from people...").

4.3 Quitting Era

Trajectories for sadness and anxiety were roughly parallel between comparison and target groups (Figure 4). There were main effects for pandemic era for both outcomes, indicating that people used more sadness and anxiety language during the pandemic than before regardless of whether they did or did not discuss quitting their jobs on Reddit [Anxiety $F(1, 18892.5) = 85.09$, Sadness $F(1, 19828.1) = 79.92$]. For both emotion variables, there were no significant two- or three-way interactions with user set (all $t < 2$), suggesting that differences between target and comparison users were not limited to the pandemic era. The effect of user set remained significant controlling for era, work-related language, and the quadratic effect of week [Anxiety $F(1, 3998.70 = 313.4$, Sadness $F(1, 4027.1) = 55.53$ both $p < .0001$].

5 Discussion

The results illustrate the potential uses of a new corpus of Reddit messages written by two groups of people: those who indicated on Reddit that they had left their jobs (voluntarily or not) and matched comparison users with similar posting histories who had not discussed a job change. Though the findings are correlational, they have potential relevance for future interventions aimed at helping people cope with career changes and other stressful life events more effectively.

Examining emotional language in the year before and after quitting showed that people who quit their jobs used more anxiety and sadness language than matched comparison users, and these differences were largest for people using high rates of "I" pronouns at baseline. Our results build on self-distancing and expressive writing research to suggest that avoiding self-focus (Kross and Ayduk, 2011) or flexibly regulating perspectives by changing personal pronouns (Seih et al., 2011) may help people experience less distress as they prepare for major life changes.

Anxious language before and in the weeks immediately surrounding quitting was highest for those using high rates of cognitive language at baseline, relative to both comparison users and people with low baseline rates of cognitive language. Words such as *think* and *wonder* reflect self-insight and sense-making in diaries, therapy, or expressive writing (Pennebaker and Chung, 2007) but may be more reflective of hedging (e.g., "I think"), uncertainty (e.g., "I guess"), and rumination in everyday social contexts, such as Reddit (Dean and Boyd, 2020). People with larger negative emotional vocabularies tend to also use more cognitive processing language (Vine et al., 2020). Though our results are preliminary, they support the conjecture that cognitive language may be risky in some social contexts, such as social media.

Together, our findings suggest that predictive models aiming to predict specific life events or linguistic sentiment may produce more accurate or precise results if moderation by pronouns and cognitive language are considered. Exploring possible psychological moderators is especially consequential in transparent models, such as regression or structural equation modeling, where the aim is to model and interpret every feature in depth—understanding each variable's relation to the outcome and other predictors as well as its variance structure—in order to facilitate psychological and behavioral insights (Rudin, 2019).

Methodologically, this project illustrates potential uses of SALLEE, a new sentiment analysis approach used primarily in industry settings (Adams, 2022). SALLEE integrates practical aspects of both traditional dictionary-based emotion measures and syntax-aware techniques that such lexicons typically lack. Like other top-down methods, SALLEE is relatively transparent and face-valid, including only words with explicit emotional content (e.g., *lonely* and *wept* for sadness), in contrast with data-driven approaches that often assign sentiment weights to superficially neutral words. Face validity is not a panacea, and seemingly unambiguous emotion terms may have different implications across different contexts (Chan et al., 2021; Hamilton et al., 2016); however, using explicitly emotional words facilitates straightforward interpretation of results and lowers the adoption barrier for researchers who are new to sentiment analysis. We should note, however, that machine learning or open-vocabulary methods often outperform dictionary approaches in cases with sparse words, novel contexts, and many low-frequency or out-of-vocabulary words (see Eichstaedt et al., 2021).

5.1 Limitations

The sample we collected has limitations shared by most datasets focusing on naturalistic behavior surrounding some event. First, because the starting point in this sample was the quit messages, and not every person posts each week, there are necessarily more target posts than other posts. Quit weeks and weeks immediately surrounding target posts were also more verbose than other weeks and, because work-related concerns were salient at the time, likely included more comments and posts about work or career planning.

Second, the Reddit sample we analyzed is heterogeneous. In most respects, that is a benefit of these data—the conversations covered diverse topics and took place in groups with varying social norms, cohesiveness, and cultures. In that way, these messages are more naturalistic than language from controlled experiments or narrowly focused social media research. Yet there are better options than simply averaging over these differences. For example, emotional expressions are both inflated and suppressed by forum norms regarding emotional self-disclosure (see Balani and De Choudhury, 2015), and the same terms take on different affective meanings across communities (Hamilton et al., 2016). Future research on these or similar data may benefit from clustering forums into psychologically meaningful groups or developing sentiment lexicons tailored to each forum.

Finally, as with any analysis of self-labeled data on social media, we are taking people at their word, accepting the likelihood that some of the messages about quitting in our sample were exaggerated or fabricated (Coppersmith et al., 2015). Despite efforts to stringently filter out hypothetical, satirical, fictional, remembered, or otherwise non-literal references to recent quitting, there are also no doubt some remaining false positives.

5.2 Future Analyses

The corpus we have compiled—including both the messages focused on in the present analyses and those we excluded—is dynamic and growing. In addition to adding new messages as the users in the sample continue to use Reddit, the sample offers a cornucopia of options for studying the psychology of job changes. The sample of excluded users alone is rife for analyses involving sarcasm detection ("Quitting will solve everything!"), advice requests ("What's next if I quit today?"), and counterfactual thinking ("If I'd quit a year ago..."). We are sharing the filtered data as-is but will continue refining it over time. Future analyses may compare SALLEE and LIWC with other language-based emotion measures and experiment with machine learning approaches to forecasting quitting.

The corpus may also be useful for specific workplace applications. Being able to predict voluntary turnover from everyday conversations that are not explicitly about quitting would be invaluable to employers, as organizations lose expertise, social capital, and tangible and intangible investments when employees resign (Rubenstein et al., 2018). However, devising algorithms for predicting the likelihood of leaving a job from language used outside of work introduces ethical quagmires that are beyond the scope of this paper, including questions about the costs of false positives (i.e., being wrongly labeled as a turnover risk by employers).

From the perspective of social-personality psychology, our results add to previous research showing that individual differences in self-focus or self-distancing are relevant to emotional experiences, especially during times of stress or distress (Kross and Ayduk, 2011). We additionally build on the less-established link between cognitive processing and negative emotion (Vine et al., 2020), showing that, independent of self-focus, people who tend to use words referring to thought processes (*think*, *realize*, *wonder*, etc.) at high rates when posting on Reddit use more negative emotional language.

6 Conclusion

Analyses of naturalistic language used in messages on Reddit in the year before and after voluntarily leaving a job showed that people who used the most self-references and cognitive processing language at baseline used more sad and anxious language in the months surrounding quitting. Consistent with research on self-distancing and rumination, low rates of self-referential pronouns and cognitive processing language may be part of a broader pattern of healthy coping with stress and negative emotion. Finally, emotional trajectories for quitting before and after the start of the COVID-19 pandemic were parallel, but pandemic messages were more negative overall. Beyond the psychological implications of this research, methodologically, we have contributed a new publicly-available Reddit corpus and a reliable method for identifying the timing of major life events discussed on social media.

Acknowledgements

Special thanks to Sean Farrell for collecting an initial version of the dataset and generating research ideas during early stages of this project.

References

Kiki Adams. 2022. SALLEE documentation. Online Receptiviti Inc. documentation.

Ashwini Ashokkumar and James W Pennebaker. 2021. Social media conversations reveal large psychological shifts caused by covid-19's onset across us cities. *Science advances*, 7(39):eabg7843.

Jenna L Baddeley, James W Pennebaker, and Christopher G Beevers. 2013. Everyday social behavior during a major depressive episode. *Social Psychological and Personality Science*, 4(4):445–452.

Sairam Balani and Munmun De Choudhury. 2015. Detecting and characterizing mental health related self-disclosure in social media. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, pages 1373–1378.

Douglas Bates, Martin Mächler, Ben Bolker, and Steve Walker. 2015. Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, 67(1):1–48. R package version 1.1.33.

Ryan L Boyd and H Andrew Schwartz. 2021. Natural language analysis and the psychology of verbal behavior: The past, present, and future states of the field. *Journal of Language and Social Psychology*, 40(1):21–41.

Charles S Carver and Eddie Harmon-Jones. 2009. Anger is an approach-related affect: evidence and implications. *Psychological bulletin*, 135(2):183.

Chung-hong Chan, Joseph Bajjalieh, Loretta Auvil, Hartmut Wessler, Scott Althaus, Kasper Welbers, Wouter Van Atteveldt, and Marc Jungblut. 2021. Four best practices for measuring news sentiment using 'off-the-shelf'dictionaries: A large-scale p-hacking experiment. *Computational Communication Research*, 3(1):1–27.

Gilad Chen, Robert E Ployhart, Helena Cooper Thomas, Neil Anderson, and Paul D Bliese. 2011. The power of momentum: A new model of dynamic relationships between job satisfaction change and turnover intentions. *Academy of Management Journal*, 54(1):159–181.

Glen Coppersmith, Mark Dredze, Craig Harman, and Kristy Hollingshead. 2015. From adhd to sad: Analyzing the language of mental health on twitter through self-reported diagnoses. In *Proceedings of the 2nd workshop on computational linguistics and clinical psychology: from linguistic signal to clinical reality*, pages 1–10.

Philip J Corr. 2013. Approach and avoidance behaviour: Multiple systems and their interactions. *Emotion Review*, 5(3):285–290.

Hannah J Dean and Ryan L Boyd. 2020. Deep into that darkness peering: a computational analysis of the role of depression in edgar allan poe's life and death. *Journal of affective disorders*, 266:482–491.

R Maria del Rio-Chanona, Alejandro Hermida-Carrillo, Melody Sepahpour-Fard, Luning Sun, Renata Topinkova, and Ljubica Nedelkoska. 2022. Mental health concerns prelude the great resignation: Evidence from social media. *arXiv preprint arXiv:2208.07926*.

Department of Health and Human Services. 2017. Federal Policy for the Protection of Human Subjects ('Common Rule'). [revised in 2018 and 2019; became effective on July 19, 2018 and delayed until January 21, 2019, and then again delayed until January 21, 2020].

Johannes C Eichstaedt, Margaret L Kern, David B Yaden, H Andrew Schwartz, Salvatore Giorgi, Gregory Park, Courtney A Hagan, Victoria A Tobolsky, Laura K Smith, Anneke Buffone, et al. 2021. Closed- and open-vocabulary approaches to text analysis: A review, quantitative comparison, and recommendations. *Psychological Methods*, 26(4):398.

Johannes C Eichstaedt, Robert J Smith, Raina M Merchant, Lyle H Ungar, Patrick Crutchley, Daniel Preoţiuc-Pietro, David A Asch, and H Andrew Schwartz. 2018. Facebook language predicts depression in medical records. *Proceedings of the National Academy of Sciences*, 115(44):11203–11208.

Stephanie Ferguson. 2022. Understanding america's labor shortage: The most impacted industries. *US Chamber of Commerce: Washington, DC, USA*, 7.

Joseph Fuller and William Kerr. 2022. The great resignation didn't start with the pandemic. *Harvard Business Review*.

Maury Gittleman. 2022. The "great resignation" in perspective. *Monthly Labor Review*.

Jeff Greenberg and Tom Pyszczynski. 1986. Persistent high self-focus after failure and low self-focus after success: The depressive self-focusing style. *Journal of Personality and Social Psychology*, 50(5):1039.

Ulrich Halekoh and Søren Højsgaard. 2014. A kenward-roger approximation and parametric bootstrap methods for tests in linear mixed models – the R package pbkrtest. *Journal of Statistical Software*, 59(9):1–30. R package version 0.5.2.

William L Hamilton, Kevin Clark, Jure Leskovec, and Dan Jurafsky. 2016. Inducing domain-specific sentiment lexicons from unlabeled corpora. In *Proceedings of the conference on empirical methods in natural language processing. conference on empirical methods in natural language processing*, volume 2016, page 595. NIH Public Access.

Joseph Henrich, Steven J Heine, and Ara Norenzayan. 2010. The weirdest people in the world? *Behavioral and brain sciences*, 33(2-3):61–83.

Nicholas S Holtzman et al. 2017. A meta-analysis of correlations between depression and first person singular pronoun use. *Journal of Research in Personality*, 68:63–68.

Gabriela Ksinan Jiskrova. 2022. Impact of covid-19 pandemic on the workforce: from psychological distress to the great resignation. *J Epidemiol Community Health*, 76(6):525–526.

Ewa Kacewicz, Richard B Slatcher, and James W Pennebaker. 2007. Expressive writing: An alternative to traditional methods. *Low-cost approaches to promote physical and mental health: Theory, research, and practice*, pages 271–284.

Ethan Kross and Ozlem Ayduk. 2011. Making meaning out of negative experiences by self-distancing. *Current directions in psychological science*, 20(3):187–191.

Matthew Matero, Albert Hung, and H Schwartz. 2022. Understanding BERT's mood: The role of contextual-embeddings as user-representations for depression assessment. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 89–94.

Alan Medlar, Yang Liu, and Dorota Glowacka. 2022. Nobody wants to work anymore: An analysis of r/antiwork and the interplay between social and mainstream media during the great resignation. *arXiv preprint arXiv:2210.07796*.

Jiyoung Park, Özlem Ayduk, and Ethan Kross. 2016. Stepping back to move forward: Expressive writing promotes self-distancing. *Emotion*, 16(3):349.

Kim Parker and Juliana Menasce Horowitz. 2022. Majority of workers who quit a job in 2021 cite low pay, no opportunities for advancement, feeling disrespected. *Pew Research Center*.

James W Pennebaker and Cindy K Chung. 2007. Expressive writing, emotional upheavals, and health. In *Foundations of health psychology*, page 263–284. Oxford University Press.

JW Pennebaker, RJ Booth, RL Boyd, and ME Francis. 2015. Linguistic inquiry and word count: LIWC 2015.

Tom Pyszczynski, Kathleen Holt, and Jeff Greenberg. 1987. Depression, self-focused attention, and expectancies for positive and negative future life events for self and others. *Journal of personality and social psychology*, 52(5):994.

R Core Team. 2023. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. Version 4.3.0.

Riyaz Ahmad Rainayee. 2013. Employee turnover intentions: Job stress or perceived alternative external opportunities. *Business and Management*, 5(1):48–59.

Antje Riepenhausen, Carolin Wackerhagen, Zala C Reppmann, Hans-Christian Deter, Raffael Kalisch, Ilya M Veer, and Henrik Walter. 2022. Positive cognitive reappraisal in stress resilience, mental health, and well-being: A comprehensive systematic review. *Emotion Review*, 14(4):310–331.

Alex L Rubenstein, Marion B Eberly, Thomas W Lee, and Terence R Mitchell. 2018. Surveying the forest: A meta-analysis, moderator investigation, and future-oriented discussion of the antecedents of voluntary employee turnover. *Personnel Psychology*, 71(1):23–65.

Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature machine intelligence*, 1(5):206–215.

William Sattelberg. 2021. The demographics of reddit: Who uses the site. *Alphr*.

Rebecca A Schwartz-Mette and Amanda J Rose. 2016. Depressive symptoms and conversational self-focus in adolescents' friendships. *Journal of abnormal child psychology*, 44:87–100.

Yi-Tai Seih, Cindy K Chung, and James W Pennebaker. 2011. Experimental manipulations of perspective taking and perspective switching in expressive writing. *Cognition & emotion*, 25(5):926–938.

Sarah Seraj, Kate G Blackburn, and James W Pennebaker. 2021. Language left behind on social media exposes the emotional and cognitive costs of a romantic breakup. *Proceedings of the National Academy of Sciences*, 118(7):e2017154118.

Han-Chin Shing, Suraj Nair, Ayah Zirikly, Meir Friedenberg, Hal Daumé III, and Philip Resnik. 2018. Expert, crowdsourced, and machine assessment of suicide risk via online postings. In *Proceedings of the fifth workshop on computational linguistics and clinical psychology: from keyboard to clinic*, pages 25–36.

Rachel A Simmons, Peter C Gordon, and Dianne L Chambless. 2005. Pronouns in marital interaction: What do "you" and "i" say about marital health? *Psychological science*, 16(12):932–936.

Sabine Sonnentag and Ute-Vera Bayer. 2005. Switching off mentally: predictors and consequences of psychological detachment from work during off-job time. *Journal of occupational health psychology*, 10(4):393.

Allison M Tackman, David A Sbarra, Angela L Carey, M Brent Donnellan, Andrea B Horn, Nicholas S Holtzman, To'Meisha S Edwards, James W Pennebaker, and Matthias R Mehl. 2019. Depression, negative emotionality, and self-referential language: A multi-lab, multi-measure, and multi-language-task research synthesis. *Journal of personality and social psychology*, 116(5):817.

Vera Vine, Ryan L Boyd, and James W Pennebaker. 2020. Natural emotion vocabularies as windows on distress and well-being. *Nature communications*, 11(1):4525.

Edward R Watkins and Henrietta Roberts. 2020. Reflecting on rumination: Consequences, causes, mechanisms and treatment of rumination. *Behaviour Research and Therapy*, 127:103573.

Robert A Wicklund. 1975. Objective self-awareness. In *Advances in experimental social psychology*, volume 8, pages 233–275. Elsevier.

Jan Wijffels. 2023. *udpipe: Tokenization, Parts of Speech Tagging, Lemmatization and Dependency Parsing with the 'UDPipe' 'NLP' Toolkit*. R package version 0.8.11.

Communicating Climate Change: A Comparison Between Tweets and Speeches by German Members of Parliament

Robin Schaefer[1], **Christoph M. Abels**[2], **Stephan Lewandowsky**[2,3,4], **Manfred Stede**[1]
[1] Applied Computational Linguistics, University of Potsdam, Potsdam, Germany
[2] Department of Psychology, University of Potsdam, Potsdam, Germany
[3] School of Psychological Science, University of Bristol, Bristol, UK
[4] School of Psychological Science, University of Western Australia, Perth, Australia
`{robin.schaefer|abels|stede}@uni-potsdam.de`
`stephan.lewandowsky@bristol.ac.uk`

Abstract

Twitter and parliamentary speeches are very different communication channels, but many members of parliament (MPs) make use of both. Focusing on the topic of climate change, we undertake a comparative analysis of speeches and tweets uttered by MPs in Germany in a recent six-year period. By keyword/hashtag analyses and topic modeling, we find substantial differences along party lines, with left-leaning parties discussing climate change through a crisis frame, while liberal and conservative parties try to address climate change through the lens of climate-friendly technology and practices. Only the AfD denies the need to adopt climate change mitigating measures, demeaning those concerned about a deteriorating climate as climate cult or fanatics. Our analysis reveals that climate change communication does not differ substantially between Twitter and parliamentary speeches, but across the political spectrum.

1 Introduction

Anthropogenic climate change (CC) is one of the major issues that societies are facing today. This is reflected in the vast amount of CC-related discourse that is distributed via different forms of media including, for instance, traditional newspapers and various social media. By their nature, platforms like Twitter are available to professional users and to the general public alike. In our work, we focus on members of the German parliament (MPs) as one specific group of actors that engages in CC discourse. More precisely, we are interested in their communication on Twitter on the one hand, and their parliamentary speeches on the other.

While Twitter is a more recent development, parliamentary speeches are a traditional means for politicians to convey their ideas and positions. Obviously, these domains imply different audiences: Tweets are aimed at the general public (and thus to potential voters), while speeches are first of all directed at other MPs, and only indirectly meant to – sometimes – reach a broader audience.

This situation prompts us to compare the communications of MPs in the two arenas: Is CC equally prominent as a topic? Are the same aspects of CC discussed? How does this differ between parties? – We take the German Bundestag, and especially a six-year period starting in 2016, as the basis for our study.

The paper is structured as follows: After discussing related work, we explain our data collection and present corpus statistics in Section 3. Then, in Section 4 we provide analyses of term and hashtag use, while Section 5 conducts a topic modeling experiment. The findings are discussed in Section 6, and we conclude the paper in Section 7.

2 Background and Related Work

Social media has become an essential tool of political communication. Research includes the role of social media in political campaigning (e.g., Bright et al., 2020), voting (e.g., Jacobs and Spierings, 2016), popularity of politicians (e.g., Van Aelst et al., 2017), disinformation campaigns (Keller et al., 2020), as well as elite communication with citizens (e.g., Munger et al., 2019).

Among various currently popular social platforms, Twitter has become a major player in political communication (Freelon and Karpf, 2015; Graham et al., 2013). In Germany, the 2009 Bundestag elections became the first instance in which MPs used Twitter on a large-scale (Plotkowiak and Stanoevska-Slabeva, 2013). Since then, Twitter has established itself as a crucial communication channel for many MPs. Yet, social media creates a distinct communication environment, in which salient topics do not necessarily resemble those discussed among a mass audience (Stier et al., 2020). As Stier

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 479–496
July 14, 2023 ©2023 Association for Computational Linguistics

et al. point out, however, when discussing policies, politicians and their followers on Twitter tend to share similar priorities, pointing to a communication approach that is targeted at a like-minded audience (Plotkowiak and Stanoevska-Slabeva, 2013). Recently, this aspect has been investigated from the perspective of news-sharing behavior by MPs in the U.S., the UK, and in Germany (Lasser et al., 2022).

With this growing relevance of Twitter for political communication, the platform has also become a tool for science communication, in particular in respect to CC. In 2013, the United Nations Intergovernmental Panel on Climate Change (IPCC) released its Fifth Assessment Report (AR5), which was the first report that utilized Twitter as a communication tool (Newman, 2017). Newman found that events such as the IPCC report provide an opportunity to draw attention to CC-related issues on Twitter, allowing actors to strategically communicate with their audience, e.g., by adjusting their language in the interaction with different audiences (Walter et al., 2019).

Given that CC becomes an increasingly pressing issue (Ripple et al., 2021), CC-related communication of political elites also deserves further attention. Recent evidence shows that politicians promoting environmental messages on Twitter are also more likely to vote pro-environmentally (Greenwell and Johnson, 2023).

Besides such approaches in the Social Sciences, the NLP community has recently also given more attention to CC communication (Stede and Patz, 2021). For example, Vaid et al. (2022) detect subtopics as well as stances toward CC on a Twitter dataset. An interesting combination of methods was proposed by Jiang et al. (2017), who use LDA to generate topics containing keywords that represent sentiment targets, and employ SentiWordNet to annotate that part of text with a sentiment value. In this way, they try to distinguish different media for reporting with specific attitudes toward CC topics. Farrell (2019) uses NLP to identify several ten thousand people and philanthropic organizations in the U.S. engaged in disseminating CC misinformation.

Parliamentary speech on CC has recently been tackled with manual analyses of framing, for the European Parliament (Kinski and Servent, 2022) and in comparison to the U.S. (Wendler, 2022). Automatic approaches, on the other hand, to our knowledge are rare; one related long-term study on the development of debates about coal in the Bundestag, using Dynamic Topic Modeling, was presented by Müller-Hansen et al. (2021).

Party Name	Orientation
The Left	Left
The Greens	Ecological
SPD	Social Democratic
FDP	Liberal
CDU/CSU	Conservative
AfD	Right/Populist

Table 1: Parties in Parliament and Political Orientation.

3 Data

In this work, we gather MP communications from two sources: tweets and parliament speeches. We use a subset of a tweet set collected by Lasser et al. (2022) who compared the distribution of low-quality news sources on Twitter by political elites in the U.S., the UK and Germany from January 1st 2016 until March 15th 2022. We apply the same timeframe to our present study that addresses only Germany.

During this time, six parties were represented in parliament[1] – see Table 1 for a list with their political orientation. Our timeframe includes (parts of) the three legislative periods 18 (2013-2017), 19 (2017-2021) and 20 (as of 2021). Importantly, both FDP and AfD were not represented in the 18. parliament. Also, the government changed: While CDU/CSU and SPD formed a government in legislative periods 18 and 19, a new government consisting of SPD, the Greens and FDP was formed in period 20.

3.1 MP Tweets

The German subset of tweet IDs provided by Lasser et al. (2022) consists of all tweets from former and present MPs published in the mentioned timeframe (n=754,233). After collecting the tweets in December 2022 via the Twitter API[2] and removing non-German data, a corpus of 673,786 tweets remained. We filtered this set using keywords related to CC and activist groups (see Appendix A), leading to our final corpus of 30,242 tweets from 635 MPs, which, on average, consist of 32 tokens.

---

[1] Due to lack of data we ignore the SSW, which is a regionalist party of the Danish minority, and independent MP.

[2] Due to Twitter privacy regulations only the tweet IDs could be distributed by Lasser et al. (2022).

We apply minimal preprocessing to the tweets for topic modelling. Social media idiosyncrasies such as links, emojis, mentions and hashtags are removed for that purpose, but they remain in the data for hashtag and climate compound analyses (Section 4).

3.2 MP Speeches

For compiling our corpus of speeches, we used the Open Discourse dataset (Richter et al., 2020), which contains all transcripts of speeches in the German parliament from September 1949 until May 2021 (n=907,644). In order to search for speeches focusing on CC, we applied a slightly modified set of keywords where hashtag keywords have been replaced with their proper form, e.g *fridays for future* instead of *#fridaysforfuture*. As the Open Discourse data set only covers months until May 2021, we manually collected the relevant speech transcripts of the missing months. Our final corpus consists of 5,351 speeches, including speeches of MPs and of ministers (most of whom are also MPs).

First experiments revealed that the transcripts contain substantial noise, i.e., material that is not related to CC. To improve results, we separated the speeches into paragraphs and filtered them with the same set of keywords. For our analyses, we treat the paragraphs as individual documents (n: 13,794; average number of tokens: 69). While paragraphs and tweets are still of rather different lengths (69 vs. 32 tokens, on average), using paragraphs allows for a more meaningful comparison than using the full speeches.

4 Analyses

For determining the relevance of the CC issue, and the stances toward it, we first analyse our data statistically, looking at speech and tweet volumes, frequencies of hashtags, and use of subjective "climateX" noun compounds (e.g., Klimapolitik, Klimaschutz, Klimakrise). We are interested in differences between parties, and between tweets and speeches. We present absolute counts and proportions, where proportions in Section 4.1 are calculated with respect to corpus sizes, while in the other two sections, proportions for hashtags and compounds are calculated with respect to total hashtag/compound counts by party.

4.1 Volume of CC Communication

Both CC corpora show a rather skewed party distribution (see also Appendix 7). Of the 30,242 tweets on CC, 13,509 (45%) were tweeted by the Greens, followed by the FDP (4,057; 13%) and the SPD (4,023; 13%). The smallest amount of tweets has been contributed by the AfD (2,208; 7%).

A different pattern can be observed in the speeches on CC. As the amount of speeches roughly correlates with a party's representation in parliament, the CDU/CSU, who had most seats in legislative periods 18 and 19, held most speeches (1,446; 28%), with the SPD ranking second (1,180, 22%). Note, however, that the size of the parliamentary group can only serve as a proxy. Other factors such as political orientation have an influence, which is arguably why the Greens have a similar proportion of speeches (20%) as the SPD. The least number of speeches were given by the FDP (416; 8%).

4.2 Hashtags

Being a social media convention, hashtags are not used in speeches, and thus we entirely focus on the twitter corpus in this section.[3] Results are given in Table 2. While substantial usage of hashtags is indicated by the absolute counts, proportions appear comparatively low. This is caused by the overall large amount of individual hashtags that are being used.

All parties except the AfD most often use the tag #climateprotection (*#klimaschutz*, 1% (AfD) - 10% (SPD)). The Left and, especially, the Greens make use of the term #climatecrisis (*#klimakrise*, 2% (The Left), 4% (The Greens)), while it is not among the top hashtags of the other parties. In addition, the Left tweets on #climatejustice (*#klimagerechtigkeit*, 2%), on #transporttransition (*#verkehrswende*, 3%), and is the only party mentioning the anti-coal activist movement Ende Gelände (*#endegelaende*, 2%). While mentioning transport as well (2%), the Greens also use #energytransition (*#energiewende*, 2%) and #coalphaseout (*#kohleausstieg*, 2%). Being in the government for the whole timeframe, the social democratic SPD frequently uses #climateprotectionlaw (*#klimaschutzgesetz*, 2%) and #climatechange (*#klimawandel*, 2%), a term that is rather avoided by the Greens (in favor of *crisis*). All the

---

[3] Note that we will subsequently mention the translated hashtags where appropriate in addition to the original German ones.

The Left		The Greens		SPD	
Hashtag	C | P	Hashtag	C | P	Hashtag	C | P
#klimaschutz	351 | .04	#klimaschutz	2,751 | .09	#klimaschutz	624 | .10
#fridaysforfuture	238 | .03	#klimakrise	1276 | .04	#energiewende	261 | .04
#verkehrswende	206 | .03	#verkehrswende	736 | .02	#spd	170 | .03
#klimawandel	168 | .02	#kohleausstieg	608 | .02	#klimaschutzgesetz	144 | .02
#klimagerechtigkeit	148 | .02	#energiewende	497 | .02	#klimawandel	116 | .02
#klimakrise	146 | .02	#klima	429 | .01	#fridaysforfuture	89 | .01
#endegelaende	130 | .02	#fridaysforfuture	388 | .01	#klima	85 | .01
#kohleausstieg	121 | .01	#btw21	334 | .01	#bundestag	82 | .01
#klimastreik	113 | .01	#groko	293 | .01	#kohleausstieg	70 | .01
#dielinke	112 | .01	#laschet	275 | .01	#co2	56 | .01
FDP		CDU/CSU		AfD	
Hashtag	C | P	Hashtag	C | P	Hashtag	C | P
#klimaschutz	608 | .08	#klimaschutz	477 | .08	#afd	602 | .09
#co2	276 | .04	#klima	124 | .02	#energiewende	198 | .03
#energiewende	246 | .03	#co2	100 | .02	#co2	174 | .02
#klima	171 | .02	#klimawandel	87 | .02	#klimahysterie	152 | .02
#emissionshandel	150 | .02	#energiewende	75 | .01	#klima	144 | .02
#klimawandel	115 | .02	#bundestag	57 | .01	#fridaysforfuture	134 | .02
#kohleausstieg	115 | .02	#cdu	57 | .01	#klimawandel	89 | .01
#fdp	105 | .01	#wasserstoff	56 | .01	#klimaschutz	84 | .01
#versorgungssicherheit	101 | .01	#nachhaltigkeit	53 | .01	#grüne	80 | .01
#groko	91 | .01	#grünen	50 | .01	#grünen	76 | .01

Table 2: Hashtag Counts (C) and Proportions (P) By Party.

more left-leaning parties mention #fridaysforfuture (1% (The Greens; SPD) – 3% (The Left)).

The liberal FDP emphasizes economic interests with #energytransition (*#energiewende*, 3%), #emissiontrading (*#emissionshandel*, 2%), #coalphaseout (*#kohleausstieg*, 2%), and #servicesecurity (*#versorgungssicherheit*, 1%). The conservative CDU/CSU is the only party mentioning #hydrogen (*#wasserstoff*, 1%) and #sustainability (*#nachhaltigkeit*, 1%), in addition to #climate (*#klima*, 2%) and #climatechange (*#klimawandel*, 2%). The rightwing AfD uses #afd (9%) as their top hashtag and is the only party using the term #climatehysteria (*#klimahysterie*, 2%). Neither of the more right-leaning parties has Fridays For Future or any other social movement among their top hashtags. In contrast, they often apply #co2 (2% (CDU/CSU; AfD) – 4% (FDP)).

4.3 Stance-Conveying Climate Compounds

For the climate compound analysis we use a slightly modified version of the "climate change discourse glossary", which consists of 249 politically charged German noun compounds starting with *climate (Klima)*, i.e., compounds whose use indicates the author's stance towards the issue.[4] For matching glossary entries against terms, hyphenated and non-hyphenated variants are treated as the same compound. Results are shown in Table 3.

Tweets: We decided to keep the hashtags when matching the tweets, which results in some overlap with the previous analysis. This, however, allows us to show a more complete picture on the usage of climate compounds.[5] In contrast to hashtag use, only a small number of climate compound types are used frequently by the parties, and we concentrate on those here.

---

[4] The glossary of noun-noun compounds is available here: `http://www.klimadiskurs.info`. As one exception to the N-N pattern, we add the N-Adj compound *climate-friendly (klimafreundlich)* to our study, as it also conveys a stance toward CC.

[5] Not all climate compounds used as hashtags are represented in the glossary, e.g., *Klimaschutz* ('climate protection'), a term that neutrally describes actions.

The Left (Tweet)		The Greens (Tweet)		SPD (Tweet)	
Compound	C | P	Compound	C | P	Compound	C | P
klimagerechtigkeit	235 | .44	klimakrise	1,951 | .77	klimafreundlich	109 | .43
klimakrise	208 | .39	klimafreundlich	164 | .06	klimakrise	88 | .35
klimafreundlich	25 | .05	klimagerechtigkeit	157 | .06	klimaleugner	10 | .04
klimaaktivist	18 | .03	klimakanzler	54 | .02	klimakanzler	8 | .03
klimakanzler	7 | .01	klimakanzlerin	37 | .01	klimagerechtigkeit	6 | .02
FDP (Tweet)		CDU/CSU (Tweet)		AfD (Tweet)	
Compound	C | P	Compound	C | P	Compound	C | P
klimafreundlich	75 | .43	klimafreundlich	90 | .52	klimahysterie	193 | .34
klimakrise	27 | .16	klimakrise	23 | .13	klimawahn	70 | .12
klimaaktivist	13 | .08	klimakonsens	11 | .06	klimakrise	30 | .05
klimakonsens	7 | .04	klimaaktivist	10 | .06	klimaaktivist	27 | .05
klimanationalismus	6 | .03	klimakompetenz	7 | .04	klimakult	26 | .05
The Left (Speech)		The Greens (Speech)		SPD (Speech)	
Compound	C | P	Compound	C | P	Compound	C | P
klimakrise	54 | .35	klimakrise	403 | .84	klimafreundlich	68 | .56
klimagerechtigkeit	31 | .20	klimafreundlich	31 | .06	klimakrise	34 | .28
klimafreundlich	30 | .19	klimaleugner	7 | .01	klimakonsens	4 | .03
klimapäckchen	11 | .07	klimazerstörung	6 | .01	klimaleugner	4 | .03
klimaleugner	7 | .04	klimapäcken	5 | .01	klimaaktivist	2 | .02
FDP (Speech)		CDU/CSU (Speech)		AfD (Speech)	
Compound	C | P	Compound	C | P	Compound	C | P
klimafreundlich	11 | .28	klimafreundlich	120 | .79	klimahysterie	35 | .19
klimakonsens	9 | .23	klimakrise	17 | .11	klimareligion	16 | .09
klimanationalismus	4 | .10	klimaleugner	4 | .03	klimarettung	15 | .08
klimakrise	4 | .10	klimakonsens	2 | .01	klimaleugner	14 | .08
klimakanzler	2 | .05	klimakompetenz	1 | .01	klimakrise	13 | .07

Table 3: Compound Counts (C) and Proportions (P) By Party (Top: Tweet Corpus; Bottom: Speech Corpus).

First, *climate crisis* (*Klimakrise*) is used by all parties, though with very different frequency (5% (AfD) - 77% (The Greens)), which contrasts with the respective hashtag which is only used by the the Left and the Greens. *Climate-friendly* (*klimafreundlich*) is applied by all parties except the AfD (5% (The Left) - 52% (CDU/CSU)). Other frequently used compounds include *climate justice* (*Klimagerechtigkeit* by the Left (44%) and the Greens (6%)) and *climate activist* (*Klimaaktivist*). Importantly, the AfD exclusively uses a number of compounds that clearly convey negative judgement on the activist movement, including *climate hysteria* (*Klimahysterie*, 34%), *climate delusion* (*Klimawahn*, 12%), and *climate cult* (*Klimakult*, 5%).

Speeches: Some of the patterns found in tweets re-emerge in the speeches. First, *climate crisis* is used by all parties, with proportions ranging from 7% to 84%, the highest of which is associated with the Greens. Second, *climate-friendly* is mentioned by all parties except the AfD (6% (The Greens) - 79% (CDU/CSU)). Third, *climate justice* is frequently used by the Left (20%), and the AfD often applies *climate hysteria* (19%). Contrasting with the tweet data, however, the FDP makes more notable usage of *climate consensus* (*Klimakonsensus*, 23%), while the AfD uses *climate religion* (*Klimareligion*, 9%), *climate rescue* (*Klimarettung*, 8%), and *climate denier* (*Klimaleugner*, 8%).

	Topic Labels Tweet Corpus
T1	emissions,trading,reduction
T2	railroad,transport infrastructure
T3	dates
T4	climate crisis,package,justice
T5	energy transition,supply
T6	EU, multilateralism
T7	climate protection
T8	politics,police
T9	innovation,technology
	Topic Labels Speech Corpus
T1	CO2 emissions,trading,reduction
T2	investment,finances
T3	EU,security,NATO
T4	dates,goals,climate neutrality
T5	CC,dogma,politics
T6	energy transition,renewables,efficiency
T7	railroad,transport infrastructure
T8	politics,democracy,human rights
T9	climate protection,implementation

Table 4: Topic Labels For Tweet and Speech Corpora.

5 Topic Modeling

For topic modeling we use BERTopic (Grootendorst, 2022), a technique based on transformers (Devlin et al., 2019), clustering and a class-based variation of TF-IDF, i.e. c-TF-IDF. The BERTopic algorithm works as follows:

In a first step, documents are embedded in order to create vector space representations that can be used for semantic clustering. Here, we use Sentence-BERT (SBERT) (Reimers and Gurevych, 2019). Second, the dimensionality of embeddings is reduced using UMAP (McInnes et al., 2020), after which documents are clustered using HDBSCAN (McInnes et al., 2017), a hierarchical version of the original DBSCAN algorithm. This allows for soft clustering as unrelated documents are treated as noise. Third, topic representations are created using c-TF-IDF.

We use the same hyperparameters for both the tweet and speech datasets. We remove frequent words 1) by applying BM-25 weigthing and 2) by taking the square root of the term frequency after normalizing the frequency matrix instead of taking the default term frequency in the c-TF-IDF algorithm. Both options can be set in BERTopic. We use the SBERT model *all-mpnet-base-v2*. The minimum cluster size of HDBSCAN is set to 15. Finally, we apply the Maximal Marginal Relevance algorithm (Carbonell and Goldstein, 1998) to diversify the topic representation, using a mild score of 0.2 (with 0=no diversity, 1=complete diversity). We train topic models on the full tweet and speech corpora and on individual party subsets, resulting in 14 topic models.

In the following, we will describe the topics provided by the trained models. Due to space limitations we will concentrate on the topics that were associated with most documents and were interpretable from a climate (policy) perspective. The topic labels, which were manually derived from the models, can be found in Table 4 for the full tweet and speech corpora, and in Table 5 for individual party subsets. See Appendix C for the actual topic representations.

5.1 Topics in Tweets

The topic model trained on the full tweet corpus obtains a diverse set of CC-related topics (see Table 4; top). The most prominent topic (T1) focuses on emissions and their trading and reduction, followed by a topic about transport infrastructure (T2). Other topics concentrate on climate crisis and justice (T4), energy transition (T5), climate protection (T7) and innovation and technology (T9). Another one deals with the EU and multilateralism (T6).

With respect to the models trained on party subsets, there are several similarities and differences to be observed (see Table 5; top). First, all parties talk about CO2 emissions. However, certain differences can be observed. While the Left mainly discusses emission reduction, the other parties except the AfD also focus on pricing. The liberal FDP mentions emission trading. Second, all parties except the AfD are associated with date topics. These topics are characterized by the mention of several years, which are connected to certain climate goals. Third, the Left, the Greens and the SPD have topics related to transport transition and infrastructure among their most common topics (The Left: T6; The Greens: T2; SPD: T2), while the other parties do not. Fourth, most parties have a topic concentrating on energy transition and renewables[6]. The AfD is the only party mentioning nuclear energy and having their own topic on wind energy. Fifth, both FDP and CDU/CSU have innovation/technology

---

[6] Note that while not among the top topics, the Greens had two topics on energy: 1. energy transition; 2. fossil fuels.

	The Left (Tweet)	The Greens (Tweet)	SPD (Tweet)
T1	politics,parties	climate crisis,adaptation	CO2 emissions,pricing
T2	climate politics,protection	railroad,streets	cities,transport
T3	profit,economy	CO2 emissions,reduction,pricing	dates
T4	CO2 emissions,reduction	dates	climate protection,elites
T5	dates	EU,green(washing)	EU,council presidency
T6	street,city	politics,police	energy transition,renewables
	FDP (Tweet)	CDU/CSU (Tweet)	AfD (Tweet)
T1	CO2 emissions,trading	CO2 emissions,pricing,reduction	CO2 emissions
T2	climate politics,protection	dates	CC,protection,world climate
T3	innovation,technology	innovation,technology	climate hysteria,socialism
T4	dates	economy,feasibility	diesel,car,fuel
T5	energy sources, transition	industry,sector	energy transition,nuclear
T6	economy	energy sources,transition	wind energy
	The Left (Speech)	The Greens (Speech)	SPD (Speech)
T1	climate politics,protection	climate crisis,protection	climate protection,costs
T2	energy,transition, "bürgerenergie"	CO2 emissions,trading	energy transition,renewables
T3	CO2 emissions,car	energy transition,renewables	CO2 emissions,pricing
T4	transport infrastructure	coal committe,phase-out	technology,industry
T5	finances,investment	EU	EU
T6	digitalisation,transformation	dates,goals	money,investment
	FDP (Speech)	CDU/CSU (Speech)	AfD (Speech)
T1	CO2, emissions trading	CO2 emissions,trading	energy sources,transition
T2	energy,transition,politics	EU,Russia	CO2 emissions
T3	-	energy transition,renewables	CC,climate religion
T4	-	railroad,money	CC,climate dogma
T5	-	e-mobility,fuels	money
T6	-	farming	economy,EU

Table 5: Topic Labels By Party (Top: Tweet Corpus; Bottom: Speech Corpus).

topics. Finally, some topics are attached to individual parties. The Greens provide a topic concerning the EU and green(washing). The SPD has a topic mentioning EU and EU council presidency. Industry is a topic exclusive to the CDU/CSU and the AfD is the only party talking about climate hysteria and socialism.

5.2 Topics in Speeches

The topic model trained on the full speech corpus produces similar topics as the one trained on tweet data, e.g emissions trading and reduction (T1), energy transition (T6) and transport infrastructure (T7) (see Topic 4; bottom). However, new topics include investment and finances (T2) and politics, democracy and human rights (T8). Furthermore, the EU topic has a special focus on security and NATO, while the transport infrastructure topic (T7) is less dominant.

The general trend found in the tweet data, can be identified as well in the party subsets of the speech data (see Table 5; bottom, labelled (Speech)). Again, all parties concern themselves with CO2 emissions and/or energy transition, the latter of which, however, appears now to be a more dominant topic for left-leaning parties. Furthermore, the AfD has speech topics revolving around climate religion and dogma, which are semantically similar to climate hysteria.

Still, some minor differences can be found. For instance, transport was represented in the most common tweet topics of the Greens and SPD, which is not the case for the respective speech subsets. Except for the Greens, no topics on years/goals can be found. Finally, the CDU/CSU has stronger focus on e-mobility and transport in general than in the tweet corpus.

6 Discussion

Although political communication can be domain dependent (Stier et al., 2020), our findings indicate a strong overlap between compounds, hashtags, and topics prevalent on social media and in parliamentary debates. Politically left-leaning parties (such as the Left, the Greens, and the SPD) tend to discuss CC in terms of its negative and heterogeneous impact (as indicated by the use of the compound *climate crisis (Klimakrise)* and *climate* *justice (Klimagerechtigkeit)* in both speeches and tweets. Especially the Greens make use of this terminology, thereby connecting to various activist groups (e.g., Fridays For Future, Extinction Rebellion, Last Generation) that use *climate crisis* as a frame to highlight the severe and urgent need to counteract the negative consequences of CC. The Left uses *climate justice* more consistently on Twitter and in parliament, compared to both Greens and SPD. While also using *climate crisis*, the SPD most frequently uses *climate-friendly (klimafreundlich)*, which might be indicative for an approach that tries to balance the urgency of CC with measures that are seen as not too invasive for citizens and the economy. This balanced approach is also reflected in the topics, including cities, transport; climate protection, elites (Twitter); and climate protection, costs; technology, industry; and money, investment (speeches).

In contrast, traditionally more business-friendly parties, such as the FDP and CDU/CSU, use *climate-friendly* to a greater extent, which might indicate a stronger orientation towards economic transformation and technological advancements. This interpretation is supported by the topics for both parties, which include innovation and technology, energy sources and transition, as well as e-mobility and fuels in case of the CDU/CSU.

Whereas these five parties acknowledge the need to address CC, although using different framings, the AfD uses terminology that frames CC and corresponding protective measures as hysterical *(Klimahysterie)*, fanatical *(Klimawahn, Klimareligion)*, or cult-like *(Klimakult)*. A similar pattern is found in the AfD topics, which include climate hysteria and socialism on Twitter and climate religion as well as climate dogma in parliament speeches.

This general pattern is reflected by the hashtags #transporttransition (*#verkehrswende*) and #energytransition (*#energiewende*). While the Greens, the SPD, the FDP and the CDU/CSU frequently use the latter, only the Left and the Greens explicitly mention the transition of transport. This difference also emerges in the topics, where those related to transport infrastructure or railroads are associated with left-leaning parties, while topics related to energy transition and emissions reduction are more broadly distributed. This may be due to transport being framed as a social topic by the Left and the Greens, e.g., with a potential for many people to benefit from lower train ticket prices and improved railroad infrastructure. The FDP and CDU/CSU, however, may interpret transport transition as having negative consequences for the automobile industry, a traditionally strong industry sector in Germany, which is why the CDU/CSU discusses e-mobility and alternative fuels in parliament and uses *#hydrogen* (*#wasserstoff*). In contrast, the energy transition is seen as having a strong potential for economy and technological innovation, while being at the same time crucial for emissions reduction. This makes it a more suitable topic to be adopted by parties of different orientations. In sum, compounds, hashtags, and topics reflect the respective parties' political orientation and their stance towards CC and climate protection, with comparatively small differences between parliamentary speeches and communication on Twitter.

7 Conclusion

We studied the communication of German MPs on Twitter and in parliamentary speeches, concentrating on the topic of CC, for which we created corpora that cover the same 6-year period. By topic modeling, hashtag analysis and comparing the use of stance-marked noun compounds, we looked for differences between parties on the one hand, and between the communication in the two channels, on the other hand.

Our analysis reveals, that differences in climate communication primarily emerge across the political spectrum, with left-leaning parties focusing more heavily on the negative impacts of CC and economics-oriented parties concentrating more on economic transformation and technological advancements. The AfD is the only party denying the need to adopt CC mitigating measures and demeaning those concerned about a deteriorating climate as climate cult or fanatics. Differences between tweets and speeches are less salient.

In a follow-up study, we plan to investigate the climate communicative behavior of individual MPs with respect to their roles in climate-related committees, as well as conduct a time-sensitive analysis using dynamic topic modeling.

Acknowledgements

Stephan Lewandowsky (SL) and Christoph Abels are supported by an Advanced Grant from the European Research Council (101020961 PRODEM-INFO) to SL.

References

Jonathan Bright, Scott Hale, Bharath Ganesh, Andrew Bulovsky, Helen Margetts, and Phil Howard. 2020. Does Campaigning on Social Media Make a Difference? Evidence From Candidate Use of Twitter During the 2015 and 2017 U.K. Elections. *Communication Research*, 47(7):988–1009.

Jaime Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, page 335–336, New York, NY, USA. Association for Computing Machinery.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

J. Farrell. 2019. The growth of climate change misinformation in US philanthropy: evidence from natural language processing. *Environmental Research Letters*, 14(3).

Deen Freelon and David Karpf. 2015. Of big birds and bayonets: hybrid Twitter interactivity in the 2012 Presidential debates. *Information, Communication Society*, 18(4):390–406.

Todd Graham, Marcel Broersma, Karin Hazelhoff, and Guido van 't Haar. 2013. BETWEEN BROADCASTING POLITICAL MESSAGES AND INTERACTING WITH VOTERS. *Information, Communication Society*, 16(5):692–716.

Matthew P. Greenwell and Thomas F. Johnson. 2023. Is it all talk: Do politicians that promote environmental messages on social media actually vote-in environmental policy? *Energy, Ecology and Environment*, 8(1):17–27.

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure.

Kristof Jacobs and Niels Spierings. 2016. Saturation or maturation? The diffusion of Twitter and its impact on preference voting in the Dutch general elections of 2010 and 2012. *Journal of Information Technology and Politics*, 13(1):1–21.

Ye Jiang, Xingyi Song, Jackie Harrison, Shaun Quegan, and Diana Maynard. 2017. Comparing attitudes to climate change in the media using sentiment analysis based on Latent Dirichlet Allocation. In *Proceedings of the 2017 EMNLP Workshop: Natural Language Processing meets Journalism*, pages 25–30, Copenhagen, Denmark. Association for Computational Linguistics.

Franziska B. Keller, David Schoch, Sebastian Stier, and Jung Hwan Yang. 2020. Political Astroturfing on Twitter: How to Coordinate a Disinformation Campaign. *Political Communication*, 37(2):256–280.

Lucy Kinski and Ariadna Ripoll Servent. 2022. Framing Climate Policy Ambition in the European Parliament. *Politics and Governance*, 10(3):251–263.

Jana Lasser, Segun Taofeek Aroyehun, Almog Simchon, Fabio Carrella, David Garcia, and Stephan Lewandowsky. 2022. Social media sharing of low-quality news sources by political elites. *PNAS Nexus*, 1(4). Pgac186.

Leland McInnes, John Healy, and Steve Astels. 2017. hdbscan: Hierarchical density based clustering. *Journal of Open Source Software*, 2(11):205.

Leland McInnes, John Healy, and James Melville. 2020. Umap: Uniform manifold approximation and projection for dimension reduction.

Kevin Munger, Richard Bonneau, Jonathan Nagler, and Joshua A. Tucker. 2019. Elites tweet to get feet off the streets: Measuring regime social media strategies during protest. *Political Science Research and Methods*, 7(4):815–834.

Finn Müller-Hansen, Max W. Callaghan, Yuan Ting Lee, Anna Leipprand, Christian Flachsland, and Jan C. Minx. 2021. Who cares about coal? Analyzing 70 years of German parliamentary debates on coal with dynamic topic modeling. *Energy Research Social Science*, 72(101869).

Todd P. Newman. 2017. Tracking the release of IPCC AR5 on Twitter: Users, comments, and sources following the release of the Working Group I Summary for Policymakers. *Public Understanding of Science*, 26(7):815–825.

Thomas Plotkowiak and Katarina Stanoevska-Slabeva. 2013. German politicians and their Twitter networks in the Bundestag Election 2009. *First Monday*, 18(5):1–19.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Florian Richter, Philipp Koch, Oliver Franke, Jakob Kraus, Fabrizio Kuruc, Anja Thiem, Judith Högerl, Stella Heine, and Konstantin Schöps. 2020. Open Discourse.

William J. Ripple, Christopher Wolf, Thomas M. Newsome, Jillian W. Gregg, Timothy M. Lenton, Ignacio Palomo, Jasper A J Eikelboom, Beverly E. Law, Saleemul Huq, Philip B. Duffy, and Johan Rockström. 2021. World Scientists' Warning of a Climate Emergency. *BioScience*, 71(9):894–898.

Manfred Stede and Ronny Patz. 2021. The climate change debate and natural language processing. In *Proceedings of the 1st Workshop on NLP for Positive Impact*, pages 8–18, Online. Association for Computational Linguistics.

Sebastian Stier, Arnim Bleier, Haiko Lietz, and Markus Strohmaier. 2020. Election Campaigning on Social Media: Politicians, Audiences, and the Mediation of Political Communication on Facebook and Twitter. *Political Communication*, 35(1):50–74.

Roopal Vaid, Kartikey Pant, and Manish Shrivastava. 2022. Towards fine-grained classification of climate change related social media text. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 434–443, Dublin, Ireland. Association for Computational Linguistics.

Peter Van Aelst, Patrick van Erkel, Evelien D'heer, and Raymond A. Harder. 2017. Who is leading the campaign charts? Comparing individual popularity on old and new media. *Information Communication and Society*, 20(5):715–732.

Stefanie Walter, Ines Lörcher, and Michael Brüggemann. 2019. Scientific networks on Twitter: Analyzing scientists' interactions in the climate change debate. *Public Understanding of Science*, 28(6):696–712.

Frank Wendler. 2022. *Framing Climate Change in the EU and US After the Paris Agreement*. palgrave macmillan, Cham/Switzerland.

A Keywords

- Table 6: Keywords.

B Basic Corpus Statistics

- Table 7: Tweet and Speech Distribution By Party.

C Topics

- Table 8: Topics Tweet Corpus.
- Table 9: Topics Tweet Corpus The Left.
- Table 10: Topics Tweet Corpus The Greens.
- Table 11: Topics Tweet Corpus SPD.
- Table 12: Topics Tweet Corpus FDP.
- Table 13: Topics Tweet Corpus CDU/CSU.
- Table 14: Topics Tweet Corpus AfD.
- Table 15: Topics Speech Corpus.
- Table 16: Topics Speech Corpus The Left.
- Table 17: Topics Speech Corpus The Greens.
- Table 18: Topics Speech Corpus SPD.
- Table 19: Topics Speech Corpus FDP.
- Table 20: Topics Speech Corpus CDU/CSU.
- Table 21: Topics Speech Corpus AfD.

Keywords Tweets	Keywords Speeches
klima	klima
erwärmung	erwärmung
treibhaus	treibhaus
co2	co2
kohle	kohle
energiewende	energiewende
verkehrswende	verkehrswende
#fff	-
#fridaysforfuture	fridays for future
#endegelände	ende gelände
#endegelaende	-
#letztegeneration	-
#extinctionrebellion	extinction rebellion

Table 6: Keywords.

Party Name	Tweet Count	Tweet Proportion	Speech | Paragraph Count	Speech Proportion
The Left	3,208	0.11	660 | 1609	0.12
The Greens	13,509	0.45	1038 | 2767	0.20
SPD	4,023	0.13	1180 | 3028	0.22
FDP	4,057	0.13	416 | 1086	0.08
CDU/CSU	3,237	0.11	1446 | 3811	0.28
AfD	2,208	0.07	611 | 1493	0.11

Table 7: Tweet and Speech Distribution By Party.

Topic 1 (n: 2,890)	Score	Topic 2 (n: 1,346)	Score	Topic 3 (n: 641)	Score
emissionen	0.151	bahn	0.203	2038	0.275
emissionshandel	0.146	straße	0.194	2030	0.229
vermieter	0.113	straßen	0.171	2022	0.172
diesel	0.113	berlin	0.166	2035	0.158
1990	0.103	autobahn	0.162	2045	0.152
reduzieren	0.102	straßenbau	0.122	2025	0.139
emissionshandels	0.101	bahnhof	0.113	kohleausstieg	0.136
benzin	0.100	bundesstraßen	0.111	2023	0.125
fuels	0.099	bahnstrecken	0.110	2026	0.114
tonne	0.097	parkplätze	0.096	2029	0.108
Topic 4 (n: 537)	Score	Topic 5 (n: 525)	Score	Topic 6 (n: 458)	Score
klimakrise	0.143	energiewende	0.243	eu	0.218
klimapolitik	0.141	energie	0.179	europa	0.216
klimapaket	0.125	energieversorgung	0.145	europäische	0.197
klimawandel	0.122	energiesicherheit	0.134	europäischen	0.165
klimapolitisch	0.117	energieeffizienz	0.121	multilateralismus	0.133
klimakompetenz	0.104	energiekonzerne	0.120	europäisch	0.131
klimanotstand	0.103	energieerzeugung	0.113	nationale	0.128
klimagerechtigkeit	0.103	energieeffizientes	0.108	europäischer	0.114
klimaprogramme	0.099	energiepreise	0.105	nationalismus	0.113
shutdown	0.099	energiepolitische	0.105	europäisches	0.106
Topic 7 (n: 444)	Score	Topic 8 (n: 384)	Score	Topic 9 (n: 362)	Score
klimaschutzes	0.128	politik	0.220	innovationen	0.318
klimaschutzmaßnahmen	0.125	politiker	0.167	innovation	0.287
klimaschützer	0.121	politischen	0.153	technologien	0.253
klimaschutzministerium	0.110	politikern	0.146	technologieoffenheit	0.224
naturschutz	0.109	politikerinnen	0.116	innovationskraft	0.160
marktzentrierter	0.108	polizeigewalt	0.113	technologischen	0.158
klimaschutzaktivisten	0.108	haushaltspolitiker	0.103	technologieoffene	0.145
lebensqualität	0.104	polizisten	0.101	technologieweltmeister	0.142
klimaschutzkonzept	0.101	klassenpolitik	0.099	technologieoffen	0.140
elitenprojekt	0.096	klimaschutzpolitik	0.098	innovativ	0.120

Table 8: Topics Tweet Corpus (n: 30,242).

Topic 1 (n: 232)	Score	Topic 2 (n: 216)	Score	Topic 3 (n: 136)	Score
politik	0.248	klimapolitik	0.212	profite	0.283
union	0.243	klimaschutz	0.209	wirtschaft	0.277
fdp	0.240	klimaziele	0.182	investitionen	0.270
gerechtigkeit	0.240	klima	0.173	profit	0.258
linke	0.223	klimakrise	0.161	finanzieren	0.199
neoliberale	0.191	klimakatastrophe	0.161	profitinteressen	0.192
Topic 4 (n: 106)	Score	Topic 5 (n: 68)	Score	Topic 6 (n: 61)	Score
co2	0.535	2030	0.428	straße	0.495
emissionen	0.349	2038	0.418	straßen	0.377
ausstoß	0.269	2035	0.320	tausenden	0.319
sinken	0.205	2022	0.297	stadt	0.272
vermieter	0.204	2020	0.262	toll	0.253
co^2	0.200	idealerweise	0.253	freitag	0.238

Table 9: Topics Tweet Corpus (The Left; n: 3,208).

Topic 1 (n: 1,011)	Score	Topic 2 (n: 689)	Score	Topic 3 (n: 606)	Score
klimakrise	0.139	autobahnen	0.216	co2	0.297
klimaanpassung	0.119	bahn	0.214	emissionen	0.205
klimaregierung	0.106	straße	0.211	bepreisung	0.171
klimaschutzmaßnahmen	0.104	straßen	0.196	reduzieren	0.143
klimapaket	0.099	berlin	0.171	vermieter	0.136
klimakonferenz	0.098	autobahn	0.169	tonne	0.135
Topic 4 (n: 415)	Score	Topic 5 (n: 320)	Score	Topic 6 (n: 286)	Score
2038	0.293	europa	0.288	politik	0.280
2030	0.277	green	0.233	politische	0.202
2035	0.182	europäische	0.186	politischen	0.176
2019	0.173	europäischen	0.185	politiker	0.174
2022	0.165	europäisch	0.175	politisch	0.155
2021	0.155	greenwashing	0.173	polizei	0.150

Table 10: Topics Tweet Corpus (The Greens; n: 13,509).

Topic 1 (n: 287)	Score	Topic 2 (n: 143)	Score	Topic 3 (n: 115)	Score
co2	0.401	berlin	0.328	2030	0.390
vermieter	0.269	bahn	0.324	2038	0.343
bepreisung	0.260	straße	0.292	2045	0.291
mieter	0.233	bahnfahren	0.280	2040	0.267
emissionen	0.221	hamburg	0.228	2021	0.265
ausstoß	0.216	städten	0.219	2050	0.259
Topic 4 (n: 105)	**Score**	**Topic 5 (n: 82)**	**Score**	**Topic 6 (n: 81)**	**Score**
klimaschutz	0.256	eu	0.436	energiewende	0.435
klar	0.199	ratspräsidentschaft	0.367	energien	0.375
elitenprojekt	0.187	europäische	0.277	erneuerbaren	0.342
vorgeschoben	0.187	europas	0.225	energie	0.283
moorschutz	0.187	rat	0.224	thyssengas	0.237
klimaprogramme	0.187	europäischer	0.217	energy	0.237

Table 11: Topics Tweet Corpus (SPD; n: 4,023).

Topic 1 (n: 1,035)	Score	Topic 2 (n: 345)	Score	Topic 3 (n: 175)	Score
emissionen	0.221	klimapolitik	0.237	innovationen	0.356
co2	0.214	klimaschutz	0.225	innovation	0.319
emissionshandel	0.214	klima	0.189	technologieoffenheit	0.290
ausstoß	0.185	klimawandel	0.185	digitalisierung	0.273
ets	0.181	retten	0.171	technologien	0.265
zertifikate	0.157	klimaziele	0.166	bildung	0.238
Topic 4 (n: 78)	**Score**	**Topic 5 (n: 70)**	**Score**	**Topic 6 (n: 70)**	**Score**
2030	0.370	energiewende	0.490	wirtschaft	0.314
2038	0.365	energien	0.350	planwirtschaftlichen	0.272
2022	0.300	energie	0.333	planwirtschaftlicher	0.262
2021	0.284	energiepolitischen	0.309	wirtschaftliches	0.255
2035	0.274	energieträger	0.276	plan	0.248
2026	0.258	volatile	0.274	sonderwirtschaftsregionen	0.240

Table 12: Topics Tweet Corpus (FDP; n: 4,057).

Topic 1 (n: 369)	Score	Topic 2 (n: 130)	Score	Topic 3 (n: 82)	Score
co2	0.362	2020	0.314	innovationen	0.488
ausstoß	0.244	2038	0.304	technologien	0.435
emissionen	0.235	früher	0.245	innovation	0.349
bepreisung	0.201	nrw	0.243	friedman	0.288
reduziert	0.180	beschlossen	0.233	technik	0.286
heißt	0.176	jahre	0.218	extremismus	0.259
Topic 4 (n: 75)	**Score**	**Topic 5 (n: 51)**	**Score**	**Topic 6 (n: 50)**	**Score**
wirtschaft	0.361	industrieland	0.486	energiewende	0.475
marktwirtschaft	0.294	industrie	0.356	energien	0.358
wirtschaftlichkeit	0.241	industriestandort	0.300	gas	0.315
wirtschaftlich	0.239	industrienation	0.285	offshore	0.312
landwirtschaft	0.230	sozialverträglich	0.255	windenergie	0.309
wirtschaftliche	0.230	strukturwandel	0.241	pv	0.309

Table 13: Topics Tweet Corpus (CDU/CSU; n: 3,237).

Topic 1 (n: 169)	Score	Topic 2 (n: 120)	Score	Topic 3 (n: 73)	Score
co2	0.490	klimawandel	0.317	klimahysterie	0.316
ausstoß	0.258	klima	0.288	sozialismus	0.297
emissionen	0.223	klimaschutzes	0.261	klima	0.294
einführung	0.193	wärmer	0.253	klimaziele	0.269
gegen	0.192	weltklima	0.250	klimahysteriker	0.266
abschaffen	0.190	klimaschutz	0.245	türkei	0.265
Topic 4 (n: 69)	**Score**	**Topic 5 (n: 67)**	**Score**	**Topic 6 (n: 36)**	**Score**
diesel	0.475	energiewende	0.497	wind	0.465
autos	0.447	energie	0.386	windräder	0.419
benzin	0.400	energiepolitik	0.334	windkraftanlagen	0.367
liter	0.394	kernenergie	0.254	windrad	0.367
ein	0.233	elon	0.240	windkraft	0.367
emissionsfreien	0.232	kostensteigerung	0.240	windkraftwerk	0.327

Table 14: Topics Tweet Corpus (AfD; n: 2,208).

Topic 1 (n: 1,062)	Score	Topic 2 (n: 433)	Score	Topic 3 (n: 351)	Score
co2	0.229	euro	0.267	europa	0.216
emissionshandel	0.198	investitionen	0.194	eu	0.215
emissionen	0.190	investieren	0.163	europäische	0.213
ausstoß	0.164	ausgaben	0.143	europäischen	0.209
emissionshandels	0.124	investiert	0.141	sicherheitspolitik	0.134
carbon	0.120	finanzieren	0.133	nato	0.131
atmosphäre	0.118	finanzierung	0.125	europas	0.131
sektoren	0.114	billionen	0.124	multilateralismus	0.128
leakage	0.105	subventionen	0.121	europäer	0.121
reduzierung	0.097	bundeshaushalt	0.120	terrorismus	0.120
Topic 4 (n: 279)	Score	Topic 5 (n: 239)	Score	Topic 6 (n: 210)	Score
2030	0.265	klimawandel	0.236	energiewende	0.268
klimaschutzplan	0.198	klimapolitik	0.229	energien	0.226
klimaziele	0.168	klima	0.218	energie	0.154
klimaneutralität	0.154	klimadogma	0.209	erneuerbarer	0.138
klimaschutzprogramm	0.150	klimavertagungsprogramm	0.193	energieeffizienz	0.131
koalitionsvertrag	0.147	klimapaket	0.181	fertigstellung	0.128
klimaschutzziel	0.137	klimabesteuerung	0.178	umweltverträglichkeit	0.128
ambitionierter	0.136	abrüstung	0.162	energiepolitische	0.123
fahrplan	0.123	klimapolitisch	0.158	energieversorgung	0.122
jahre	0.123	klimaretter	0.157	energiekonzern	0.119
Topic 7 (n: 186)	Score	Topic 8 (n: 165)	Score	Topic 9 (n: 154)	Score
bahn	0.373	politik	0.264	durchzuführen	0.207
straße	0.231	demokratie	0.239	effizienten	0.185
straßen	0.217	klimaschutzpolitik	0.199	klimaschutz	0.177
straßenverkehr	0.192	menschenrechte	0.183	kämpfe	0.177
straßenbau	0.173	polizisten	0.172	klimaschutzplan	0.174
finanzierungskreislauf	0.151	demokratische	0.163	erweitert	0.169
wasserstraßen	0.145	gesellschaftliches	0.160	klimaschutzgesetz	0.167
autobahn	0.141	menschenrechtsschutz	0.159	brandt	0.161
bahnreform	0.135	kulturpolitik	0.155	klimaschutzhaushalt	0.156
bahnhöfen	0.132	populisten	0.146	klimaschutzbemühungen	0.156

Table 15: Topics Speech Corpus (n: 13,794).

Topic 1 (n: 208)	Score	Topic 2 (n: 108)	Score	Topic 3 (n: 84)	Score
klimapolitik	0.225	energiewende	0.400	co2	0.390
kollegen	0.222	energien	0.322	co	0.333
klimaschutz	0.207	bürgerenergie	0.318	emissionen	0.299
klimawandel	0.206	energie	0.298	ccs	0.227
haben	0.199	windkraft	0.241	tonne	0.227
zuschauer	0.190	pv	0.224	autos	0.224
Topic 4 (n: 82)	Score	Topic 5 (n: 50)	Score	Topic 6 (n: 37)	Score
bahn	0.351	euro	0.426	digitalisierung	0.349
autobahnen	0.310	milliarden	0.373	transformation	0.297
straßenverkehr	0.290	lufthansa	0.271	industrie	0.277
straßen	0.275	einzelplan	0.271	industriepolitik	0.268
straße	0.243	investitionen	0.265	internet	0.261
straßenbau	0.229	finanzieren	0.236	interdisziplinarität	0.248

Table 16: Topics Speech Corpus (The Left; n: 1,609).

Topic 1 (n: 306)	Score	Topic 2 (n: 186)	Score	Topic 3 (n: 129)	Score
klimakrise	0.231	co2	0.380	energiewende	0.349
klimaschutz	0.224	co	0.288	erneuerbaren	0.329
klimaforschung	0.184	emissionen	0.263	energien	0.326
klimapolitik	0.182	emissionshandel	0.257	windenergie	0.266
deutschland	0.166	ausstoß	0.223	energieeffizienz	0.236
haben	0.164	reduzieren	0.181	wind	0.185
Topic 4 (n: 94)	Score	Topic 5 (n: 93)	Score	Topic 6 (n: 70)	Score
kohlekommission	0.320	europäische	0.341	2020	0.382
kohleausstieg	0.254	europäischen	0.315	2030	0.379
kohle	0.246	eu	0.311	klimaschutzziel	0.303
kohlekraftwerk	0.236	europa	0.309	klimaschutzplan	0.278
kohleausstiegsgesetz	0.216	europas	0.210	kohleausstieg	0.252
datteln	0.213	brexit	0.202	klimapaket	0.233

Table 17: Topics Speech Corpus (The Greens; n: 2,767).

Topic 1 (n: 116)	Score	Topic 2 (n: 100)	Score	Topic 3 (n: 100)	Score
klimaschutz	0.298	energiewende	0.342	co2	0.430
kostet	0.260	energien	0.309	bepreisung	0.335
klimawandel	0.253	erneuerbaren	0.298	co	0.310
verschwörung	0.251	energie	0.239	ausstoß	0.271
bürgerinnen	0.230	energieversorgung	0.206	tonne	0.222
klimapaket	0.224	energieforschung	0.187	emissionen	0.201
Topic 4 (n: 90)	Score	Topic 5 (n: 83)	Score	Topic 6 (n: 64)	Score
technologien	0.229	europa	0.373	euro	0.487
industrie	0.215	eu	0.313	milliarden	0.451
digitalisierung	0.205	europäische	0.271	investieren	0.315
innovationen	0.204	europäischen	0.269	investitionen	0.290
wirtschaft	0.200	ratspräsidentschaft	0.202	forschung	0.238
gewerkschaften	0.186	migration	0.193	bildung	0.234

Table 18: Topics Speech Corpus (SPD; n: 3,028).

Topic 1 (n: 186)	Score	Topic 2 (n: 40)	Score
emissionshandel	0.407	energiewende	0.635
co2	0.402	vermieter	0.397
co	0.385	energiepolitik	0.388
das	0.369	mieter	0.387
auf	0.333	energie	0.355
nicht	0.323	allerletzter	0.355

Table 19: Topics Speech Corpus (FDP; n: 1,086).

Topic 1 (n: 85)	Score	Topic 2 (n: 84)	Score	Topic 3 (n: 71)	Score
co2	0.349	europa	0.359	energiewende	0.439
emissionen	0.252	europäische	0.325	erneuerbaren	0.263
ksg	0.228	eu	0.323	energien	0.261
emissionshandel	0.215	union	0.319	haben	0.218
zur	0.207	europäischen	0.273	erfolg	0.204
maßnahmen	0.204	russland	0.245	energieversorgung	0.202
Topic 4 (n: 51)	Score	Topic 5 (n: 40)	Score	Topic 6 (n: 38)	Score
bahn	0.357	elektromobilität	0.423	landwirtschaft	0.377
euro	0.329	mobilität	0.313	landwirte	0.272
mobilität	0.322	fuels	0.311	bauern	0.245
milliarden	0.308	elektroautos	0.258	land	0.231
verkehrsträger	0.284	kraftstoffen	0.256	forstwirtschaft	0.228
millionen	0.236	verbrennungsmotor	0.243	erwartungen	0.221

Table 20: Topics Speech Corpus (CDU/CSU; n: 1,446).

Topic 1 (n: 244)	Score	Topic 2 (n: 217)	Score	Topic 3 (n: 103)	Score
energiewende	0.336	co2	0.347	klimawandel	0.353
strom	0.289	co	0.319	klima	0.313
energien	0.232	emissionen	0.297	klimareligion	0.255
gigawatt	0.227	ausstoß	0.229	öffentlichen	0.230
wind	0.225	ipcc	0.220	menschengemachten	0.218
gas	0.207	celsius	0.218	haben	0.214
Topic 4 (n: 75)	Score	Topic 5 (n: 62)	Score	Topic 6 (n: 60)	Score
klimawandel	0.374	euro	0.527	wirtschaft	0.324
klimadogma	0.344	milliarden	0.481	eu	0.322
klima	0.332	millionen	0.281	budgetrecht	0.264
ablasshandel	0.315	000	0.267	europäischen	0.259
keine	0.304	ministerin	0.255	gesellschaft	0.240
klimabesteuerung	0.293	vierköpfige	0.247	planwirtschaft	0.232

Table 21: Topics Speech Corpus (AfD; n: 1,493).

Modelling Political Aggression on Social Media Platforms

Akash Rawat[1], **Nazia Nafis**[1], **Dnyaneshwar Bhadane**[1],
Diptesh Kanojia[1,2], **Rudra Murthy**[1,3]
[1]Indian Institute of Information Technology Lucknow, India.
[2]Surrey Institute for People-Centred AI, University of Surrey, United Kingdom.
[3]IBM Research, Bangalore, India.
[1]{mcs21014,mcs21004,mcs21010}@iiitl.ac.in
[2]d.kanojia@surrey.ac.uk, [3]rmurthyv@in.ibm.com

Abstract

Recent years have seen a proliferation of aggressive social media posts, often wreaking even real-world consequences for victims. Aggressive behaviour on social media is especially evident during important sociopolitical events such as elections, communal incidents, and public protests. In this paper, we introduce a dataset in English to model political aggression[1]. The dataset comprises public tweets collated across the time-frames of two of the most recent Indian general elections. We manually annotate this data for the task of aggression detection and analyze this data for aggressive behaviour. To benchmark the efficacy of our dataset, we perform experiments by fine-tuning pre-trained language models and comparing the results with models trained on an existing but general domain dataset. Our models consistently outperform the models trained on existing data. Our best model achieves a macro F1-score of 66.66 on our dataset. We also train models on a combined version of both datasets, achieving the best macro F1-score of 92.77, on our dataset. Additionally, we create subsets of code-mixed and non-code-mixed data from the combined dataset to observe variations in results due to the Hindi-English code-mixing phenomenon. We publicly release the anonymized data, code, and models for further research.

1 Introduction

In recent years, social media has risen as one of the most popular ways in which people share opinions with each other (Pelicon et al., 2019). On such platforms, anonymity is a major factor that impacts user behavior (Bernstein et al., 2011; Postmes et al., 1998), and the possibility of posting anonymously on platforms such as Twitter and Reddit has changed the way people communicate (Décieux et al., 2019). This has given rise to a significant amount of aggressive behavior- including but not limited to the use of snide remarks, abusive words, and personal attacks, going as far as rape threats (Hardaker and McGlashan, 2016). Modern definitions of human aggression establish it as *any behavior enacted with the intention of harming another person who is motivated to avoid that harm* (Anderson et al., 2002; Bushman and Huesmann, 2014). Aggression is now defined as social behavior patterns, and several studies have noted the proliferation of abusive language and an increase in aggressive content on social media (Mantilla, 2013; Suzor et al., 2019). Such behavior begets the automated analysis of social media content for aggressive behavior, lying at the intersection of Natural Language Processing (NLP) and Computational Social Sciences (CSS).

NLP research community has proposed various tasks to analyze aggressive behavior, some of which are well-known, *viz.,* offensive language identification (Zampieri et al., 2020), hate speech detection (Warner and Hirschberg, 2012; MacAvaney et al., 2019; Paz et al., 2020), aggression detection (Kumar et al., 2018b,a), cyber bullying (Dadvar et al., 2013; Kumar et al., 2020), and so on. Various shared tasks have been organized for these NLP sub-areas, motivating us to investigate the phenomenon of aggression on social media. Aggression is displayed not only by unnamed and anonymous troll accounts but, on occasion, by known personalities who can influence thousands of followers (O'Toole et al., 2014). However, we investigate this problem in the context of political trolling and aggression displayed on social media close to the government election.

In this paper, we investigate the task of aggression detection on a social media platform, *i.e.,* Twitter, in the context of Indian elections. We curate a set of political-themed tweets from the user handles of known personalities ($\sim$ 110 in number) and perform manual annotation to create a dataset for the task. Our annotation schema aligns with

---

[1] https://doi.org/10.5281/zenodo.7540489

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 497–510
July 14, 2023 ©2023 Association for Computational Linguistics

OAG	He will kill 22000, will abolish NREGA, nullify food security. NO control on the present. [MASK] baba derives satisfaction in being astro-baba.
CAG	Also at 9pm: Did you know our Parliament has a record number of MPs facing criminal cases? What does that tell you about our democracy?
NAG	We wont be detrimental to the development. We are partners in development & Progress.

Table 1: Examples of Overtly, Covertly, and Non-aggressive tweets from our dataset. [MASK] token is to avoid naming an individual in this example.

the existing aggression detection datasets where text sequences are labelled as overtly-aggressive, covertly-aggressive and non-aggressive as shown in Table 1 with an example for each class. We also collected the datasets released at TRAC-2018 (Kumar et al., 2018a) and TRAC-2020 (Kumar et al., 2020) shared tasks to benchmark task performance. With the help of pre-trained language models, we perform a topic analysis along with various experiments to perform the task of aggression detection and discuss the obtained results in terms of precision, recall, and macro F1 scores. We also perform transfer learning-based experiments to observe the cross-dataset performance.

While these datasets are mostly in the Latin script, many words belong to one of the Indian languages, such as Hindi but are transliterated into the Latin script. This led us to label our data instances as code-mixed *vs.* non-code-mixed using a known heuristics-based approach, and we performed additional experiments on these data sub-sets. Our **key contributions** are:

- We release an English tweet dataset to model political aggression along with our code and models[2].
- Experimental analysis of aggressive behavior with multiple subsets of our dataset.
- Evaluation of task performance using language models, including observations over the presence of Hindi-English code-mixing.

2 Related Work

The earliest approaches to the task of classifying derogatory messages used decision trees (Spertus, 1997). Manual rules with syntactic and semantic text features were the basis of these models. Since then, much of the focus has been on feature engineering the text which includes features like Bag-of-Words (BOW) (Kwok and Wang, 2013; Liu et al., 2019a), N-grams in the word level (Pérez and Luque, 2019; Liu and Forss, 2014; Watanabe et al., 2018), N-grams in character level (Gambäck and Sikdar, 2017; Pérez and Luque, 2019), typed dependencies (Burnap and Williams, 2016), part-of-speech tags (Davidson et al., 2017), dictionary-based approaches (Tulkens et al., 2016) and other lexicons (Burnap and Williams, 2016; Alorainy et al., 2019).

Later, word-embedding-based approaches for automatic extraction of semantic features reigned as state-of-the-art approaches (Nobata et al., 2016; Zhang et al., 2018; Badjatiya et al., 2017; Kshirsagar et al., 2018; Orăsan, 2018; Pratiwi et al., 2019; Galery et al., 2018). Approaches using Deep Neural Networks have also been explored in the literature (Nina-Alcocer, 2019; Ribeiro and Silva, 2019). Use of Convolutional Neural Networks (Gambäck and Sikdar, 2017; Roy et al., 2018; Huang et al., 2018), Long-Short Term Memory (LSTM) (Badjatiya et al., 2017; Pitsilis et al., 2018; Nikhil et al., 2018) and Gated Recurrent Unit (GRU) (Zhang et al., 2018; Galery et al., 2018) or a combination of different Deep Neural Network architectures in an ensemble setting (Madisetty and Sankar Desarkar, 2018) have been explored for obtaining better feature representation and thereby improving the aggression detection performance.

Recently, the use of contextual embedding-based approaches like ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019) have become state-of-the-art approaches in terms of performance on aggressive language identification tasks (Bojkovský and Pikuliak, 2019; Ramiandrisoa and Mothe, 2020; Mozafari et al., 2019). Particularly, the use of BERT-based approaches is gaining traction within shared tasks on abusive language detection for performance improvement. This can be observed in SemEval-2019 Task 6 (Zampieri et al., 2019) for English tweets, and TRAC (Kumar et al., 2018a) for Hindi and English tweets and Facebook comments. This motivates us to explore the use of several pre-trained language models for this work.

---

[2] `https://github.com/surrey-nlp/political-aggression-detection`

3 Datasets

For our experiments, we collected two datasets as described below.

3.1 D1 (TRAC Dataset)

The TRAC dataset used for our experiments is a collated and pruned version of two shared task datasets released with Trolling, Aggression, and Cyber-bullying (TRAC) 2018 (Kumar et al., 2018a) and 2020 (Kumar et al., 2020) workshops. This data has been crawled from Facebook, Twitter, and YouTube comments and was mainly collected from pages containing issues concerning the Indian population. As described in their papers, this dataset contains English and Hindi-English code-mixed data. However, upon manual observation, we noticed some Hindi instances too and pruned them using steps mentioned in section Code-mixed Data 3.3.1. The remaining instances only contain English and Hindi-English code-mixed data, and the number of instances for each class is shown in Table 2.

3.2 D2 (Our Dataset)

The objective of creating this dataset was to analyze aggression on social media, specifically in the context of Indian general elections. We could scrape approximately 10,000 tweets made through the public Twitter handles of 110 most influential Indian personalities. These tweets were made when general elections were held in India in 2014 and 2019. The ratio of tweets collected in the pre and post-election time frames was about 3 to 2. These personalities **belong to the following domains**:

- Political figures and official handles of political parties like Indian National Congress (INC), Bharatiya Janta Party (BJP), Aam Aadmi Party (AAP), and so on.
- Journalists, independent and affiliated with mainstream news organizations, followed by many people.
- Other prominent personalities who hold a massive follower count and make political tweets, such as actors, sports persons, *etc.* which can be considered influential.

We filtered these tweets based on the context, as we aim to model for political aggression, using some of the popular election-related keywords (such as “EVMs”, “rallies”, “election results”, *etc.*). This keyword-based manual pruning reduced the number of data instances to 2000. Data were sampled during collection based on language, including only English and some Hindi-English code-mixed data. We labeled it manually with the help of two annotators. Both our annotators are graduate students who are native speakers of Hindi, with proficiency in English and an understanding of the political context in which the tweets were made. We also assess the inter-annotator agreement using **Cohen’s Kappa** score and discuss it in a subsection below. The statistics for the datasets we use for experimentation are shown in Table 2.

	OAG	CAG	NAG	Total
D1 (TRAC)	2715	4093	5436	12244
D2 (Our)	489	519	992	2000
D3 (Combined)	3204	4612	6428	14244
Code-mixed	943	1364	2670	4977
Non-Code-mixed	2261	3248	3758	9267

Table 2: Statistics for the different datasets used in our experiments.

To perform experiments over a combined dataset, *we also concatenate both D1 and D2 to create D3 - a combined dataset* (Table 2). Our experiments include applying our aggression detection approaches to this dataset as well.

3.3 Code-mixing: A Challenge

Code-mixing is the intermixing of units like words or phrases from one language (embedded language) within a second or primary language (matrix language) (Sitaram et al., 2019). Some of the most prevalent instances of such types of sentences can be observed in Hinglish (Hindi-English) (Srivastava and Singh, 2021) and Spanglish (Spanish-English) (Bullock et al., 2019) datasets. Although such text can be considered informal, with the increasing number of multilingual speakers, its usage has become quite the norm today. Thus, it has become essential to study the opinions and mindsets of people using code-mixing to express their views, especially when investigating data from social media platforms The most popular platform for observing code-mixing nowadays is social media. With people expressing their innate views, understanding and analyzing such data has garnered interest from different research communities.

Since the data is not exclusive to a single language, there are challenges associated with handling it. Each language has its own set of rules. Standardizing text that deviates from a canonical form happens at the token level or even at the semantic level (Çetinoğlu et al., 2016; Parikh and Solorio, 2021). Parsing poses yet another problem due to the syntactic rules that apply to one language but not to the other and the fact that errors may propagate from the previous layer (Çetinoğlu et al., 2016). Further, language identification poses challenges when languages are closely related and have common false friends (semantically different words sharing the same ancestor language).

Such challenges make code-mixing data harder to work with, as compared to working with monolingual data or even datasets containing well-separated monolingual instances from multiple languages.

3.3.1 Code-mixed Data

Tweets posted in the Indian political context are known to contain code-mixed data, *i.e.,* the presence of transliterated Hindi words (written in the Latin script). Such data presents challenges even for pre-trained multilingual language models, as they do not encounter code-mixed data during pre-training.

We obtained two separate sub-parts from D3 (Combined) to address the challenges presented by code-mixed data - *code-mixed* and *non-code-mixed* (Table 2). To obtain this separation, we perform some initial pre-processing, use a heuristics-based approach and utilize a Language Identification (LID) Model (Nayak and Joshi, 2022) as follows:

1. All the punctuation marks, special symbols, and their respective words, for entities like @mentions, and #hashtags were removed from the sentences.
2. If the sequence length obtained after punctuation or special mention removal became null, those sentences were omitted (classified as non-code-mixed).
3. We provide each data instance after following the above steps as input to the LID model to obtain token-level labels.
4. For classifying the sentences into Hindi, English, and Hindi-English code-mixed categories, a range of thresholds from 2%-20% were applied. Finally, after observing the number of sentences that fell into each language across these thresholds, 12% was chosen as a filter for categorizing language for each sentence. For example, in a sentence containing 36 words, if 5 or more words (equivalent to greater than or equal to 12%) were identified as Hindi, it would be labeled as code-mixed; else, it would be counted as English.
5. There were instances where even the complete sentences were in Hindi. Such sentences were also removed as we were dealing primarily with English data, with instances of Hindi code-mixing.
6. We also mask all usernames using [MASK] in the tweets to avoid biasing our models.

Table 7 in the appendix section reports the language-wise statistics obtained after these steps. It is to be noted here that the above steps were performed only for the separation of code-mixed data *from* non-code-mixed data. No pre-processing was performed for the aggression detection task.

3.4 Dataset Validation and Analysis

D2 (Our Dataset) was curated from over $10,000$ tweets, and as discussed, keyword-based manual pruning with the help of annotators reduced the final data instances to $2,000$. This data was collected from tweets posted four months before the Indian elections and two months after the declaration of election results. Out of these $2,000$ tweets, $1,200$ were collected in the 'pre-election' period, and 800 were obtained in the 'post-results' period.

Two annotators labelled this data manually, and we obtained an inter-annotator agreement score of 0.76 (Cohens' Kappa), which indicates "substantial agreement" ($p < 0.05$). Our annotators belonged to different political ideologies, and substantial agreement was obtained on the aggression label. Given a disagreement on any instance, we obtained a class label on such instances with the help of a third annotator.

Dataset Analysis: We also perform topic modeling-based analysis on D2 (our dataset) using BERTopic (Grootendorst, 2022). In Figures 1, 2, and 3, we show the most frequently occurring topic-wise token distribution for top-k words (k = 5). We make the following observations:

1. **Overtly Aggressive**: Among the topics clustered by BERTopic, as seen in Figure 1, the

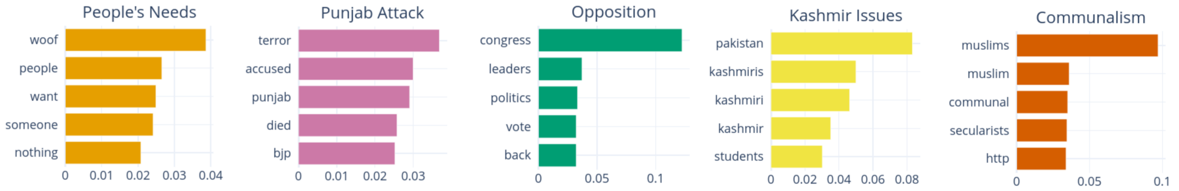


Figure 1: **Topics in the OAG category.** We note that people on Twitter are the most overtly aggressive about the political opposition, Kashmir and other internal security issues, and communal topics, whereas people's actual needs, and a terror attack that took place in the state of Punjab, take a backseat.

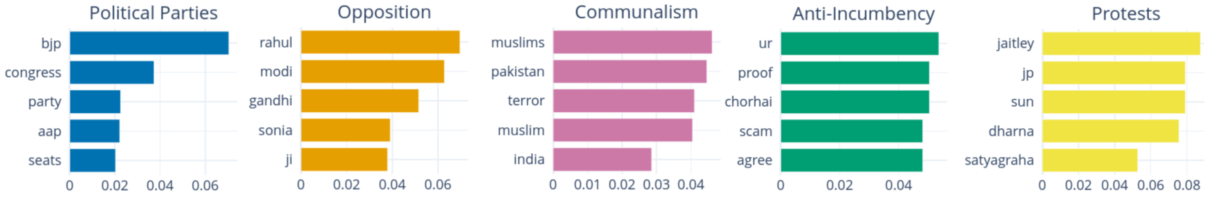


Figure 2: **Topics in the CAG category.** We note that most covert aggression is reserved for the protests going on around in the country for various issues, and an overall anti-incumbency sentiment against the ruling central government. Aggression is also prevalent against the political opposition as well as other political parties in general.

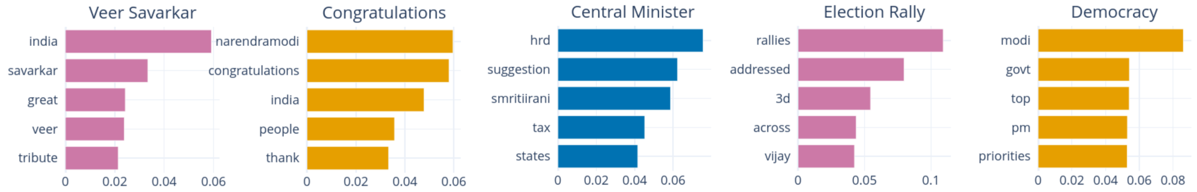


Figure 3: **Topics in the NAG category.** Most non-aggressive tweets correspond to the congratulatory messages sent to members of the party that emerged victorious in the elections. Discussions also take place around key cabinet roles and their possible contenders, election rallies, and the overall state of democracy in general.

data suggests that the most discussed topics where people were overtly aggressive in their tweets are *people's needs*, *Punjab attack*, *opposition*, *Kashmir issues*, and *communalism*. We note that political, religious, and national security issues are more aggressively discussed and debated online than people's needs for food and housing.

2. **Covertly Aggressive**: Similarly, the data (Figure 2) also suggests that there was minor aggressive behavior displayed when topics like *political parties*, *opposition*, *anti-incumbency*, *protests*, and *communalism* are concerned. Aggressiveness against the ruling party, the opposition, and all political parties, in general, is observed. This subsection also consists of tweets made with regard to various public protests that were being carried out in the run-up to the elections.

3. **Not Aggressive**: However, the data in Figure 3 shows us that social media discussions were non-aggressive when topics like *Veer Savarkar*, *congratulations*, *central minister*, *election rally*, and *democracy* are concerned. It includes congratulatory messages extended to the winning party members. Similarly, contenders for cabinet ministry posts are speculated, and the overall state of democracy is pondered upon.

We also perform additional topic modeling-based analysis for these tweets by segregating them into the 'pre-election' and 'post-result' periods (please see Appendix A.1). We choose four months before the election since this time is sensitive, and *exit polls* in the mainstream media start creating the election buzz. However, post-results, as observed from the data, the political scenario becomes rather concentrated on congratulating the winning party, diminishing data on development-related issues.

	TRAC (D1)	**Ours (D2)**	**Combined (D3)**	**Code-Mixed**	**Non Code-Mixed**
BERT$_{base}$	67.17±0.53	58.89±2.42	65.44±0.70	66.40±1.13	63.84±1.42
RoBERTa$_{base}$	69.05±0.57	**66.66±3.82**	66.85±1.23	65.16±2.06	65.11±1.08
ALBERT$_{base-v2}$	66.03±0.89	54.61±4.28	64.71±0.76	62.15±3.89	59.97±2.60
XLM-RoBERTa$_{base}$	67.73±2.02	61.08±2.21	62.88±3.08	64.52±2.56	60.97±2.73
MURIL$_{base}$	66.64±1.08	60.62±2.00	65.47±0.83	66.71±1.37	62.33±0.88
XLM-RoBERTa$_{large}$	68.00±1.29	66.38±1.84	**67.95±1.37**	67.83±2.52	64.92±0.97
Hing-BERT	**69.37±0.96**	62.41±3.02	67.48±1.91	**68.50±1.35**	65.13±1.62
Hing-mBERT	67.41±1.06	57.65±2.36	65.70±0.66	65.84±1.71	**65.84±1.40**
HingRoBERTa	68.85±1.28	64.81±2.79	66.95±1.43	68.36±1.71	63.11±1.85

Table 3: Mean macro F1-Score (F) from various pre-trained language models on TRAC (D1), Our Dataset (D2, Combined (D3), code-mixed and non-code-mixed subsets of D3; reported in percentage points. The values in **bold** highlight the best-performing language model on each dataset.

4 Approach

Recently, sequence classification via fine-tuning of pre-trained language models has become a standard approach for performing various NLP tasks. We take a similar approach and fine-tune various pre-trained language models for the task of aggression detection to report the results below. We select some monolingual, some multilingual, and some pre-trained language models specific to Hindi-English code-mixing.

Every sentence/tweet containing a sequence of words is tokenized into a sequence of sub-words using the model-specific tokenizer. The input to the model is a sequence of sub-word tokens that pass through the Transformer encoder layers. The output from the transformer is an encoder representation for each token in the sequence. We take the encoder representation of the [CLS] token in the case of BERT or the last encoder hidden states for other models. The output layer is a linear layer followed by *softmax* function, which takes in the above representation. The model is trained by optimizing for the cross-entropy loss value.

5 Experimental Setup

We fine-tune various pre-trained languages (both monolingual and multilingual) for the task of aggression detection and use the following pre-trained language models: BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019b), ALBERT (Lan et al., 2020) which are pre-trained over English data. We also include XLM-RoBERTa (Conneau et al., 2020), both base and large variants, which are trained over multilingual data (containing both English and Hindi) and MURIL (Khanuja et al., 2021), a multilingual language model specifically built for Indian language. Finally, HingBERT, HingMBERT, and, HingRoBERTa (Nayak and Joshi, 2022) are also included as they are pre-trained over code-mixed Hindi-English data.

Data Split and Evaluation We report macro F1 Score on TRAC Dataset (D1), Our dataset data set (D2), combined dataset (D3) along with code-mixed and non-code-mixed subsets of data as discussed in Section 3. For the train/validation/test split sizes, we choose uniform 80% / 10% / 10% from each dataset to perform our experiments. We additionally report results on the subset of data containing code-mixed instances extracted from the combined dataset. To demonstrate the efficacy of our dataset, we also perform zero-shot domain transfer experiments. We evaluate the model trained on the TRAC dataset and tested it on our dataset and vice-versa to report zero-shot domain transfer results in Table 5.

Experiment Settings We perform experiments using the Huggingface Transformers library (Wolf et al., 2020). We monitor the *validation* set Macro-F1 to find the best hyperparameter values. We use the following range of values for selecting the best hyperparameter:

- **Batch Size:** 8, 16, 32
- **Learning Rate:** 1e-5, 1e-6, 3e-5, 3e-6, 5e-5, 5e-6

We repeat each training five times with different random seeds and report the mean macro F1-score along with its standard deviation. Our experiments were performed using 2 x Nvidia RTX A5000 and a single training run usually takes approximately 1 hour, on the combined dataset. For other datasets, however, the runtime is approximately 30 minutes. We generate various models during our experiments where the number of trainable parameters varies from 100M to 200M depending on the language model used.

Custom Weighted Loss As the dataset exhibits class imbalance, we use weighted cross-entropy loss in all our experiments. We assign a weight to the loss of every instance depending on the class label. We find the percentage of examples by class belonging to each class from the train split. We take the inverse of the probability values as the weight for the particular class. In this way, we provide more importance to the instances belonging to the minority class.

6 Results and Discussion

We report the results obtained via fine-tuning pre-trained language models in this section. Table 3 reports the Test set F1-Score from various pre-trained language models on the TRAC dataset, our dataset, and combined results. In addition to this, we also present the scores on code-mixed and non-code-mixed subsets of the entire data. We observe that Hing-BERT model outperforms other pre-trained language models on the TRAC dataset, achieving the highest macro F1-score of 69.37 across all combinations. On our dataset, however, we observe that RoBERTa$_{base}$ outperforms other pre-trained language models. For clarity in resultant observations, we provide a separation between monolingual, multilingual, and language models pre-trained on the code-mixed data.

Multi-Dataset Fine-Tuning

From Table 4 we can observe that training on the combined dataset (D3) results in significant performance improvements on both our dataset and TRAC dataset. On the TRAC dataset we observe an increase in the best F-Score from 69.37 to 93.51. Similarly, we observe an increase in best F-Score from 66.66 to 92.77 on our dataset across models.

Code-mixed

As both the TRAC dataset and our dataset contain code-mixed instances, we fine-tune and report F-Score on these subsets of instances (Table 3). As expected, we get the best F1 score on code-mixed instances with *Hing** models. This may be attributed to the fact that *Hing-** models have been *pre-trained on millions of code-mixed Hindi-English sentences*. However, to our surprise, the Hing-mBERT model outperforms other monolingual and multilingual models on non-code-mixed data as well. This result may be attributed to the fact that a significant amount of code-mixed data used in the pre-training of the *Hing** models comes from the social-media domain.

Models	**D3 –>D1**	**D3 –>D2**
BERT$_{base}$	86.12±7.95	81.49±11.73
RoBERTa$_{base}$	90.56±2.71	90.01±3.51
ALBERT$_{base-v2}$	75.40±6.62	75.59±6.68
XLM-RoBERTa$_{base}$	78.14±11.05	73.09±15.13
MURIL$_{base}$	84.11±3.62	81.58±4.62
XLM-RoBERTa$_{large}$	87.87±2.74	88.02±6.18
HingBERT	89.57±6.56	87.86±10.19
Hing-mBERT	88.71±6.34	86.28±8.51
Hing-RoBERTa	**93.51±1.14**	**92.77±1.17**

Table 4: Zero-Shot Test Set F1-Score from various language models trained on D3; D1 represents the TRAC dataset, D2 is our manually curated dataset, and D3 is the combined dataset.

Models	**D1 –>D2**	**D2 –>D1**
BERT$_{base}$	48.82±2.55	50.55±1.33
RoBERTa$_{base}$	46.29±3.60	55.33±1.53
ALBERT$_{base-v2}$	46.32±2.58	47.14±1.23
XLM-RoBERTa$_{base}$	47.32±2.28	52.53±1.19
MURIL$_{base}$	48.77±3.42	52.49±0.68
XLM-RoBERTa$_{large}$	47.67±2.84	55.77±0.98
HingBERT	47.08±2.38	54.34±1.12
Hing-mBERT	43.06±3.38	52.09±1.87
Hing-RoBERTa	49.30±3.43	52.12±0.71

Table 5: Zero-Shot Test Set F1-Score from language models trained on D1 and D2 respectively. D1 represents the TRAC dataset, D2 is our manually curated dataset, and D3 is the combined dataset.

Zero-Shot Transfer Learning

Table 5 presents the results from our transfer learning setup. Columns D1 –>D2 and D2 –>D1 present a zero-shot setup from which we observe the performance of models fine-tuned on the D1 (TRAC) dataset and tested on D2 (our data) and vice-versa, respectively. From here, we observe that models trained on our dataset consistently obtain better F1-

Tweet	**GT**	**M1**	**M2**	**M3**	**Error Type**
As per Zee News 405 for seats for BJP in UP. Total constituency is 403. Two seats given by Zee News on free of cost.	**CAG**	NAG	NAG	NAG	**Sarcasm**
Finally paused the video . It's so nice now lol	**CAG**	NAG	NAG	NAG	**Sarcasm**
Do you know Malda. ??	**CAG**	NAG	CAG	NAG	**Short sequence**
Oh really	**CAG**	NAG	NAG	NAG	**Short sequence**
ek problem hai Main parents ke saath nahi dekh payunga.	**NAG**	CAG	NAG	NAG	**Code-Mixing**
Jay hind Pakistan me jabrdast Hamla Kare Hmari Sena jbab dena jaruri h	**CAG**	NAG	CAG	CAG	**Code-Mixing**

Table 6: Prediction on test set examples from some of the fine-tuned models. **GT**: Ground Truth label, **M1**: RoBERTa$_{base}$, **M2**: XLM-RoBERTa$_{large}$, **M3**: Hing-BERT.

Score compared to models trained on the TRAC dataset. This performance benchmark is surprising, given the dataset size of the TRAC data is larger compared to our data; and given approximately similar underlying class balance ratio for both datasets.

Discussion: Error Analysis

For error analysis, we pick the best-performing models on a combined dataset from the monolingual, multilingual, and code-mixed categories which were RoBERTa$_{base}$, XLM-RoBERTa$_{large}$ and Hing-BERT respectively.

Upon going through examples, we encountered various examples which were part of the TRAC dataset, where we found a disagreement with the annotated labels. For instance, the following sentences are labeled 'Not Aggressive', even though they have some amount of aggression:

- "Oh yeah cave civilisation can claim that.. After all u r their illegal creation"
- "He is modi dog. Godi media not usefull in India."

Leaving such disputed annotations aside, we report some of the most common error patterns in Table 6. Instances carrying sarcasm were quite often not recognized correctly by the three models, since it is not an easy task to recognize the latent intent or in this case, the aggression in such a sense. Another common error we noticed included very short sequences. Such types of sentences are quite common on social media, where these often carry some hidden context or a backstory. But the models find it difficult to predict the exact category for such examples. Finally, since the data contains some amount of code-mixing, we see a monolingual model, RoBERTa performing relatively worse than multilingual and code-mixed models like XLM-R and Hing-BERT which have seen more such kind of data while pre-training.

7 Conclusion and Future Work

In this paper, we curate a novel dataset to model political aggression. We analyze this dataset using various approaches like topic modeling, aggression detection, and report results. To benchmark our performance, we also perform the aggression detection task with the help of an existing dataset. Our results and analyses also take into account the code-mixing phenomenon observed on social media platforms. The zero-shot cross-dataset experiments show the efficacy of our dataset, which consistently outperforms the approaches used with existing data. While political aggression is subtle occasionally, we observe that some data instances show overtly aggressive behavior. It is important to note the limitations of such a study and we discuss them in the next section. We release any data, code, and models produced during this study (including any raw data, but keeping user handles anonymous) publicly for further research by the community. We license this release under CC-BY-NC-SA 4.0.

In the future, we aim to collect more data from multiple social media platforms and release it to model aggressive behavior. We plan to perform similar experiments on a large dataset while benchmarking and comparing our current models' performance. We also plan to investigate online or active learning for the same. Finally, we also aim to expand on the theoretical underpinnings of sublime aggression and offense by attempting to identify these within other more tangential domains, *viz.*, comedy.

Limitations

Our work can be considered to have the following possible limitations:

1. The dataset we introduce and use to perform analysis contains 2000 tweets sampled from a specific time frame over a single social media platform. However, we aim to extend this work by collecting more political data across various social media platforms and using it to model aggressive behavior. Please do note that these tweets have been manually filtered from a larger set of 10,000 tweets while manually labelling them and ensuring that they are relevant to the political domain.

2. The number of user handles that we scrape tweets from for this study is around 110. This number might not be reflective of a large political space considering the plethora of politically active personalities in India. However, it is noteworthy that each of these 110 user handles has a minimum of $100,000$ followers on Twitter, on the basis of which we consider them to be influential on a social media platform.

Ethics Statement

Our dataset of tweets was obtained by scraping Twitter. We also obtain a subset of data from existing aggressive detection datasets cited in this paper, complying with the terms of use of each of these datasets. All datasets were anonymized, no tweet-ids or Twitter usernames or any of their demographics are included in the data used to train our models. We plan to release only the tweet ids as part of our dataset, along with the labels, in the final version.

References

Wafa Alorainy, Pete Burnap, Han Liu, and Matthew L. Williams. 2019. "the enemy among us": Detecting cyber hate speech with threats-based othering language embeddings. *ACM Trans. Web*, 13(3).

Craig A Anderson, Brad J Bushman, et al. 2002. Human aggression. *Annual review of psychology*, 53(1):27–51.

Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion - WWW '17 Companion*. ACM Press.

Michael Bernstein, Andrés Monroy-Hernández, Drew Harry, Paul André, Katrina Panovich, and Greg Vargas. 2011. 4chan and/b: An analysis of anonymity and ephemerality in a large online community. In *Proceedings of the international AAAI conference on web and social media*, volume 5, pages 50–57.

Michal Bojkovský and Matúš Pikuliak. 2019. STUFIIT at SemEval-2019 task 5: Multilingual hate speech detection on Twitter with MUSE and ELMo embeddings. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 464–468, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Barbara Bullock, Wally Guzmán, and Almeida Jacqueline Toribio. 2019. The limits of Spanglish? In *Proceedings of the 3rd Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 115–121, Minneapolis, USA. Association for Computational Linguistics.

Peter Burnap and Matthew Leighton Williams. 2016. Us and them: identifying cyber hate on twitter across multiple protected characteristics. *Epj Data Science*, 5.

Brad J Bushman and L Rowell Huesmann. 2014. Twenty-five years of research on violence in digital games and aggression revisited: A reply to elson and ferguson (2013).

Özlem Çetinoğlu, Sarah Schulz, and Ngoc Thang Vu. 2016. Challenges of computational processing of code-switching. *arXiv preprint arXiv:1610.02213*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Maral Dadvar, Dolf Trieschnigg, Roeland Ordelman, and Franciska de Jong. 2013. Improving cyberbullying detection with user context. In *European Conference on Information Retrieval*, pages 693–696. Springer.

Thomas Davidson, Dana Warmsley, Michael W. Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *ICWSM*, pages 512–515.

Jean Philippe Décieux, Andreas Heinen, and Helmut Willems. 2019. Social media and its role in friendship-driven interactions among young people: A mixed methods study. *Young*, 27(1):18–31.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Thiago Galery, Efstathios Charitos, and Ye Tian. 2018. Aggression identification and multi lingual word embeddings. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 74–79, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Björn Gambäck and Utpal Kumar Sikdar. 2017. Using convolutional neural networks to classify hate-speech. In *Proceedings of the First Workshop on Abusive Language Online*, pages 85–90, Vancouver, BC, Canada. Association for Computational Linguistics.

Maarten Grootendorst. 2022. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*.

Claire Hardaker and Mark McGlashan. 2016. “real men don’t hate women”: Twitter rape threats and group identity. *Journal of Pragmatics*, 91:80–93.

Qianjia Huang, Diana Inkpen, Jianhong Zhang, and David Van Bruwaene. 2018. Cyberbullying intervention based on convolutional neural networks. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 42–51, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, et al. 2021. Muril: Multilingual representations for indian languages. *arXiv preprint arXiv:2103.10730*.

Rohan Kshirsagar, Tyrus Cukuvac, Kathy McKeown, and Susan McGregor. 2018. Predictive embeddings for hate speech detection on Twitter. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 26–32, Brussels, Belgium. Association for Computational Linguistics.

Ritesh Kumar, Atul Kr. Ojha, Bornini Lahiri, Marcos Zampieri, Shervin Malmasi, Vanessa Murdock, and Daniel Kadar, editors. 2020. *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*. European Language Resources Association (ELRA), Marseille, France.

Ritesh Kumar, Atul Kr. Ojha, Shervin Malmasi, and Marcos Zampieri. 2018a. Benchmarking aggression identification in social media. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 1–11, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Ritesh Kumar, Atul Kr. Ojha, Marcos Zampieri, and Shervin Malmasi, editors. 2018b. *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*. Association for Computational Linguistics, Santa Fe, New Mexico, USA.

Irene Kwok and Yuzhou Wang. 2013. Locate the hate: Detecting tweets against blacks. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, AAAI’13, page 1621–1622. AAAI Press.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Han Liu, Pete Burnap, Wafa Alorainy, and Matthew L. Williams. 2019a. A fuzzy approach to text classification with two-stage training for ambiguous instances. *IEEE Transactions on Computational Social Systems*, 6(2):227–240.

Shuhua Liu and Thomas Forss. 2014. Combining n-gram based similarity analysis with sentiment analysis in web content classification. In *Proceedings of the International Conference on Knowledge Discovery and Information Retrieval - Volume 1: SSTM, (IC3K 2014)*, pages 530–537. INSTICC, SciTePress.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Sean MacAvaney, Hao-Ren Yao, Eugene Yang, Katina Russell, Nazli Goharian, and Ophir Frieder. 2019. Hate speech detection: Challenges and solutions. *PloS one*, 14(8):e0221152.

Sreekanth Madisetty and Maunendra Sankar Desarkar. 2018. Aggression detection in social media using deep neural networks. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 120–127, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Karla Mantilla. 2013. Gendertrolling: Misogyny adapts to new media. *Feminist studies*, 39(2):563–570.

Marzieh Mozafari, Reza Farahbakhsh, and Noel Crespi. 2019. A bert-based transfer learning approach for hate speech detection in online social media.

Ravindra Nayak and Raviraj Joshi. 2022. L3cube-hingcorpus and hingbert: A code mixed hindi-english dataset and bert language models. In *Proceedings of The WILDRE-6 Workshop within the 13th Language Resources and Evaluation Conference*, pages 7–12, Marseille, France. European Language Resources Association.

Nishant Nikhil, Ramit Pahwa, Mehul Kumar Nirala, and Rohan Khilnani. 2018. LSTMs with attention for aggression detection. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 52–57, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Victor Nina-Alcocer. 2019. HATERecognizer at SemEval-2019 task 5: Using features and neural networks to face hate recognition. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 409–415, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive language detection in online user content. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, page 145–153, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Constantin Orăsan. 2018. Aggressive language identification using word embeddings and sentiment features. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 113–119, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Moderator: Mary Ellen O'Toole, Participants: Eugene V Beresin, Alan Berman, Steven M Gorelick, Jacqueline B Helfgott, and Chuck Tobin. 2014. Celebrities through violence: The copycat effect and the influence of violence in social media on mass killers. *Violence and gender*, 1(3):107–116.

Dwija Parikh and Thamar Solorio. 2021. Normalization and back-transliteration for code-switched data. In *Proceedings of the Fifth Workshop on Computational Approaches to Linguistic Code-Switching*, pages 119–124, Online. Association for Computational Linguistics.

María Antonia Paz, Julio Montero-Díaz, and Alicia Moreno-Delgado. 2020. Hate speech: A systematized review. *Sage Open*, 10(4):2158244020973022.

Andraž Pelicon, Matej Martinc, and Petra Kralj Novak. 2019. Embeddia at semeval-2019 task 6: Detecting hate with neural network and transfer learning approaches. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 604–610.

Juan Manuel Pérez and Franco M. Luque. 2019. Atalaya at SemEval 2019 task 5: Robust embeddings for tweet classification. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 64–69, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Georgios K. Pitsilis, Heri Ramampiaro, and Helge Langseth. 2018. Effective hate-speech detection in twitter data using recurrent neural networks. *Applied Intelligence*, 48:4730–4742.

Tom Postmes, Russell Spears, and Martin Lea. 1998. Breaching or building social boundaries? side-effects of computer-mediated communication. *Communication research*, 25(6):689–715.

Nur Indah Pratiwi, Indra Budi, and Ika Alfina. 2019. Hate speech detection on indonesian instagram comments using fasttext approach. In *2018 International Conference on Advanced Computer Science and Information Systems, ICACSIS 2018*, 2018 International Conference on Advanced Computer Science and Information Systems, ICACSIS 2018, pages 447–450, United States. Institute of Electrical and Electronics Engineers Inc. Publisher Copyright: © 2018 IEEE.; 10th International Conference on Advanced Computer Science and Information Systems, ICACSIS 2018 ; Conference date: 27-10-2018 Through 28-10-2018.

Faneva Ramiandrisoa and Josiane Mothe. 2020. Aggression identification in social media: a transfer learning based approach. In *TRAC*.

Alison Ribeiro and Nádia Silva. 2019. INF-HatEval at SemEval-2019 task 5: Convolutional neural networks for hate speech detection against women and immigrants on Twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 420–425, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Arjun Roy, Prashant Kapil, Kingshuk Basak, and Asif Ekbal. 2018. An ensemble approach for aggression identification in English and Hindi text. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 66–73, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Sunayana Sitaram, Khyathi Raghavi Chandu, Sai Krishna Rallabandi, and Alan W Black. 2019. A survey of code-switched speech and language processing. *arXiv preprint arXiv:1904.00784*.

Ellen Spertus. 1997. Smokey: Automatic recognition of hostile messages. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Conference on Innovative Applications of Artificial Intelligence*, AAAI'97/IAAI'97, page 1058–1065. AAAI Press.

Vivek Srivastava and Mayank Singh. 2021. HinGE: A dataset for generation and evaluation of code-mixed Hinglish text. In *Proceedings of the 2nd Workshop on*

Evaluation and Comparison of NLP Systems, pages 200–208, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nicolas Suzor, Molly Dragiewicz, Bridget Harris, Rosalie Gillett, Jean Burgess, and Tess Van Geelen. 2019. Human rights by design: The responsibilities of social media platforms to address gender-based violence online. *Policy & Internet*, 11(1):84–103.

Stéphan Tulkens, Lisa Hilte, Elise Lodewyckx, Ben Verhoeven, and Walter Daelemans. 2016. A dictionary-based approach to racism detection in dutch social media.

William Warner and Julia Hirschberg. 2012. Detecting hate speech on the world wide web. In *Proceedings of the second workshop on language in social media*, pages 19–26.

Hajime Watanabe, Mondher Bouazizi, and Tomoaki Ohtsuki. 2018. Hate speech on twitter: A pragmatic approach to collect hateful and offensive expressions and perform hate speech detection. *IEEE Access*, 6:13825–13835. Publisher Copyright: © 2018 IEEE.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. Semeval-2020 task 12: Multilingual offensive language identification in social media (offenseval 2020). *arXiv preprint arXiv:2006.07235*.

Z Zhang, D Robinson, and J Tepper. 2018. Detecting hate speech on twitter using a convolution-gru based deep neural network. In A Gangemi, R Navigli, M-E Vidal, P Hitzler, R Troncy, L Hollink, A Tordai, and M Alam, editors, *The Semantic Web: Proceedings of the 15th European Semantic Web Conference (ESWC 2018), Heraklion, Crete, Greece, 3-7 June 2018*, volume 10843 of *Lecture notes in computer science*, pages 745–760. Springer, Cham, Switzerland.

A Appendix

In this appendix, we provide additional details which could not be included in the paper. We start by showing the language-wise class distribution in our dataset, in Table 7.

	OAG	CAG	NAG	Total
English	2263	3254	3756	9273
Hindi(excluded from D3)	8	21	96	125
Code-Mixed	941	1358	2672	4971
Total	3212	4633	6524	14369

Table 7: Language-wise class distribution

A.1 Additional Dataset Analysis

We create two subsections of D2 (our dataset) by categorizing tweets that were made *before* the conduct of the elections (both 2014 and 2019) and *after* the declaration of results (both 2014 and 2019). We individually perform topic modeling on these subsections using BERTopic to get an insight into what issues were prominent before and after the elections.

Pre-Elections

1. **Overtly Aggressive**: Among the topics that were discussed online before elections, the most overtly aggressive debates happened on the tussle between *journalists* and the *ruling party*. *Casteism*, *communalism*, and national security including the *Kashmir issue* also find their way amongst the overtly aggressive tweets.

2. **Covertly Aggressive**: Our data analysis results also suggest that the top topics in the covertly aggressive category were *political parties*, *terrorism*, *development* and *unemployment* related issues.

3. **Not Aggressive**: The social media discussions were non-aggressive when topics like *martyrs*, and people's *rights* and the overall situation of *democracy* were being discussed. The presence of *religion* and *communalism* in this section also suggests that the peacemakers are equally active on this social media platform, as are the notorious aggressive tweeters.

Post-Results

We generally observe a stark decline in the number of aggressive tweets (OAG+CAG) when the post-

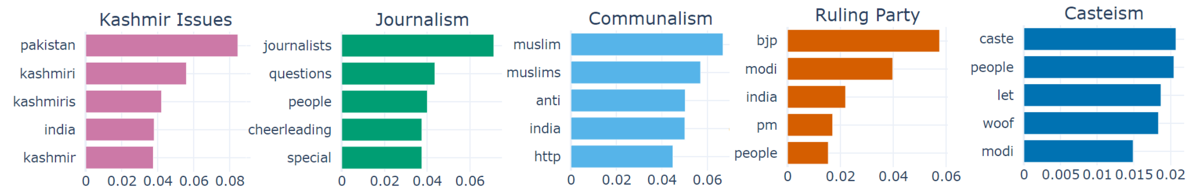


Figure 4: **Topics in the OAG category** in the 'Pre-Elections' sub-section of dataset D2. We note that there is a tussle between journalists in popular media and the ruling party. Aggressive tweets are also shared on topics of casteism, communalism, and issues related to Kashmir and the internal security of India.

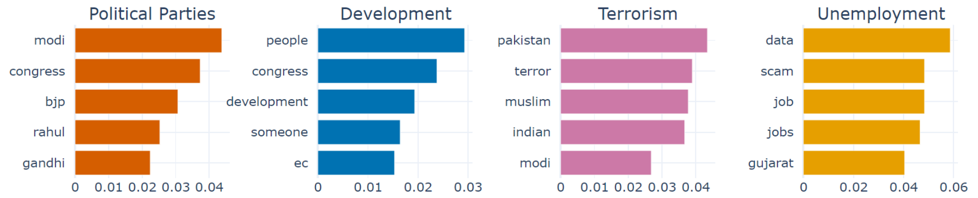


Figure 5: **Topics in the CAG category** in the 'Pre-Elections' sub-section of dataset D2. The tweets in this class see covertly aggressive debates on political parties, development, and social and security challenges such as unemployment and terrorism.

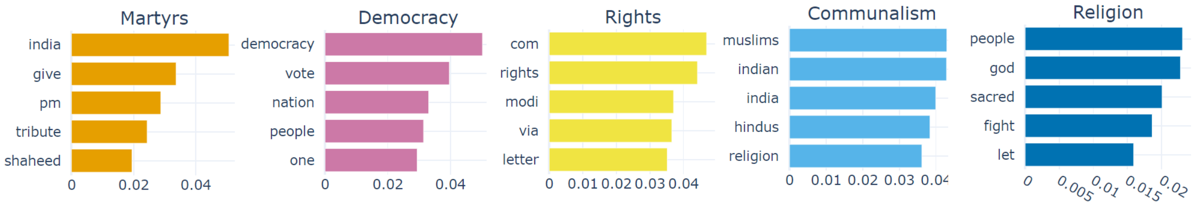


Figure 6: **Topics in the NAG category** in the 'Pre-Elections' sub-section of dataset D2. This includes tweets paying respect to the martyrs, and discussions on socio-political rights and democracy in general. It also sees non-aggressive discussions on religion and the problem of communalism.

elections data is taken into consideration. Compared to 709 out 1200 (59.33%) for "pre-elections" the ratio of aggressive tweets "post-results" comes down to 298 out of 800 (37.50%).

1. **Overtly Aggressive**: The overtly aggressive class in the after-elections category saw discussions on *political parties* in general and the *ruling party* in particular. It also saw heated debates on the issue of *religion*.
2. **Covertly Aggressive**: Tweets belonging in the covertly aggressive class include *activism*, *post-election address* made by victors, and calls to *democracy*.
3. **Not Aggressive**: Non-aggressive tweets saw *tributes* offered to veteran political leaders, and speeches made by the *ruling party* which contained mentions of *development*, *law enforcement*, and a *New India*.

Additionally, we have a full hyperparameter table which we are omitting due to space constraints; to be added to the camera-ready on acceptance. If accepted, we will try to add it to the camera-ready copy of our paper; along with the visualizations from the 'Pre-elections' and 'Post-elections' topic modeling discussed here.

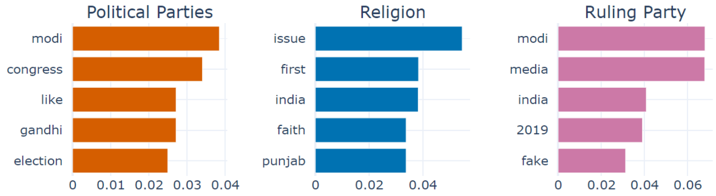


Figure 7: **Topics in the OAG category** in the ‘Post-Results’ sub-section of dataset D2. BERTopic gives clusters for tweets on the ruling party and other political parties, apart from an omniscient presence on the topic of religion.

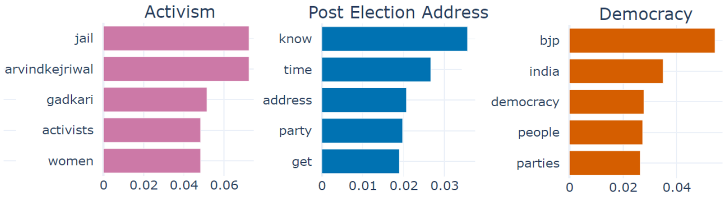


Figure 8: **Topics in the CAG category** in the ‘Post-Results’ sub-section of dataset D2. It includes tweets on activism and post-election addresses including references to democracy.

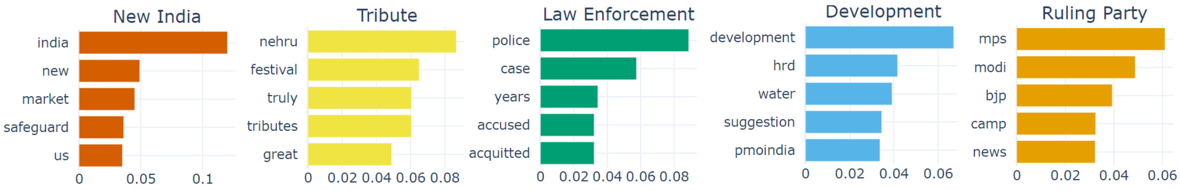


Figure 9: **Topics in the NAG category** in the ‘Post-Results’ sub-section of dataset D2. We note that the topics here are related to the ruling party that emerged victorious once again, tokens of tribute to veteran political leaders, and mentions of development, law enforcement, and a New India.

Findings of WASSA 2023 Shared Task on Empathy, Emotion and Personality Detection in Conversation and Reactions to News Articles

Valentin Barriere[1], **João Sedoc**[2], **Shabnam Tafreshi**[3], **Salvatore Giorgi**[4]
[1]Centro Nacional de Inteligencia Artificial, Santiago, Chile
[2]New York University, New York, USA
[3]ARLIS, University of Maryland, College Park, USA
[4]University of Pennsylvania, Philadelphia, USA
`valentin.barriere@cenia.cl`,`jsedoc@stern.nyu.edu`
`stafresh@umd.edu`,`sgiorgi@seas.upenn.edu`

Abstract

This paper presents the results of the WASSA 2023 shared task on predicting empathy, emotion, and personality in conversations and reactions to news articles. Participating teams were given access to a new dataset from Omitaomu et al. (2022) comprising empathic and emotional reactions to news articles. The dataset included formal and informal text, self-report data, and third-party annotations. Specifically, the dataset contained news articles (where harm is done to a person, group, or other) and crowd-sourced essays written in reaction to the article. After reacting via essays, crowd workers engaged in conversations about the news articles. Finally, the crowd workers self-reported their empathic concern and distress, personality (using the Big Five), and multi-dimensional empathy (via the Interpersonal Reactivity Index). A third-party annotated both the conversational turns (for empathy, emotion polarity, and emotion intensity) and essays (for multi-label emotions). Thus, the dataset contained outcomes (self-reported or third-party annotated) at the turn level (within conversations) and the essay level. Participation was encouraged in five tracks: (i) predicting turn-level empathy, emotion polarity, and emotion intensity in conversations, (ii) predicting state empathy and distress scores, (iii) predicting emotion categories, (iv) predicting personality, and (v) predicting multi-dimensional trait empathy. In total, 21 teams participated in the shared task. We summarize the methods and resources used by the participating teams.

1 Introduction

Affect-related phenomena have been widely studied in the last two decades (Picard, 2000). They are crucial for social interactions between humans as they create a bond between the different social agents (Cassell, 2001), whether humans or machines. They are also essential to make machines understand the world and gain common-sense knowledge, which is essential when tackling complex human-related tasks. Studying the affective and social phenomena like opinions, emotions, empathy, distress, stances, persuasiveness (Buechel et al., 2018a; Barriere and Balahur, 2023; Park et al., 2014b) or speaker traits allows machine learning practitioners to dramatically improve the response from automated agents (Pelachaud et al., 2021; Zhao et al., 2016). Social skills like empathy are essential for human(-agent) communication (Parmar et al., 2022; Reis et al., 2017). Right now, it is helpful for as many applications such as an empathic agent (Rashkin et al., 2019; Zhong et al., 2020), a way to de-bias a corpus to train a language model (Lahnala et al., 2022b), or as a tool to help human to communicate or to find consensus (Pérez-Rosas et al., 2017; Sharma et al., 2023; Argyle et al., 2023). In general, empathic utterances can be emotional; therefore, examining emotion in text-based conversations may significantly impact predicting empathy. Moreover, according to (Lahnala et al., 2022a), many studies make an amalgam between empathy and emotion by poorly defining the former. Hence, studying emotion and empathy together can help to remove this bias, even though more psycho-linguistic work would be welcome.

This paper presents the WASSA 2023 Empathy Shared Task: Predicting Empathy, Emotion, and Personality in Conversations and Reaction to News Articles, which allows studying empathy and emotion in human interactions. Past WASSA shared tasks were also held on emotion, empathy, distress, or personality detection in text essays (Tafreshi et al., 2021; Barriere et al., 2022b). Thus, this year's task builds on past shared tasks, with data very similar to past years, plus a brand new type of data. We used a new dataset from (Omitaomu et al., 2022) containing reactions to news article data and annotations similar to (Buechel et al., 2018b) and (Tafreshi et al., 2021), including news articles that express harm to an entity (e.g., individual, group

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 511–525
July 14, 2023 ©2023 Association for Computational Linguistics

of people, nature).

The news articles are accompanied by essays where authors express their empathy and distress in response to the content. Each essay is annotated for empathy and distress, and multi-label emotions. They are also enriched with additional information, such as the authors' personality traits, IRI, and demographic details, including age, gender, ethnicity, income, and education level. The new type of data introduced in this year's shared task consists in the subsequent conversations that the study participants had after writing their essays, which were annotated in perceived emotional polarity and intensity and perceived empathy. For more specific information, please refer to Section 3 in the paper.

Given this dataset as input, the shared task consists of five tracks (see Section 4 for each tracks' respective definitions of empathy and emotion):

1. Predicting Perceived Empathy and Emotion in Conversations (CONV): Teams develop models to predict several values linked to emotion and empathy for each speech turn in a textual conversation. The targets are third-party assessment of emotional polarity, emotional intensity, and empathy.
2. Predicting State Empathy (EMP): Teams develop models to predict, for each essay, empathy and distress scores quantified by Batson's empathic concern ("feeling for someone") and personal distress ("suffering with someone") (Batson et al., 1987) scales.[1]
3. Emotion Label Prediction (EMO): Teams develop models to predict, for each essay, a categorical emotion tag from the following Ekman's six basic emotions (sadness, joy, disgust, surprise, anger, or fear) (Ekman, 1971), as well as *hope* and *neutral* tag.
4. Personality Prediction (PER): Teams develop models to predict, for each essay, Big Five (OCEAN) personality traits (conscientiousness, openness, extraversion, agreeableness, emotional stability; John et al. 1999).
5. Predicting Multi-dimensional Trait Empathy (IRI): Teams develop models to predict, for each essay, multi-dimensional empathy (via the Interpersonal Reactivity Index; Davis, 1980): perspective taking, personal distress, fantasy, and empathic concern.

2 Related Work

We provide related work for each track: affect-related phenomena in interactions (Section 2.1), emotion predictions (Section 2.2), empathy and distress (Section 2.3), and personality prediction (Section 2.4).

2.1 Affective Phenomena in Interactions

Affect-related phenomena in interactions is a field of study that comprises emotion recognition in conversations (McKeown et al., 2012; Ma et al., 2020; Firdaus et al., 2020; Ringeval et al., 2013), opinion analysis in interactions (Barriere et al., 2018, 2022a), first impressions assessment (Cafaro et al., 2017), or personality detection (Mairesse and Walker, 2006) among many others. The interest of these approaches is to use the interactional context in order to model the dynamics of the target phenomena within a conversation (Hazarika et al., 2018; Majumder et al., 2019; Poria et al., 2019b,a). Recent works are using speaker-dependent vectors (Majumder et al., 2019), graph neural networks to model the interactions (Ghosal et al., 2019), or dialog-aware attention mechanism (Shen et al., 2020).

2.2 Emotion Prediction

Emotion classification is the task of predicting a single- or multi-label emotion classes (Ekman, 1971), or a value in the valence-arousal space, which has been widely studied in non-verbal language (Schuller et al., 2009; McKeown et al., 2012; Vinciarelli et al., 2008), or even in music (Soleymani et al., 2013). Emotion classification in text, more recently, has been studied thoroughly in terms of modeling, resources, and features as part of SemEval shared tasks for Affect computing and emotion classification (Strapparava and Mihalcea, 2007; Mohammad and Bravo-Marquez, 2017; Mohammad et al., 2018; Chatterjee et al., 2019; Sharma et al., 2020c).

Most emotion prediction models are learned in a supervised manner with feature engineering or continuous representation learned through pretrained language models (Peters et al., 2018; Devlin et al., 2018a), and now in an unsupervised way using emerging abilities of large language models (Choi et al., 2023; Brown et al., 2020). Acheampong et al. (2020); Murthy and Kumar (2021); Nandwani and Verma (2021); Acheampong et al. (2021); Ezzameli and Mahersia (2023) survey state-of-the-art

---

[1] *Distress* is a self-focused and negative affective state (*suffering with someone*) while *empathy* is a warm, tender, and compassionate state (*feeling for someone*).

emotion detection techniques and resources and discuss open issues in this area.

2.3 Empathy and Distress

As seen in this shared task, empathy can have varying definitions: empathic utterances, state empathy (as measured via the Batson scale), and trait empathy (as measured via the IRI), among others. Thus, research on empathy in natural language processing often uses varying or even under-specified measures (Lahnala et al., 2022a). Prior work on modeling text-based empathy focused on the empathic concern, which is to share others' emotions in the conversations (Litvak et al., 2016; Fung et al., 2016). For instance, Xiao et al. (2015, 2016); Gibson et al. (2016) modeled empathy based on the ability of a therapist to adapt to the emotions of their clients; Zhou and Jurgens (2020) quantified empathy in condolences in social media using appraisal theory; Sharma et al. (2020b) developed a model based on fine-tuning contextualized language models to predict empathy specific to mental health in text-based platforms. Guda et al. (2021) additionally utilized demographic information (e.g., education, income, age) when fine-tuning contextualized language modeling for empathy and distress prediction. While empathy is vital for human(-agent) communication, some have argued that empathy is a poor guide for moral decision-making (Bloom, 2017). To this end, recent work has shown that language associated with empathy, when separated from compassion, is more self-focused and contains negative emotions (Yaden et al., 2023).

2.4 Personality Prediction

Vora et al. (2020) and Beck and Jackson (2022) survey and analyze personality prediction models, theories, and techniques. Ji et al. (2020) review such models specifically to detect suicidal behavior. Developing personality detection models range from feature engineering methods (Bharadwaj et al., 2018; Tadesse et al., 2018) to deep learning techniques (Yang et al., 2021; Ren et al., 2021; Lynn et al., 2020). Yang et al. (2021) developed a transformer-based model to predict users' personality based on Myers-Briggs Type Indicator (Myers et al., 1985, MBTI) personality trait theory given multiple posts of the user instead of predicting personality for a single post. Ren et al. (2021) utilized deep learning techniques to develop a multi-label personality prediction and sentiment analysis model based on MBTI and Big 5 datasets. Given the cost and time needed to collect personality survey responses, Vu et al. (2020) developed methods to predict out-of-sample survey questions. More recently, Large Language Models (such as GPT-3) have been used for zero-shot personality classification (Ganesan et al., 2023).

3 Data Collection and Annotation

The source of the data for the shared task is from Omitaomu et al. (2022). We extend this dataset with essay-level emotion annotations by the authors. Although the dataset is different from the data set of Buechel et al. (2018b) used in WASSA 2021 and 2022 shared task (Tafreshi et al., 2021; Barriere et al., 2022b), it can be considered an extension. Table Table 1 shows the train, development, and test splits. We first briefly present how the original dataset was collected and annotated in subsection 3.1. We discuss the additional emotion annotation in subsection 3.2.

	Train	Dev	Test
People	41	34	65
Conversations	386	114	50
Essays	792	208	100
Speech-Turns	9,176	2,000	1,425

Table 1: Corpus statistics detailing the number of annotations.

3.1 Initial Data Collection and Annotation

Here we provide a brief overview of the data collection process employed by Omitaomu et al. (2022). They recruited crowd workers from MTurk.com and utilized the Qualtrics survey platform and ParlAI for data collection. The data collection process began with an intake phase, during which crowd workers provided their demographic information and completed surveys for the Big Five (OCEAN) personality traits and the Interpersonal Reactivity Index (IRI). Next, pairs of crowd workers read news articles. Each pair read one article of the 100 articles. After reading the article, the crowd workers wrote an essay of 300 to 800 characters about the article they read and rated their empathy and distress levels using the Batson scale. Then, the pair of crowd workers engaged in online text conversation where they were instructed to talk about the article for a minimum of 10 turns per person in training and development sets and 15 turns per person in the test set.

After the conversations were collected, a new task was created to collect turn-level annotations for each conversation. The workers were asked to rate the empathy, emotional polarity, and emotional intensity of each turn. Three crowd workers annotated each turn and were given the context of the previous turns in the conversation.

3.2 External Emotion Annotation

We enriched the dataset by annotating the essays with multi-label emotion tags. We used the six Ekman's emotions to determine whether certain basic emotions are (Ekman, 1971) more correlated with empathy and distress. We added another emotion which is hope, as it is fairly present in our dataset and used in the GoEmotion dataset as a sub-emotion of Joy, and we wanted to separate them. With the neutral label, this gave us a total of 8 label tags. Three of the four coders annotated each essay using a maximum of two emotion tags (including neutral), yielding three to six tags for each essay. We used the LEAP protocol to reach a higher agreement between the annotators (Lee et al., 2023). We calculated the inter-annotator agreement using Krippendorff's α (Krippendorff, 2013) with the MASI distance (Passonneau, 2006) that has been proven helpful for multi-label annotations, and obtained 0.40 (0.44 with Jaccard distance). We computed the ground truth by labeling all the emotions with at least two tags among the three to six possible tags. The distribution of the train and development datasets are shown in Figure 1. The matrix of co-occurrences of the train/dev sets is shown in Figure 1. Disgust is positively correlated with two emotions: anger and surprise. The highest number of co-occurrences between two emotions is 36, which happens between disgust and anger. Neutral rarely happens with other emotions. Sadness is statistically more correlated to Fear and Hope.

4 Shared Task

We set up all four tracks in CodaLab[2]. We describe each task separately in Section 4.1 and then describe dataset, resources, and evaluation metrics in Section 4.2. Note that the last four tracks are similar to the ones offered by WASSA 2022 shared task, even though this year it is possible to use the conversations to get more context.

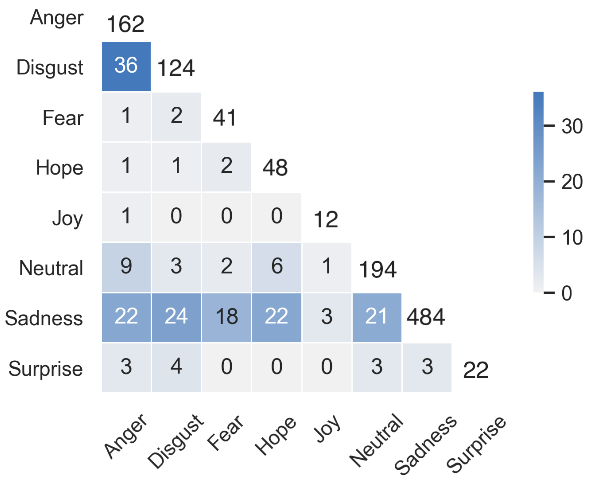


Figure 1: Co-occurrence matrix of the EMO labels on the train and dev sets

4.1 Tracks

Track 1 - Turn-level Empathy and Emotion in Conversations (CONV): The formulation of this task is to predict, for each conversational turn, the emotion polarity and intensity as well as the third party annotations of empathy. The targets are third-party assessment of emotional polarity (positive, negative, or neutral) and both emotional intensity and empathy coded on an ordinal scale from 1 to 5 with a not applicable option. This track is new to WASSA 2023.

Track 2 - State Empathy Prediction (EMP): The formulation of this task is to predict, for each essay, Batson's empathic concern ("feeling for someone") and personal distress ("suffering with someone") scores (Batson et al., 1987). Teams are expected to develop models that predict the empathy score for each essay (self-report data from the essay writer). Both empathy and distress scores are real values between 1 and 7. Empathy score is an average of 7-point scale ratings, representing each of the following states (warm, tender, sympathetic, softhearted, moved, compassionate); distress score is an average of 7-point scale ratings, representing each of the following states (worried, upset, troubled, perturbed, grieved, disturbed, alarmed, distressed). These are state measures: measures that vary within people across time. For optional use, we made personality, demographic information, and emotion labels available for each essay. This track was previously done in WASSA 2022 and 2021, but this year's task uses new data.

Track 3 - Emotion Label Prediction (EMO): The formulation of this task is to predict, for each

---

[2] `https://codalab.lisn.upsaclay.fr/competitions/11167`

essay, one or more emotion labels from the following Ekman's six basic emotions (sadness, joy, disgust, surprise, anger, or fear) (Ekman, 1971), as well as *neutral* (like in (Barriere et al., 2022b)), and we also added *hope*.

The same set of metadata that we described above was also provided for each essay in this task. Participants optionally could use this information as features to predict emotion labels. The essay-level emotion labels are third party annotations. This task was also done in WASSA 2022 and 2021, but this year's task uses new data.

Track 4 - Personality Prediction (PER): To code personality information, the Big 5 personality traits were provided, also known as the OCEAN model (Gosling et al., 2003b). In the OCEAN model, the theory identifies five factors (openness to experience, conscientiousness, extraversion, agreeableness, and neuroticism[3]). For each essay, the writer was asked to complete the Ten Item Personality Inventory (Gosling et al., 2003a). Thus, this is self-reported essay-level data. This task was previously done in WASSA 2022, but the data in this year's task (2023) is new.

Track 5 - Multi-dimensional Trait Empathy Prediction (IRI): We use the Interpersonal Reactivity Index (IRI), a measurement tool for the multidimensional assessment of empathy (Davis, 1980). The IRI consists of four subscales (Perspective Taking, Fantasy, Empathic Concern, and Personal Distress) where each subscale consists of 7 items, each on a 5-point Likert scale. The IRI is a trait-level empathy measure: a measure that is stable within people across time. Though a similar task was done in 2022, this self-reported essay-level data is new to this year's task.

Multi-task: We gave the participants a unique id for each conversation so that the participants could use multi-task learning methods to tackle all the tasks simultaneously. Moreover, speakers in the train, dev, and test datasets were given unique ids so that teams could use several of the participant's essays or conversations in order to improve the results. This was proven to help last year for the PER and IRI subtasks (Barriere et al., 2022b).

4.2 Setup

Dataset: Participants were provided the dataset described in Section 3. Participants were allowed to add the development set to the training set and submit systems trained on both. The test set was made available to the participants at the beginning of the evaluation period.

Resources and Systems Restrictions Participants were allowed to use any lexical resources (e.g., emotion or empathy dictionaries) of their choice, additional training data, or off-the-shelf emotion or empathy models. We did not put any restrictions on this shared task. We proposed several baseline models for this article, which are described in Section 4.3.

Systems Evaluation: The organizers published an evaluation script that calculates Pearson correlation for the predictions of the conversation, empathy, personality and IRI prediction tasks and precision, recall, and F1 measure for each emotion class as well as the micro and macro average for the emotion label prediction task. Pearson coefficient is the linear correlations between two variables, and it produces scores from -1 (perfectly inversely correlated) to 1 (perfectly correlated). A score of 0 indicates no correlation. The official competition metric for the empathy and emotion in conversation task (CONV) is the average of the three Pearson correlations. The official competition metric for the empathy prediction task (EMP) is the average of the two Pearson correlations. The official competition metric for the emotion evaluation is the macro F1-score, which is the harmonic mean between precision and recall. The official competition metric for the personality (resp. IRI prediction) task PER (resp. IRI) is the average of the Pearson correlations of the 5 (resp. 4) variables.

4.3 Baselines

CONV: Following Omitaomu et al. (2022), we fine-tuned a RoBERTa (base) pretrained language model (Liu et al., 2019a). The model was trained on the training set and used the development set for model validation. We trained one model for each of the turn-level label types. The training was for 50 epochs, and the model checkpoint with the best validation set Pearson correlation was kept.

EMP: Like the CONV models, we fine-tuned a RoBERTa (base) pretrained language model (Liu

---

[3] For the shared task, neuroticism has been reverse coded as emotional stability

et al., 2019a). For training, we used both the training data of the essays and the WASSA22 training data (Barriere et al., 2022b). We created separate models for empathy and distress, and used the same checkpoint and stopping criteria as the conv task models.

EMO: We created two baselines for the EMO subtask. As a first baseline, we fine-tuned a pretrained base RoBERTa transformer model (Liu et al., 2019b) over the GoEmotions dataset (Demszky et al., 2020) in a multi-label way. This lead to a macro-averaged F1 score of 0.64 on the GoEmotions test set with emotions grouped using Ekman's taxonomy, in line with the original article. We applied this model directly to the WASSA test set. This model is called Baseline$_{\text{FT}}$. As a second baseline, we fine-tuned once again this model with the essays from the training set in a multi-label way. This second model is called Baseline$_{\text{FT}}$. The pretrained models that we used were made available online using the `transformers` library (Wolf et al., 2019). We used the Adam algorithm (Kingma and Ba, 2014) with early stopping for the optimization of the training loss, using a learning rate of 10^{-5}. We followed the official partitions is both cases.

PER: We used a Big 5 personality model developed by Park et al. (2014a). This model was trained on Facebook status updates and questionnaire-based self-reported Big Five personality traits from 66,732 people. This model used ngrams and topics extracted from the Facebook status updates in an ℓ_2 penalized Ridge regression and resulted in an out-of-sample accuracy (Pearson r) of 0.43 (Openness), 0.37 (Conscientiousness), 0.42 (Extraversion), 0.35 (Agreeableness), and 0.35 (Neuroticism). This model was then applied to each essay in the test set for the shared task, producing Big 5 estimates for each.

IRI: We use the Empathic Concern model built by Giorgi et al. (2023) and train additional models for the three remaining dimensions of the IRI: Fantasy, Perspective Taking, and Personal Distress. These models were built over existing data sets where 2,805 consenting participants shared their lifetime Facebook status updates and responded to the IRI questionnaire (Abdul-Mageed et al., 2017). For each participant, we extract RoBERTa embeddings (averaging word embeddings across sentences, sentence embeddings averaged across Facebook status updates, and status embeddings averaged within participants). We used the second to last layer of the roberta-large model, producing a 1,024-dimensional vector for each participant. Using a penalized Ridge regression (with a ℓ_2 regularization strength of 10^5; tuned during nested cross-validation) in a 10-fold cross-validation resulted in a prediction accuracy (Pearson r) of 0.276 (Empathic Concern), 0.294 (Fantasy), 0.116 (Perspective Taking), and 0.291 (Personal Distress). This model was then applied to each essay in the test set for the shared task after extracting RoBERTA embeddings from each essay.

5 Results and Discussion

5.1 Empathy Prediction (CONV)

Table 2 shows the results of the track on Emotion Polarity (Emo Pol), Emotion Intensity (Emo Int) and Observed Empathy (Emp). All are regression tasks and were evaluated using Pearson correlation. The participants are ranked using the average of all three metrics. Nine teams submitted results to this track. The best system is *HIT-SCIR* obtaining the best results overall (averaged $r = .758$) but also for all the targets: an emotion polarity ($r = .852$), emotion intensity ($r = .714$) and perceived empathy ($r = .708$).

Team	Emo Pol	Emo Int	Emp	Avg
HIT-SCIR	**0.852**	**0.714**	**0.708**	**0.758**
YNU-HPCC	0.824	0.693	0.674	0.730
Hawk	0.809	0.701	0.665	0.725
NCUEE-NLP	0.803	0.698	0.669	0.724
warrior1127	0.770	0.701	0.660	0.710
CAISA	0.783	0.686	0.652	0.707
Curtin OCAI	0.750	0.683	0.573	0.669
sushantkarki	0.778	-0.030	-0.023	0.242
Cordyceps	-0.005	0.039	0.018	0.017
Baseline	0.781	0.692	0.660	0.711

Table 2: Results of the teams participating in the CONV track (Pearson correlations).

5.2 Empathy Prediction (EMP)

Table 3 shows the main results of the track on empathy (Emp) and distress (Dis) prediction. 9 teams submitted results and the best scoring system is *NCUEE-NLP* team (averaged $r = .418$). They also obtain the best separate scores for empathy and distress with respective r of .415 and .421.

Comparison with previous results: In (Buechel et al., 2018b), the best-performing system obtained r=.404 for empathy and r=.444 for distress. These

Team	Emp	Dis	Avg
NCUEE-NLP	**0.415**	**0.421**	**0.418**
CAISA	0.348	0.420	0.384
PICT-CLRL	0.358	0.334	0.346
zex	0.293	0.391	0.342
HIT-SCIR	0.329	0.354	0.342
YNU-HPCC	0.331	0.245	0.288
Curtin OCAI	0.187	0.344	0.266
Hawk	0.270	0.207	0.238
Cordyceps	-0.020	0.096	0.038
Baseline	**0.536**	**0.575**	**0.555**

Table 3: Results of the teams participating in the EMP track (Pearson correlations).

results were achieved only on the training set using ten-fold cross-validation experiments, which is not comparable to the results in this shared task. In WASSA 2021 and 2022 (Tafreshi et al., 2021; Barriere et al., 2022b), that had the largest training sets, the best scoring systems reached an averaged *r* of .545 and .540. These past scores are in line with the one of the baseline that we proposed, which was trained also using the past years' datasets and gives far better performances than the systems of this year's participants (average *r*=.555 compared to *r*=.418).

5.3 Emotion Recognition (EMO)

Table 4 presents the results for 13 teams for emotion prediction models. The best-performing system in terms of Macro F1 (70.1%) as well as micro-Jaccard (60.1%) is *Adityapatkar* which is significantly higher than the remaining emotion prediction models. To get more insight, we also provide a breakdown of the results by emotion class in Table 7. Fear was easily predicted by the majority of the participant's systems, as per the neutral and sadness classes that are the most present in the dataset. The results are very heterogenous among the participants in the breakdown for all emotion labels. The emotion model submitted by team *LingJing* outperforms the other models on Disgust, while team *andeldiko* performs best on Anger.

5.4 Personality and Interpersonal Reactivity Prediction (PER/IRI)

The results of the tracks on personality and IRI predictions are presented in Table 5. Five and six teams submitted results to respectively the PER and IRI subtasks. The best scoring system for both tasks is the one of *YNU-HPCC*. For the PER task, like last year, participants obtained negative correlations in all the tasks. There were four submitted systems with negative correlations for Extraversion and three with negative correlations for Consciousness prediction. For the IRI task, the best results on the different dimensions were distributed over the different teams and the baseline: team *CAISA* obtained the best r for the *perspective taking*, team *Xuao* for the *personal distress*, team *Hawk* for the *fantasy* and our baseline for the *empathic concern*.

Team	P	R	F1	Jac
Adityapatkar	**0.810**	0.677	**0.701**	**0.600**
Bias Busters	0.630	**0.731**	0.647	0.538
HIT-SCIR	0.721	0.631	0.644	0.562
zex	0.699	0.637	0.643	0.562
lazyboy.blk	0.776	0.601	0.613	0.554
Converge	0.596	0.560	0.565	0.539
amsqr	0.752	0.479	0.533	0.507
surajtc	0.463	0.668	0.522	0.451
YNU-HPCC	0.575	0.502	0.514	0.542
VISU	0.257	0.301	0.272	0.421
Cordyceps	0.191	0.236	0.202	0.241
Sidshank	0.295	0.211	0.150	0.287
mimmu3302	0.092	0.200	0.126	0.271
Baseline$_{\text{FT}}$	0.631	0.645	0.632	0.551
Baseline$_{\text{EXT}}$	0.860	0.539	0.602	0.522

Table 4: Results of the teams participating in the EMO track (macro-averaged precision (P), recall (R), F1-score (F1) and micro-Jaccard (Jac)).

6 Overview of Submitted Systems

A total of 21 teams participated in the shared tasks, with 9, 9, 13, 5, and 6 teams participating for the five tracks, respectively. In this section, we provide a summary of the machine learning algorithms and resources that were used by the teams.

6.1 Machine Learning Architectures and Resources

Architectures: The majority of the proposed systems are based on neural networks architectures and transformers (Vaswani et al., 2017). Approaches include classical models like BERT (Devlin et al., 2018b), RoBERTa (Liu et al., 2019b), DeBERTaV2 and DeBERTaV3 (He et al., 2020, 2021), ELECTRA (Clark et al., 2020), ALBERT (Lan et al., 2020), DistilBERT (Sanh et al., 2019), SimCSE (Gao et al., 2021), and also transformers for long texts like BigBird and LongFormer (Zaheer et al., 2020; Beltagy et al., 2020). Some of them trained their model using Parameter Efficient Fine Tuning methods like Low-Rank Adapters

Team	Consc.	Open.	Extr.	Agree.	Stab.	**PER**	Persp.	Distr.	Fant.	Emp.	**IRI**
YNU-HPCC	0.289	**0.372**	-0.130	**0.410**	**0.317**	**0.252**	0.102	0.256	0.033	**0.226**	**0.154**
Xuhao	∅	∅	∅	∅	∅	∅	0.132	**0.366**	0.036	0.076	0.153
CAISA	**0.323**	0.327	-0.197	0.290	0.256	0.200	**0.158**	-0.188	-0.056	0.180	0.024
Curtin OCAI	0.186	0.152	0.014	-0.038	0.183	0.099	-0.092	0.193	-0.014	-0.114	-0.007
Cordyceps	-0.059	-0.187	**0.160**	0.101	-0.010	0.001	0.004	0.191	-0.018	0.089	0.067
Hawk	-0.082	0.066	-0.109	-0.119	-0.114	-0.072	-0.013	-0.020	**0.138**	-0.153	-0.012
Baselines	-0.131	-0.037	-0.134	0.195	0.081	-0.005	0.107	-0.046	0.063	**0.340**	0.116

Table 5: Results of the teams participating in the PER/IRI tracks (Pearson correlations).

(Hu et al., 2021) or AdapterHub (Pfeiffer et al., 2020). Only two of the submitted systems used an interaction-aware model. The first one is based on Kim and Vossen (2021), which is able to learn intra- and inter-speaker states and context to predict the emotion of a current speaker. The second one is a RoBERTa transformer using a context window containing past and future utterances. One team used Large Language Models with GPT3 (Brown et al., 2020) and GPT4 (OpenAI, 2023) with in-context learning but also by fine-tuning them. One team used bidirectional LSTM (Hochreiter and Schmidhuber, 1997) with classical text embeddings like Glove (Pennington et al., 2014), Fastext (Bojanowski et al., 2016) and Flair (Akbik et al., 2018). Finally, one team proposed to use a Multinomial Naive Bayes.

Two systems proposed integrating the writer's metadata using in-context learning, one by rewriting the sentences with natural language templates and another by prompting the table.

Resources: Two teams used RoBERTa transformers that were already fine-tuned for sentiment and emotion tasks before fine-tuning them on the data. These models were trained on nearly 58M tweets and fine-tuned for sentiment analysis and emotion recognition using the TweetEval benchmark (Barbieri et al., 2020). One team used the Epitome empathy dataset of (Sharma et al., 2020a) composed of support-seeker and responder posts on Reddit in order to pre-train the weights of their adapter layers. Finally, one team used an interaction-aware model trained on emotion recognition in conversations (Kim and Vossen, 2021).

Others: Three teams used data augmentation to create new examples: two by paraphrasing the under-represented classes using a T5 (Raffel et al., 2019), and one by generating examples of the under-represented classes with a GPT-4. One team used a FLAN-T5 model (Chung et al., 2022) to summarize the long articles in order to reduce the number of tokens used as input to their classifier. Four teams used ensemble methods, which are classics for coding competitions.

ML Alg.	# of team	CONV	EMP	EMO	PER/IRI
BERT-like	11	✓	✓	✓	✓
Ensemble	4	✓	✓	✓	
Data-Aug.	3	✓	✓	✓	✓
Adapters	2	✓	✓	✓	✓
LLM	1			✓	
biLSTM	1			✓	
Naive Bayes	1			✓	

Table 6: Algorithms used by the different teams. We listed all the techniques that teams reported in their system description papers.

7 Conclusion

In this paper, we presented the shared task on empathy, emotion, and personality detection in essays and conversations in reactions to news articles, to which 21 teams participated and 12 submitted a paper. Like last year, neural models are the major parts of the submissions, especially transformer models. The systems obtaining the best results for the five subtasks relied on BERT, RoBERTa, and DeBERTa models. Nobody used task-related features extracted from lexicons, as was the case in the previous editions. External data still helps improve the results, like leveraging Emotion, Sentiment, and Empathy external datasets. Nevertheless, more is needed to make the systems competitive enough to beat fine-tuned bigger models like the biggest DeBERTa (1.3B) used by the winning teams of the CONV, PER, and IRI subtasks. Likewise, using a finely crafted model for interactions cannot compete with a model 10 times its size, using a simple window to integrate context. Finally, some participants used features from a track to give more context, but no approach has considered using multi-task learning between the tracks, even though it was possible to do it. Surprisingly, no teams used the identifier of the speaker to integrate their conversations in order to get more context to find the empathy or emotion of the essay.

Limitations

The test dataset size makes it difficult to draw meaningful conclusions for the Tracks 2 to 5. Similarly, the text data associated with this task (i.e., reaction essays) may make it difficult to infer person-level traits using preexisting models, which may be trained on other domains of text (e.g., social media data). This could explain the negative correlations with extraversion and conscientiousness in Table 5 for the baseline model. Finally, annotating text for emotions and perceived empathy are difficult, subjective tasks. Often statements in the essays are ambiguous and could be interpreted in various ways, especially considering the fact that these are written essay and void of speech cues and the body language of the speaker. Thus, the third-party annotators' own reactions to the news articles could influence how the reaction essays are perceived (e.g., interpreting reactions to the Syrian civil war may depend on the political beliefs of the annotator). Therefore, cultural and social biases may be present in the third party annotations.

Ethics Statement

The main ethical concern is the possibility of misuse of the data and models for manipulation of others. For example, models could be used to produce political ads which elicit empathetic responses which further influence voting or donations. Models could be used to deploy malicious bots on social media platforms (Giorgi et al., 2021), design public health messages (which could be especially problematic around sensitive topics such as vaccines), or spread misinformation (Himelein-Wachowiak et al., 2021). More information are available in the original dataset article (Omitaomu et al., 2022).

Acknowledgements

V.B. has been funded by the grant National Center for Artificial Intelligence CENIA FB210017, Basal ANID.

References

Muhammad Abdul-Mageed, Anneke Buffone, Hao Peng, Salvatore Giorgi, Johannes C Eichstaedt, and Lyle H Ungar. 2017. Recognizing pathogenic empathy in social media. In *ICWSM*, pages 448–451.

Francisca Adoma Acheampong, Henry Nunoo-Mensah, and Wenyu Chen. 2021. Transformer models for text-based emotion detection: a review of bert-based approaches. *Artificial Intelligence Review*, 54(8):5789–5829.

Francisca Adoma Acheampong, Chen Wenyu, and Henry Nunoo-Mensah. 2020. Text-based emotion detection: Advances, challenges, and opportunities. *Engineering Reports*, 2(7):e12189.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual String Embeddings for Sequence Labeling. In *Coling*.

Lisa P. Argyle, Ethan Busby, Joshua Gubler, Chris Bail, Thomas Howe, Christopher Rytting, and David Wingate. 2023. AI Chat Assistants can Improve Conversations about Divisive Topics. *ArXiv*.

Francesco Barbieri, Jose Camacho-collados, and Leonardo Neves Luis Espinosa-anke. 2020. TWEETEVAL : Unified Benchmark and Comparative Evaluation for Tweet Classification. pages 1644–1650.

Valentin Barriere and Alexandra Balahur. 2023. Multilingual Multi-target Stance Recognition in Online Public Consultations. *accepted to MDPI Mathematics*.

Valentin Barriere, Chloe Clavel, and Slim Essid. 2018. Attitude Classification in Adjacency Pairs of a Human-Agent Interaction with Hidden Conditional Random Fields. In *ICASSP*.

Valentin Barriere, Slim Essid, and Chloé Clavel. 2022a. Opinions in interactions : New annotations of the SEMAINE database. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 7049–7055, Marseille, France. European Language Resources Association.

Valentin Barriere, Shabnam Tafreshi, João Sedoc, and Sawsan Alqahtani. 2022b. WASSA 2022 shared task: Predicting empathy, emotion and personality in reaction to news stories. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 214–227, Dublin, Ireland. Association for Computational Linguistics.

C Daniel Batson, Jim Fultz, and Patricia A Schoenrade. 1987. Distress and empathy: Two qualitatively distinct vicarious emotions with different motivational consequences. *Journal of personality*, 55(1):19–39.

Emorie D Beck and Joshua J Jackson. 2022. A mega-analysis of personality prediction: Robustness and boundary conditions. *Journal of Personality and Social Psychology*, 122(3):523.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The Long-Document Transformer.

Srilakshmi Bharadwaj, Srinidhi Sridhar, Rahul Choudhary, and Ramamoorthy Srinath. 2018. Persona traits identification based on myers-briggs type indicator (mbti)-a text classification approach. In *2018 international conference on advances in computing,* *communications and informatics (ICACCI)*, pages 1076–1082. IEEE.

Paul Bloom. 2017. *Against empathy: The case for rational compassion*. Random House.

Piotr Bojanowski, Edouard Grave, Armand Joulin, Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Tomas Mikolov, Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, Samuel J. Gershman, and Jürgen Schmidhuber. 2016. Bag of Tricks for Efficient Text Classification. *arXiv:1604.00289v1[cs.AI]*, pages 1–55.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners.

Sven Buechel, Anneke Buffone, Barry Slaff, Lyle Ungar, and João Sedoc. 2018a. Modeling empathy and distress in reaction to news stories. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, (2017):4758–4765.

Sven Buechel, Anneke Buffone, Barry Slaff, Lyle Ungar, and João Sedoc. 2018b. Modeling empathy and distress in reaction to news stories. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4758–4765.

Angelo Cafaro, Johannes Wagner, Tobias Baur, Soumia Dermouche, Mercedes Torres Torres, Catherine Pelachaud, Elisabeth André, and Michel Valstar. 2017. The NoXi database: multimodal recordings of mediated novice-expert interactions. In *ICMI*.

Justine Cassell. 2001. Embodied conversational agents: Representation and intelligence in user interfaces. *AI Magazine*, 22(4):67–83.

Ankush Chatterjee, Kedhar Nath Narahari, Meghana Joshi, and Puneet Agrawal. 2019. SemEval-2019 task 3: EmoContext contextual emotion detection in text. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 39–48, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Minje Choi, Jiaxin Pei, Sagar Kumar, Chang Shu, and David Jurgens. 2023. Do LLMs Understand Social Knowledge? Evaluating the Sociability of Large Language Models with SocKET Benchmark. (1).

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H Chi, Jeff Dean, Jacob Devlin, Adam Robert, Denny Zhou, Quoc V Le, and Jason Wei. 2022. Scaling Instruction-Finetuned Language Models.

Kevin Clark, Minh-Thang Luong, Quoc Le, and Christopher D. Manning. 2020. Pre-Training Transformers as Energy-Based Cloze Models. pages 285–294.

Mark H Davis. 1980. *Interpersonal Reactivity Index*. Edwin Mellen Press.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A Dataset of Fine-Grained Emotions. pages 4040–4054.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018a. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jacob Devlin, Ming-wei Chang, Kenton Lee, and Kristina Toutanova. 2018b. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

Paul Ekman. 1971. Universals and cultural differences in facial expressions of emotion. In *Nebraska symposium on motivation*. University of Nebraska Press.

K Ezzameli and H Mahersia. 2023. Emotion recognition from unimodal to multimodal analysis: A review. *Information Fusion*, page 101847.

Mauajama Firdaus, Hardik Chauhan, Asif Ekbal, and Pushpak Bhattacharyya. 2020. MEISD : A Multimodal Multi-Label Emotion , Intensity and Sentiment Dialogue Dataset for Emotion Recognition and Sentiment Analysis in Conversations. In *COLING*, pages 4441–4453.

Pascale Fung, Dario Bertero, Yan Wan, Anik Dey, Ricky Ho Yin Chan, Farhad Bin Siddique, Yang Yang, Chien-Sheng Wu, and Ruixi Lin. 2016. Towards empathetic human-robot interactions. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 173–193. Springer.

Adithya V Ganesan, Yash Kumar Lal, August Håkan Nilsson, and H. Andrew Schwartz. 2023. Systematic evaluation of gpt-3 for zero-shot personality estimation. In *Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple Contrastive Learning of Sentence Embeddings. *EMNLP 2021 - 2021 Conference on Empirical Methods in Natural Language Processing, Proceedings*, pages 6894–6910.

Deepanway Ghosal, Navonil Majumder, Soujanya Poria, Niyati Chhaya, and Alexander Gelbukh. 2019. DialogueGCN: A graph convolutional neural network for emotion recognition in conversation. *arXiv*, 2.

James Gibson, Dogan Can, Bo Xiao, Zac E Imel, David C Atkins, Panayiotis Georgiou, and Shrikanth Narayanan. 2016. A deep learning approach to modeling empathy in addiction counseling. *Commitment*, 111:21.

Salvatore Giorgi, Shreya Havaldar, Farhan Ahmed, Zuhaib Akhtar, Shalaka Vaidya, Gary Pan, Lyle H Ungar, H Andrew Schwartz, and Joao Sedoc. 2023. Human-centered metrics for dialog system evaluation. *arXiv preprint arXiv:2305.14757*.

Salvatore Giorgi, Lyle Ungar, and H. Andrew Schwartz. 2021. Characterizing social spambots by their human traits. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 5148–5158, Online. Association for Computational Linguistics.

Samuel D Gosling, Peter J Rentfrow, and William B Swann Jr. 2003a. A very brief measure of the big-five personality domains. *Journal of Research in personality*, 37(6):504–528.

Samuel D Gosling, Peter J Rentfrow, and Williams B Swann Jr. 2003b. A very brief measure of the big-five personality domains. *Journal of Research in Personality*, 37:504–528.

Bhanu Prakash Reddy Guda, Aparna Garimella, and Niyati Chhaya. 2021. Empathbert: A bert-based framework for demographic-aware empathy prediction. *arXiv preprint arXiv:2102.00272*.

Devamanyu Hazarika, Soujanya Poria, Rada Mihalcea, Erik Cambria, and Roger Zimmermann. 2018. ICON : Interactive Conversational Memory Network for Multimodal Emotion Detection. In *EMNLP*, pages 2594–2604.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. pages 1–17.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. *arXiv*, pages 1–21.

McKenzie Himelein-Wachowiak, Salvatore Giorgi, Amanda Devoto, Muhammad Rahman, Lyle Ungar, H Andrew Schwartz, David H Epstein, Lorenzo Leggio, and Brenda Curtis. 2021. Bots and misinformation spread on social media: Implications for covid-19. *Journal of medical Internet research*, 23(5):e26933.

Sepp Hochreiter and Jurgen Schmidhuber. 1997. LONG SHORT-TERM MEMORY. *Neural Computation*, 9(8):1735–1780.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Shaoxiong Ji, Shirui Pan, Xue Li, Erik Cambria, Guodong Long, and Zi Huang. 2020. Suicidal ideation detection: A review of machine learning methods and applications. *IEEE Transactions on Computational Social Systems*, 8(1):214–226.

Oliver P John, Sanjay Srivastava, et al. 1999. *The Big-Five trait taxonomy: History, measurement, and theoretical perspectives*, volume 2. University of California Berkeley.

Taewoon Kim and Piek Vossen. 2021. EmoBERTa: Speaker-Aware Emotion Recognition in Conversation with RoBERTa.

Diederik Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations*, pages 1–13.

Klaus Krippendorff. 2013. Content Analysis: An Introduction to Its Methodology. In *Content Analysis: An Introduction to Its Methodology*.

Allison Lahnala, Charles Welch, David Jurgens, and Lucie Flek. 2022a. A Critical Reflection and Forward Perspective on Empathy and Natural Language Processing. *Findings of the Association for Computational Linguistics: EMNLP 2022*, (3):2139–2158.

Allison Lahnala, Charles Welch, Béla Neuendorf, and Lucie Flek. 2022b. Mitigating toxic degeneration with empathetic data: Exploring the relationship between toxicity and empathy. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4926–4938, Seattle, United States. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Kevin Gimpel, Sebastian Goodman, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A LITE BERT FOR SELF-SUPERVISED LEARNING OF LANGUAGE REPRESENTATIONS. In *ICLR*.

Seunggun Lee, Alexandra DeLucia, Nikita Nangia, Praneeth S. Ganedi, Ryan Min Guan, Rubing Li, Britney A. Ngaw, Aditya Singhal, Shalaka Vaidya, Zijun Yuan, Lining Zhang, and João Sedoc. 2023. Common law annotations: Investigating the stability of dialog system output annotations. In *Findings of the Association for Computational Linguistics: ACL 2023*, Toronto, Canada. Association for Computational Linguistics.

Marina Litvak, Jahna Otterbacher, Chee Siang Ang, and David Atkins. 2016. Social and linguistic behavior and its correlation to trait empathy. In *Proceedings* *of the Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media (PEOPLES)*, pages 128–137.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019a. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. RoBERTa: A Robustly Optimized BERT Pretraining Approach. (1).

Veronica Lynn, Niranjan Balasubramanian, and H. Andrew Schwartz. 2020. Hierarchical modeling for user personality prediction: The role of message-level attention. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5306–5316, Online. Association for Computational Linguistics.

Yukun Ma, Khanh Linh Nguyen, Frank Z. Xing, and Erik Cambria. 2020. A survey on empathetic dialogue systems. *Information Fusion*, 64(May):50–70.

François Mairesse and Marilyn a. Walker. 2006. Automatic recognition of personality in conversation. *Proceedings of the Human Language Technology Conference of the NAACL*, (June):85–88.

Navonil Majumder, Soujanya Poria, Devamanyu Hazarika, Rada Mihalcea, Alexander Gelbukh, and Erik Cambria. 2019. DialogueRNN: An Attentive RNN for Emotion Detection in Conversations. In *AAAI*.

Gary McKeown, Michel Valstar, Roddy Cowie, Maja Pantic, and Marc Schröder. 2012. The SEMAINE database: Annotated multimodal records of emotionally colored conversations between a person and a limited agent. *IEEE Transactions on Affective Computing*, 3(1):5–17.

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. Semeval-2018 task 1: Affect in tweets. In *Proceedings of the 12th international workshop on semantic evaluation*, pages 1–17.

Saif M Mohammad and Felipe Bravo-Marquez. 2017. Wassa-2017 shared task on emotion intensity. *arXiv preprint arXiv:1708.03700*.

Ashritha R Murthy and KM Anil Kumar. 2021. A review of different approaches for detecting emotion from text. In *IOP Conference Series: Materials Science and Engineering*, volume 1110, page 012009. IOP Publishing.

Isabel Briggs Myers, Mary H McCaulley, and Robert Most. 1985. *Manual, a guide to the development and use of the Myers-Briggs type indicator*. consulting psychologists press.

Pansy Nandwani and Rupali Verma. 2021. A review on sentiment analysis and emotion detection from text. *Social Network Analysis and Mining*, 11(1):1–19.

Damilola Omitaomu, Shabnam Tafreshi, Tingting Liu, Sven Buechel, Chris Callison-Burch, Johannes Eichstaedt, Lyle Ungar, and João Sedoc. 2022. Empathic Conversations: A Multi-level Dataset of Contextualized Conversations.

OpenAI. 2023. GPT-4 Technical Report. 4:1–100.

Gregory Park, H. Andrew Schwartz, Johannes C. Eichstaedt, Margaret L Kern, Michal Kosinski, David J. Stillwell, Lyle H. Ungar, and Martin E.P. Seligman. 2014a. Automatic personality assessment through social media language. *Journal of Personality and Social Psychology*.

Sunghyun Park, Han Suk Shim, Moitreya Chatterjee, Kenji Sagae, and Louis-Philippe Morency. 2014b. Computational Analysis of Persuasiveness in Social Multimedia: A Novel Dataset and Multimodal Prediction Approach. *Proceedings of the 16th International Conference on Multimodal Interaction - ICMI '14*, pages 50–57.

Dhaval Parmar, Stefan Olafsson, Dina Utami, Prasanth Murali, and Timothy Bickmore. 2022. Designing empathic virtual agents: manipulating animation, voice, rendering, and empathy to create persuasive agents. *Autonomous Agents and Multi-Agent Systems*, 36(1):17.

Rebecca Passonneau. 2006. Measuring agreement on set-valued items (MASI) for semantic and pragmatic annotation. *Proceedings of the 5th International Conference on Language Resources and Evaluation, LREC 2006*, pages 831–836.

Catherine Pelachaud, Carlos Busso, and Dirk Heylen. 2021. Multimodal Behavior Modeling for Socially Interactive Agents. *The Handbook on Socially Interactive Agents*, 1:259–310.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. GloVe: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.

Verónica Pérez-Rosas, Rada Mihalcea, Kenneth Resnicow, Satinder Singh, and Lawrence An. 2017. Understanding and predicting empathic behavior in counseling therapy. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1426–1435, Vancouver, Canada. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulic, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. AdapterHub : A Framework for Adapting Transformers. *arXiv preprint arXiv:2007.07779*.

Rosalind W Picard. 2000. *Affective computing*. MIT press.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2019a. MELD: A Multimodal Multi-Party Dataset for Emotion Recognition in Conversations. In *ACL*, pages 527–536.

Soujanya Poria, Navonil Majumder, Rada Mihalcea, and Eduard Hovy. 2019b. Emotion Recognition in Conversation: Research Challenges, Datasets, and Recent Advances. pages 1–10.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. pages 1–53.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5370–5381, Florence, Italy. Association for Computational Linguistics.

Harry T Reis, Edward P Lemay Jr, and Catrin Finkenauer. 2017. Toward understanding understanding: The importance of feeling understood in relationships. *Social and Personality Psychology Compass*, 11(3):e12308.

Zhancheng Ren, Qiang Shen, Xiaolei Diao, and Hao Xu. 2021. A sentiment-aware deep learning approach for personality detection from text. *Information Processing & Management*, 58(3):102532.

Fabien Ringeval, Andreas Sonderegger, Juergen Sauer, and Denis Lalanne. 2013. Introducing the RECOLA multimodal corpus of remote collaborative and affective interactions. In *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition, FG 2013*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv*, pages 2–6.

Björn Schuller, Stefan Steidl, and Anton Batliner. 2009. The INTERSPEECH 2009 emotion challenge. In *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, pages 312–315.

Ashish Sharma, Inna W. Lin, Adam S. Miner, David C. Atkins, and Tim Althoff. 2023. Human–AI collaboration enables more empathic conversations in text-based peer-to-peer mental health support. *Nature Machine Intelligence*, 5(1):46–57.

Ashish Sharma, Adam S. Miner, David C. Atkins, and Tim Althoff. 2020a. A computational approach to understanding empathy expressed in text-based mental health support WARNING: This paper contains content related to suicide and self-harm. *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pages 5263–5276.

Ashish Sharma, Adam S Miner, David C Atkins, and Tim Althoff. 2020b. A computational approach to understanding empathy expressed in text-based mental health support. *arXiv preprint arXiv:2009.08441*.

Chhavi Sharma, Deepesh Bhageria, William Scott, Srinivas PYKL, Amitava Das, Tanmoy Chakraborty, Viswanath Pulabaigari, and Bjorn Gamback. 2020c. Semeval-2020 task 8: Memotion analysis–the visuo-lingual metaphor! *arXiv preprint arXiv:2008.03781*.

Weizhou Shen, Junqing Chen, Xiaojun Quan, and Zhixian Xie. 2020. DialogXL: All-in-One XLNet for Multi-Party Conversation Emotion Recognition.

M. Soleymani, M. N. Caro, E. M. Schmidt, and Y. H. Yang. 2013. The MediaEval 2013 brave new task: Emotion in music. In *Working Notes Proceedings of the MediaEval 2015 Workshop*, volume 1043.

Carlo Strapparava and Rada Mihalcea. 2007. Semeval-2007 task 14: Affective text. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 70–74.

Michael M Tadesse, Hongfei Lin, Bo Xu, and Liang Yang. 2018. Personality predictions based on user behavior on the facebook social media platform. *IEEE Access*, 6:61959–61969.

Shabnam Tafreshi, Orphee De Clercq, Valentin Barriere, Sven Buechel, João Sedoc, and Alexandra Balahur. 2021. WASSA 2021 shared task: Predicting empathy and emotion in reaction to news stories. In *Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 92–104, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *arXiv:1706.03762 [cs]*.

Alessandro Vinciarelli, Maja Pantic, Hervé Bourlard, and Alex Pentland. 2008. Social signals, their function, and automatic analysis. *Proceedings of the 10th international conference on Multimodal interfaces - IMCI '08*, page 61.

Hetal Vora, Mamta Bhamare, and Dr K Ashok Kumar. 2020. Personality prediction from social media text: An overview. *Int. J. Eng. Res*, 9(05):352–357.

Huy Vu, Suhaib Abdurahman, Sudeep Bhatia, and Lyle Ungar. 2020. Predicting responses to psychological questionnaires from participants' social media posts and question text embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1512–1524.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. HuggingFace's Transformers: State-of-the-art Natural Language Processing.

Bo Xiao, Chewei Huang, Zac E Imel, David C Atkins, Panayiotis Georgiou, and Shrikanth S Narayanan. 2016. A technology prototype system for rating therapist empathy from audio recordings in addiction counseling. *PeerJ Computer Science*, 2:e59.

Bo Xiao, Zac E Imel, Panayiotis G Georgiou, David C Atkins, and Shrikanth S Narayanan. 2015. " rate my therapist": automated detection of empathy in drug and alcohol counseling via speech and language processing. *PloS one*, 10(12):e0143055.

David B Yaden, Salvatore Giorgi, Matthew Jordan, Anneke Buffone, Johannes C Eichstaedt, H Andrew Schwartz, Lyle Ungar, and Paul Bloom. 2023. Characterizing empathy and compassion using computational linguistic analysis. *Emotion*.

Feifan Yang, Xiaojun Quan, Yunyi Yang, and Jianxing Yu. 2021. Multi-document transformer for personality detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14221–14229.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Others. 2020. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297.

Ran Zhao, Tanmay Sinha, Alan W Black, and Justine Cassell. 2016. Socially-aware virtual agents: Automatically assessing dyadic rapport from temporal patterns of behavior. In *Proceedings of Intelligent Virtual Agents (IVA 2016).*, volume 10011 LNAI.

Peixiang Zhong, Chen Zhang, Hao Wang, Yong Liu, and Chunyan Miao. 2020. Towards persona-based empathetic conversational models. *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pages 6556–6566.

Naitian Zhou and David Jurgens. 2020. Condolence and empathy in online communities. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

A Emotion-level scores

The scores of the participants' systems by emotion label are visible in Table 7. The classes *hope* and *surprise* are absent from the test set.

	Anger			Disgust			Fear			Neutral			Sadness		
Team	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Adityapatkar	86	25	39	67	40	**50**	100	100	**100**	68	89	77	85	85	**85**
Bias Busters	83	42	**56**	22	40	28	83	100	91	53	94	68	73	89	80
HIT-SCIR	88	28	44	25	20	22	100	100	**100**	67	86	76	80	80	80
zex	78	28	42	25	20	22	100	100	**100**	68	89	77	79	80	80
lazyboy.blk	100	12	22	33	20	25	100	100	**100**	67	92	**78**	88	76	81
Converge	50	25	33	0	0	0	100	80	89	69	86	77	79	89	84
amsqr	100	21	34	33	20	25	100	40	56	64	81	72	78	78	78
surajtc	50	28	37	6	20	9	56	100	71	51	100	68	68	85	76
YNU-HPCC	62	21	31	0	0	0	75	60	67	67	83	74	83	87	**85**
VISU	0	0	0	0	0	0	0	0	0	63	53	57	65	98	78
Cordyceps	14	4	6	0	0	0	0	0	0	37	53	43	44	61	51
Sidshank	0	0	0	0	0	0	0	0	0	100	6	11	47	100	64
mimmu3302	0	0	0	0	0	0	0	0	0	0	0	0	46	100	63
Baseline	56	54	55	14	20	17	100	80	89	67	83	74	78	85	81

Table 7: Emotion-level participants performances

YNU-HPCC at WASSA-2023 Shared Task 1: Large-scale Language Model with LoRA Fine-Tuning for Empathy Detection and Emotion Classification

Yukun Wang, Jin Wang and Xuejie Zhang
School of Information Science and Engineering
Yunnan University
Kunming, China
`wangyukun@mail.ynu.edu.cn, {wangjin, xjzhang}@ynu.edu.cn`

Abstract

This paper describes the system for the YNU-HPCC team in WASSA-2023 Shared Task 1: Empathy Detection and Emotion Classification. This task needs to predict the empathy, emotion, and personality of the empathic reactions. This system is mainly based on the Decoding-enhanced BERT with disentangled attention (DeBERTa) model with parameter-efficient fine-tuning (PEFT) and the Robustly Optimized BERT Pretraining Approach (RoBERTa). Low-Rank Adaptation (LoRA) fine-tuning in PEFT is used to reduce the training parameters of large language models. Moreover, back translation is introduced to augment the training dataset. This system achieved relatively good results on the competition's official leaderboard. The code of this system is available here.

1 Introduction

The purpose of WASSA-2023 Shared Task 1 (Barriere et al., 2023) is to use empathic reaction data to predict hidden sentiment and personality. This task consisted of five tracks:

- **Track 1:** Empathy and Emotion Prediction in Conversations (CONV), which consists in predicting empathy, emotion polarity, and emotional intensity in a conversation;
- **Track 2:** Empathy Prediction (EMP), which consists in predicting empathy, and personal distress in an essay;
- **Track 3:** Emotion Classification (EMO), which consists in predicting the emotion in an essay;
- **Track 4:** Personality Prediction (PER), which consists in predicting the conscientiousness, openness, extraversion, agreeableness, and stability of the essay writer;
- **Track 5:** Interpersonal Reactivity Index Prediction (IRI), which consists in predicting perspective-taking, personal distress, fantasy, and empathetic concern of the essay writer;

Although the prediction goals are different, all five tracks can be considered as either a sentiment classification (Peng et al., 2020) or regression task (Kong et al., 2022). One of the biggest challenges in this task lies in how to learn representation for the given text. The early exploration was based on text similarity (Jijkoun and Rijke, 2005) or text alignment (de Marneffe et al., 2008). With the development of neural networks, convolutional neural networks (CNN) (Kim, 2014) and recurrent neural networks (RNN) (Zaremba et al., 2014) and their variants are adopted to learn text representations. Both CNN and RNN are shallow models, which only incorporate previous knowledge in the first layer of the model. The models are also based on word embeddings that are useful in only capturing the semantic meaning of words without understanding higher-level concepts like anaphora, long-term dependencies, and many more.

Beyond word embeddings, recent studies proposed embedding from language models (ELMo), which can learn word embeddings by incorporating both word-level characteristics as well as contextual semantics (Zhang et al., 2021). This also led to the emergence of pre-trained models (PLM) using Transformers as basic units. The PLMs, such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), ALBERT (Lan et al., 2019), and DeBERTa (He et al., 2020), are first fed a large amount of unannotated data, allowing the model to learn the usage of various words and how the language is written in general. Then, they can be finetuned to be transferred to a Natural Language Processing (NLP) task where it is fed another smaller task-specific dataset. As the scale of PLMs increases, the model performance in downstream tasks becomes better and better. Nevertheless, the fine-

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 526–530
July 14, 2023 ©2023 Association for Computational Linguistics

tuning procedure brings about increased requirements for model training costs. For example, the large sequence-to-sequence model GPT-3 has 175B parameters (Brown et al., 2020). To reduce training costs, recent studies suggest using parameter-efficient fine-tuning (PEFT) (Houlsby et al., 2019) to enable the efficient adaption of PLMs to downstream applications without fine-tuning all the parameters of the PLMs.

To this end, this paper proposes to use DeBERTa fine-tuned with Low-Rank Adaptation (LoRA) (Hu et al., 2021) in PEFT and RoBERTa for all tracks in this competition. Both the DeBERTa and RoBERTa were initialized from a well-trained checkpoint, e.g., `deberta-v2-xxlarge` with 1.5B parameters and `roberta-base` with 125M parameters. For finetuning, LoRA only fine-tuned a small number of (extra) model parameters, thereby greatly decreasing the computational and storage costs. For classification tasks, a softmax head with the cross-entropy loss was applied, while a linear decoder head with the mean squared error was adopted for regression tasks.

The experimental results on the development dataset show that the XXL version of DeBERTa with LoRA and back translation achieves the best performance in tracks 1, 3, and 5. Although the number of trainable parameters decreases, the model achieves performance comparable to that of full fine-tuning. Additionally, RoBERTa with back translation achieved the best performance in tracks 2 and 4. The difference in the performance of the two models on different tracks may be due to the impact of the size of the training dataset.

The rest of this paper is organized as follows. Section 2 describes the system model and method. Section 3 discusses the specific experimental results. Conclusions are finally drawn in Section 4.

2 System description

The architecture of the proposed model is shown in Figure 1. The given text of conversations or essays is input into the tokenizer and then segmented into the corresponding token ID. Subsequently, DeBERTa or RoBERTa's encoder is used to extract the features of the text in a vector format. Meanwhile, LoRA is used to reduce fine-tuning parameters without degrading performance too much. Finally, the encoded hidden representation is used for both sentiment classification and regression.

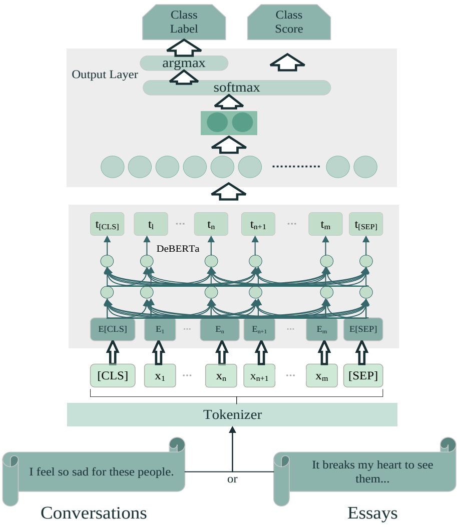

Class Label
Class Score
Output Layer
argmax
softmax
....
$t_{[CLS]}$ t_1 ... t_n t_{n+1} ... t_m $t_{[SEP]}$
DeBERTa
E[CLS] E_1 ... E_n E_{n+1} ... E_m E[SEP]
[CLS] x_1 ... x_n x_{n+1} ... x_m [SEP]
Tokenizer
I feel so sad for these people.
or
It breaks my heart to see them...
Conversations
Essays

Figure 1: The structure for the system.

2.1 Tokenizer

SentencePiece and WordPiece were used for DeBERTa and RoBERTa to divide the text into subwords, respectively. The final output X of the tokenizer is denoted as,

$$X = [CLS]x_1x_2 \dots x_m[SEP] \quad (1)$$

where m is the length of the given text, the [CLS] special tag is used to indicate the beginning of a text sequence, and the [SEP] special tag is used to indicate the separation of a text sequence.

2.2 RoBERTa

The RoBERTa used in this system is a model improved on BERT. BERT's pre-trained tasks include Masked Language Model (MLM) and Next Sentence Prediction (NSP). RoBERTa removed the NSP task, increased the batch size of the model, and used more training data. The performance improvement of RoBERTa has been demonstrated through experimental comparison. The RoBERTa used in this task was initialized from `roberta-base`, with the main structure of 12 layers, 768 hidden size, and 125M total parameters.

2.3 DeBERTa

DeBERTa used in this system improves the text representation capabilities of BERT and RoBERTa models using disentangled attention and enhanced mask decoder methods. Each word is represented using two vectors that encode its content and position, respectively. The attention weights among

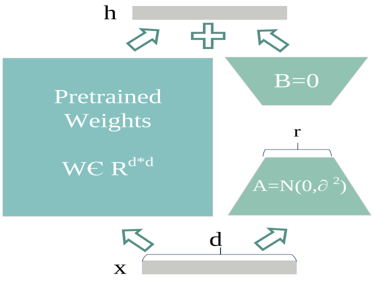


Figure 2: The conceptual diagram of parameter-efficient LoRA fine-tuning.

words are computed using disentangled matrices on their contents and relative positions. Then, an enhanced mask decoder is used to replace the output softmax layer to predict the masked tokens for model pretraining. It outperforms BERT and RoBERTa on many natural language understanding (NLU) tasks. The checkpoint of DeBERTa used in this system is `deberta-v2-xxl`, with the main structure of 48 layers, 1536 hidden size, and 1.5B total parameters.

2.4 LoRA

Transferring the models to the downstream tasks usually depends on the size of the training dataset and pre-trained model. However, the hardware cost of using large models is very significant. Meanwhile, large models are over-parameterized and have a smaller intrinsic dimension (Houlsby et al., 2019). Therefore, this system used LoRA to freeze most parameters and fine-tune the model through low-rank matrices. The LoRA decomposition is defined as,

$$W_0 + \Delta W x = W_0 + BAx \tag{2}$$

where W_0 represents the original parameter matrix. It is very huge and difficult to train. In this system, training updates to W_0 can be represented by ΔW. Therefore, W can be frozen to reduce a large number of training parameters. A and B represent the low-rank factorization matrix of W_0. A is initialized with random Gaussian and B is initialized with zero. Therefore, ΔW is initialized with zero.

LoRA reduces parameters by training a low-rank iterative decomposition matrix of the original parameter matrix. The original parameters of XXL DeBERTa used in this system are 1.5B, while the trainable parameters after LoRA processing are around 4 million. So, this method makes using a large language model on consumer-grade GPUs a reality.

2.5 Output Layer

The output layer is implemented in two distinct ways to accomplish classification and regression tasks.

Regression. Regression was performed for tracks 1, 2, 4, and 5. The training goal is to minimize the mean squared error (MSE) loss, denoted as,

$$L_1 = \frac{1}{n}\sum_{i=1}^{n}(y_i - P_i)^2 \tag{3}$$

where P_i is the predicted value, y_i represents the ground-truth, and n represents the number of training samples in a batch.

Classification. The classification was performed for track 3. A softmax function is used to predict probability distribution over the candidate labels. The training objective is to minimize the cross-entropy between the predicted labels and the ground truth, denoted as,

$$L_2 = -\frac{1}{N}\sum_{i}\sum_{c=1}^{C} y_{ic}\log P_{ic} \tag{4}$$

where C represents the number of categories classified, y_{ic} is the ground-truth label, and P_{ic} represents the prediction probability of the c-th class.

3 Experimental Results

This section evaluates the performance of the proposed system for both sentiment classification and regression tasks.

3.1 Datasets

This task is based on an Empathic Conversations dataset. The dataset marks conversations and essays after people read news stories about individuals, groups, or others who have been harmed (Omitaomu et al., 2022). This dataset for training contains two levels of sentiment classification: (1) Conversations between two users after reading the same news stories. The labels mainly include Emotional Polarity, Emotion, and Empathy. (2) Essays from each user. The labels mainly include Empathy, Emotion, Personality, and Interpersonal Reactivity Index. Each sentimental transition in user conversations or essays is interpreted as labels. The size of the training dataset for the conversation level is around 8700, while the size of the training

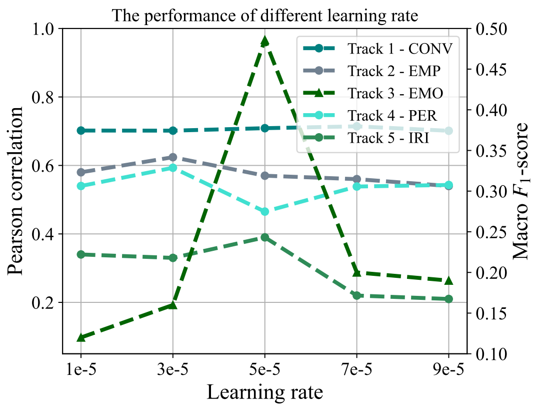


Figure 3: The performance of different learning rates on development dataset.

dataset for the essay level is around 770. Macro Correlation metric is used in tracks 1, 2, 4, and 5, Macro F1-score is used in track 3.

3.2 Implementation Details

The conversation-level dataset provided conversation text, and the essay-level dataset provided essay text and person-level demographic information (age, gender, ethnicity, income, and education level). In track 1, this system used conversation text as training data and used essay text as training data in tracks 2, 3, 4 and 5. All training datasets are first translated into Chinese and then translated back into English. This method of back translation can double the training datasets. Additionally, this system has chosen BERT as a baseline model.

The learning rate was fine-tuned on the development dataset. The results were shown in Fig. 3.

3.3 Comparative Results and Discussion

Track	BERT	RoBERTa	DeBERTa+LoRA
Track1-CONV	0.714	0.721	**0.767**
Track2-EMP	0.502	**0.624**	0.544
Track4-PER	0.342	**0.593**	0.508
Track5-IRI	0.278	0.353	**0.39**

Table 1: Comparative results using Pearson Correlation in the development dataset.

Track	BERT	RoBERTa	DeBERTa+LoRA
Track3-EMO	0.271	0.169	**0.486**

Table 2: Comparative results using Macro F_1 score in the development dataset.

Track	Score
Track1-CONV	0.730 (Pearson Correlation)
Track2-EMP	0.288 (Pearson Correlation)
Track3-EMO	0.514 (Macro F_1)
Track4-PER	0.252 (Pearson Correlation)
Track5-IRI	0.154 (Pearson Correlation)

Table 3: Final score in the test dataset.

Tables 1 and 2 show the comparative results of BERT, RoBERTa, and DeBERTa with LoRA on different classification and regression tasks on the development dataset. It can be found that the average performance of the optimized RoBERTa and DeBERTa is better than BERT. DeBERTa's disentangled attention mechanism helps to improve the model's text representation ability because it not only calculates the attention weight of content and relative position for all word pairs but also considers the absolute positions of words. The results show that DeBERTa + LoRA performs better in tracks 1, 3, and 5, while RoBERTa performs better in tracks 2 and 4. This may be due to the relatively larger scale of training data for track 1, and the fact that track 3 is a complex 31-classification task. Therefore, DeBERTa+LoRA improves the performance of sentiment classification and regression tasks. We submitted the best results of each track on the leaderboard. The final results of the test dataset are shown in Table 3.

4 Conclusion

This paper proposed a system submitted in shared task 1 of WASSA-2023, which uses RoBERTa and XXL version of DeBERTa as the pre-trained models and fine-tuning the DeBERTa model using LoRA. The experimental results indicate that this system has achieved good performance. In addition, this system has a lot of space for improvement compared to the top-ranked systems. Future works will attempt to try other text augmentation and generation methods to achieve better results.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (NSFC) under Grants Nos. 61966038 and 62266051. The authors would like to thank the anonymous reviewers for their constructive comments.

References

Valentin Barriere, Shabnam Tafreshi, João Sedoc, and Salvatore Giorgi. 2023. Wassa 2023 shared task:

Predicting empathy, emotion and personality in interactions and reaction to news stories. In *Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Marie-Catherine de Marneffe, Anna N. Rafferty, and Christopher D. Manning. 2008. Finding contradictions in text. In *ACL 2008, Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, June 15-20, 2008, Columbus, Ohio, USA*, pages 1039–1047. The Association for Computer Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced BERT with disentangled attention. *CoRR*, abs/2006.03654.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *CoRR*, abs/2106.09685.

Valentin Jijkoun and Maarten Rijke. 2005. Recognizing textual entailment using lexical similarity. *Journal of Colloid and Interface Science - J COLLOID INTERFACE SCI*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751. ACL.

Jun Kong, Jin Wang, and Xuejie Zhang. 2022. Hierarchical BERT with an adaptive fine-tuning strategy for document classification. *Knowl. Based Syst.*, 238:107872.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR*, abs/1909.11942.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Damilola Omitaomu, Shabnam Tafreshi, Tingting Liu, Sven Buechel, Chris Callison-Burch, Johannes Eichstaedt, Lyle Ungar, and Jo ao Sedoc. 2022. Empathic conversations: A multi-level dataset of contextualized conversations. *arXiv preprint arXiv:2205.12698*.

Bo Peng, Jin Wang, and Xuejie Zhang. 2020. Adversarial learning of sentiment word representations for sentiment analysis. *Information Sciences*, 541:426–441.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *CoRR*, abs/1409.2329.

You Zhang, Jin Wang, and Xuejie Zhang. 2021. Personalized sentiment classification of customer reviews via an interactive attributes attention model. *Knowl. Based Syst.*, 226:107135.

AdityaPatkar at WASSA 2023 Empathy, Emotion, and Personality Shared Task: RoBERTa-Based Emotion Classification of Essays, Improving Performance on Imbalanced Data

Aditya Patkar[1] and **Suraj Chandrashekhar**[2] and **Ram Mohan Rao Kadiyala**[3]
University of Maryland, College Park
[1]`apatkar@umd.edu` [2]`stelugar@umd.edu` [3]`rkadiyal@umd.edu`

Abstract

This paper presents a study on using the RoBERTa language model for emotion classification of essays as part of the 'Shared Task on Empathy Detection, Emotion Classification and Personality Detection in Interactions' (Barriere et al., 2023), organized as part of 'WASSA 2023' at 'ACL 2023'. Emotion classification is a challenging task in natural language processing, and imbalanced datasets further exacerbate this challenge. In this study, we explore the use of various data balancing techniques in combination with RoBERTa (Liu et al., 2019) to improve the classification performance. We evaluate the performance of our approach (denoted by adityapatkar on Codalab (Pavao et al., 2022)) on a multi-label dataset of essays annotated with eight emotion categories, provided by the Shared Task organizers. Our results show that the proposed approach achieves the best macro F1 score in the competition's training and evaluation phase. Our study provides insights into the potential of RoBERTa for handling imbalanced data in emotion classification. The results can have implications for the natural language processing tasks related to emotion classification.

1 Introduction

Emotion detection and classification in natural language processing (NLP) is a crucial task with various applications such as sentiment analysis, recommendation systems, and chat-bots. In recent years, deep learning-based models, particularly those based on transformer architectures, have shown remarkable performance in a range of NLP tasks. Among them, the RoBERTa model has gained significant attention for its superior performance on various metrics.

However, emotion classification is a particularly challenging task, as emotions are subjective and context-dependent, and often manifest in subtle and nuanced ways. Additionally, imbalanced datasets, where certain emotion categories have fewer instances than others, are common in emotion classification tasks, further complicating the task.

We had to work with a highly imbalanced dataset. The task being multi-label added further complexity. Previous works on emotion classification mainly focus on single label classification (Barriere et al., 2022), use somewhat balanced dataset (Demszky et al., 2020) or work with texts shorter than essays (Mohammad, 2012). The last point is crucial as when working with essays, we have to consider the perceived emotions of the complete essay.

Our proposed system contains a RoBERTa-large model. We will discuss various techniques that we tried to overcome the challenges faced by an imbalanced dataset, biggest of which was over-fitting to the majority label. Some approaches include using paraphrasing to increase the size of the dataset, adding class weights as a feature, weight decay etc.

2 Dataset

The dataset provided for the emotion classification task contains essays written in response to news articles where harm to individuals or groups is present. The dataset was divided into training, development, and test sets. Table 1 shows the number of samples in each split of the dataset. We focused our analysis on the 'essay' and 'emotion' columns, discarding other columns like gender and age, which were not relevant to the task.

During the exploratory data analysis, we observed that the training and development datasets were comparable in terms of essay length (averaging between 75-80 words per essay) and the split between single-label and multi-label rows. Furthermore, we analyzed the top five most frequent words after removing stop-words in the training dataset, which included 'people,' 'like,' 'feel,' 'think,' and 'sad.' This finding suggests that the essays in the dataset often expressed personal opinions and feel-

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 531–535
July 14, 2023 ©2023 Association for Computational Linguistics

Dataset	Rows
`Training`	792
`Development`	208
`Test`	100
`Total`	1100

Table 1: Number of rows in each split of the dataset.

Emotion	Occurrences
`Sadness`	383
`Neutral`	240
`Joy`	10
`Anger`	124
`Surprise`	19
`Disgust`	100
`Fear`	33
`Hope`	32

Table 2: Number of samples for each emotion in the training dataset.

ings related to human experiences.

To gain further insights, we also examined the top direct objects of the verb in the sentences for each label. For example, the words ’journey’ and ’hardship’ were amongst the top 5 most frequent direct objects for the label ’Joy,’ while for ’Sadness,’ the words ’life’ and ’child’ were present amongst the top 5. This observation highlights the differences in language use across different emotions and provides clues to the underlying emotional experiences. Our exploratory data analysis sheds light on the characteristics of the dataset and provides valuable insights into the language use associated with different emotions.

One of the major challenges encountered in the emotion classification task was the presence of ’data imbalance’. Also, the emotion column in the dataset allowed for a single essay to have multiple emotions, making the classification task more complex. For instance, an essay could express both ’Disgust’ and ’Anger.’

The training dataset exhibited a highly skewed distribution towards ’Sadness’ and ’Neutral’ emotions, as shown in Table 2. The issue of data imbalance is evident from the fact that our baseline model failed to predict the under-represented emotions, such as ’Joy’ and ’Surprise.’

In natural language processing tasks, data imbalance is often addressed through under-sampling or oversampling techniques. We implemented a few techniques to address this problem. We will discuss those in the forthcoming sections.

3 Baseline

To establish a baseline for our multi-label emotion classification task, we opted to fine-tune the BERT model (Devlin et al., 2019) using the HuggingFace Transformers library (Wolf et al., 2020). We detached the head of the model to customize it for our task. For data pre-processing, we retained only the ’essay’ and ’emotion’ columns as they were relevant for our task. We utilized the ’bert-base-uncased’ tokenizer to obtain the final embedding of the essays, which we one-hot encoded for multi-label classification.

We trained the BERT model using TensorFlow, with a learning rate of 2e-05, binary cross-entropy as the loss, and the Adam optimizer. We trained the model for 25 epochs, with a batch size of 16. To prevent over-fitting, we implemented early stopping by monitoring the validation loss. To optimize the threshold on the logits for the labels, we performed a random search. As we trained the model, our training loss kept going down, but after a point, the validation loss did not go down. Instead it increased. This is a sign of overfitting. Figure 1 shows us how the baseline model performed for each individual label.

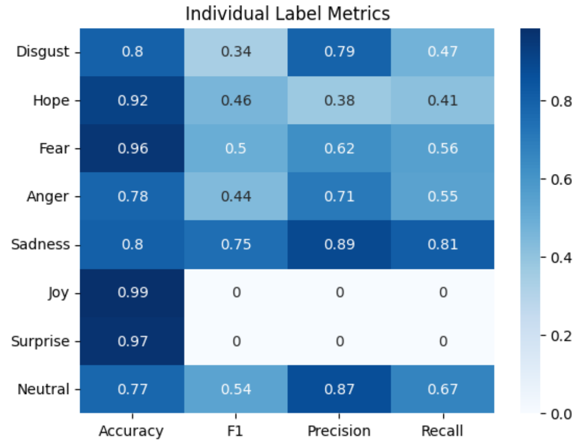


Figure 1: Individual label accuracy, precision, recall and F-1 score of the baseline model on the development dataset.

4 Proposed Approach

We introduce a system[1] that takes into account the limitations put forward by the data imbalance. We

---

[1] Source code available at `https://github.com/adityapatkar/WASSA2023_EMO`

use the 'roberta-large' model with a few tweaks which helped us beat the results of the baseline in every metric.

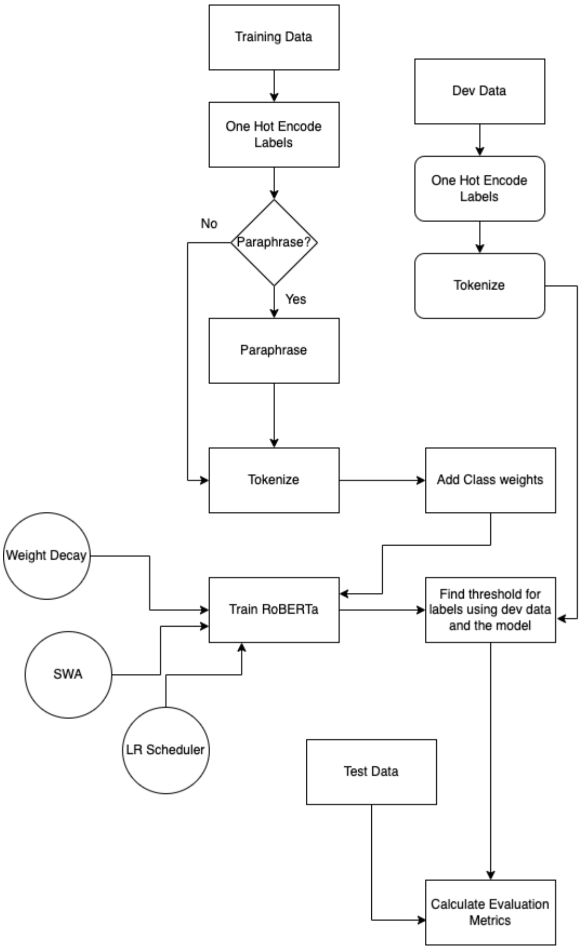


Figure 2: High-level flowchart of our proposed system.

4.1 Our System

Our proposed system deviates from the baseline in several important ways, which we believe have contributed to improved performance on the task at hand.

Firstly, we utilize the RoBERTa model, which has a deeper architecture than the BERT model used in the baseline. This deeper architecture allows for more effective learning of context, which is particularly important for the purpose of this multi-label classification task.

In addition to using a different model architecture, we also implement weight decay to address the issue of over-fitting. This involves setting a defined percentage of weights to exponentially decay to zero, helping to prevent the model from becoming too specialized to the training data. Furthermore, we also implement Stochastic Weight Averaging (SWA) as a means of improving generalization (Izmailov et al., 2019). Our implementation of SWA involves modifying the learning rate schedule and averaging weights after each epoch.

One major innovation we introduced was the use of class weights as a feature in the training dataset. By calculating class weights for each essay based on their inverse frequency, we were able to improve the macro F-1 score by 11%. Specifically, for each class i, we calculated the weight w_i as $w_i = \frac{n}{k \times n_i}$, where n is the total number of samples in the training data, k is the number of classes, and n_i is the number of samples in class i. These weights were then added to the data as a feature.

Finally, we attempted to augment the dataset size by paraphrasing essays associated with under-represented emotions. However, we found that this approach was not effective, as the model started to over-fit to the training data.

Figure 2 provides a high-level flowchart of our system, highlighting the key differences from the baseline. Overall, our changes to the system architecture and training approach have led to improved performance on the task at hand.

4.2 Training

Our system employs the 'roberta-large' model, which was trained for 25 epochs using a learning rate of 2e-5. To avoid over-fitting, we implemented a weight decay of 0.8%. We also employed the Stochastic Weight Averaging (SWA) technique, which averages the model weights after each epoch, thereby enhancing generalization. We used the Adam optimizer for optimization and binary cross-entropy as the loss function.

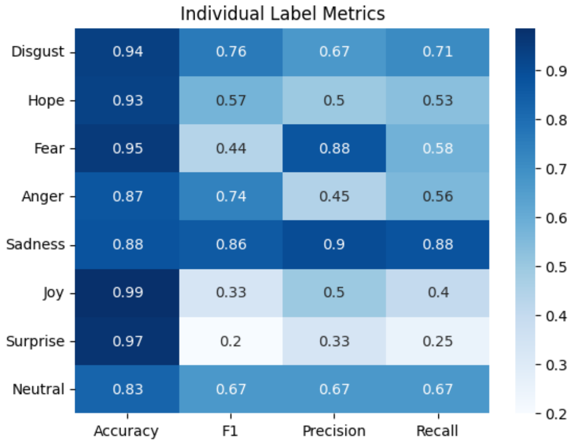


Figure 3: Individual label accuracy, precision, recall and F-1 score of our system on the development dataset.

Development Metrics

Model	Macro F-1	Macro Recall	Macro Precision
`DistilBERT`	0.4021	0.3797	0.5560
`Baseline BERT`	0.4409	0.4458	0.4835
`BERT + Paraphrasing`	0.4207	0.4062	0.4614
`BERT + RoBERTa Ensemble`	0.4251	0.3703	0.5250
`Proposed System with RoBERTa`	**0.5790**	**0.6251**	**0.5711**

Test Metrics

Model	Macro F-1	Macro Recall	Macro Precision
`Baseline BERT`	0.5464	**0.7257**	0.5039
`Proposed System with RoBERTa`	**0.7012**	0.6773	**0.8105**

Table 3: Evaluation metrics on the development and test set.

4.3 Evaluation

Similar to the baseline model, we performed a random search on the logits to determine the best threshold. Our search yielded a value of -0.075 as it maximized the F-1 score on the development set. Figure 3 displays the performance of our proposed system for each individual label. It is evident that there is an improvement in performance compared to the baseline. Notably, emotions with a low number of samples are now being predicted, which the baseline failed to predict. Table 3 compares the results of the baseline and the proposed system on the development set and the test set. Our system shows a considerable improvement in all metrics except the recall over the baseline.

5 Alternate Approaches

We employed alternative approaches in our study, in addition to the baseline model. The first approach involved utilizing a DistilBERT model (Sanh et al., 2020), for a strictly single-label classification. The model was trained for 100 epochs with a learning rate of 1e-5. However, the results were unsatisfactory.

In the second approach, we utilized BERT as the underlying model, similar to the baseline approach, but expanded the dataset by paraphrasing essays related to labels with less than 40 samples. However, this approach did not perform better than the baseline, due to overfitting of the training data.

Finally, we had observed that ensemble models had a good performance on emotion classification tasks (Maheshwari and Varma (2022), Ganaie et al. (2022)). We used the PyTorch implementation of RoBERTa and BERT and fine-tuned both models using the binary cross-entropy loss and optimized them using the Adam optimizer with a learning rate of 1e-05. We used the RoBERTa tokenizer for RoBERTa and the BERT tokenizer for BERT to tokenize the input text. After adding the sigmoid head for 8 classes, we trained both models for 40 epochs, with early stopping based on validation loss criteria. For the final prediction, we took the average output probability of the two models and used a threshold of 0.066 to predict the labels.

Details of the performance of these approaches on the development set can be found in Table 3.

6 Conclusion

In conclusion, this paper presents a study on using the RoBERTa language model for emotion classification of essays, focusing on addressing the challenges posed by imbalanced datasets. The proposed approach combines various data balancing techniques with RoBERTa to improve classification performance, and the results show that the proposed approach achieved the best macro F1 score in the competition's training and evaluation phase. The study provides valuable insights into the potential of RoBERTa for handling imbalanced data in emotion classification, which can have implications for natural language processing tasks related to emotion classification. Overall, the proposed approach offers a promising direction for future research in this field.

References

Valentin Barriere, Shabnam Tafreshi, João Sedoc, and Sawsan Alqahtani. 2022. Wassa 2022 shared task: Predicting empathy, emotion and personality in reaction to news stories. In *Proceedings of the 12th Work-*

shop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis, pages 214–227.

Valentin Barriere, Shabnam Tafreshi, João Sedoc, and Salvatore Giorgi. 2023. Wassa 2023 shared task: Predicting empathy, emotion and personality in interactions and reaction to news stories. In *Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*.

Sven Buechel, Anneke Buffone, Barry Slaff, Lyle Ungar, and João Sedoc. 2018. Modeling empathy and distress in reaction to news stories. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4758–4765, Brussels, Belgium. Association for Computational Linguistics.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. Goemotions: A dataset of fine-grained emotions.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

M.A. Ganaie, Minghui Hu, A.K. Malik, M. Tanveer, and P.N. Suganthan. 2022. Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, 115:105151.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. 2019. Averaging weights leads to wider optima and better generalization.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Himanshu Maheshwari and Vasudeva Varma. 2022. An ensemble approach to detect emotions at an essay level. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 276–279, Dublin, Ireland. Association for Computational Linguistics.

Saif Mohammad. 2012. #emotional tweets. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 246–255, Montréal, Canada. Association for Computational Linguistics.

Damilola Omitaomu, Shabnam Tafreshi, Tingting Liu, Sven Buechel, Chris Callison-Burch, Johannes Eichstaedt, Lyle Ungar, and Jo ao Sedoc. 2022. Empathic conversations: A multi-level dataset of contextualized conversations. *arXiv preprint arXiv:2205.12698*.

Adrien Pavao, Isabelle Guyon, Anne-Catherine Letournel, Xavier Baró, Hugo Escalante, Sergio Escalera, Tyler Thomas, and Zhen Xu. 2022. Codalab competitions: An open source platform to organize scientific challenges. *Technical report*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Shabnam Tafreshi, Orphee De Clercq, Valentin Barriere, Sven Buechel, João Sedoc, and Alexandra Balahur. 2021. Wassa 2021 shared task: Predicting empathy and emotion in reaction to news stories. In *Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 92–104.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Huggingface's transformers: State-of-the-art natural language processing.

Curtin OCAI at WASSA 2023 Empathy, Emotion and Personality Shared Task: Demographic-Aware Prediction Using Multiple Transformers

Md Rakibul Hasan[1] **Md Zakir Hossain**[1] **Tom Gedeon**[1]
Susannah Soon[1] **Shafin Rahman**[2]
[1]Optus Centre for AI, Curtin University, Perth WA 6102, Australia
[2]North South University, Dhaka 1229, Bangladesh
`{rakibul.hasan,zakir.hossain1,tom.gedeon,susannah.soon}@curtin.edu.au`
`shafin.rahman@northsouth.edu`

Abstract

The WASSA 2023 shared task on predicting empathy, emotion and other personality traits consists of essays, conversations and articles in textual form and participants' demographic information in numerical form. To address the tasks, our contributions include (1) converting numerical information into meaningful text information using appropriate templates, (2) summarising lengthy articles, and (3) augmenting training data by paraphrasing. To achieve these contributions, we leveraged two separate T5-based pre-trained transformers. We then fine-tuned pre-trained BERT, DistilBERT and ALBERT for predicting empathy and personality traits. We used the Optuna hyperparameter optimisation framework to fine-tune learning rates, batch sizes and weight initialisation. Our proposed system achieved its highest performance – a Pearson correlation coefficient of 0.750 – on the conversation-level empathy prediction task[1]. The system implementation is publicly available at `https://github.com/hasan-rakibul/WASSA23-empathy-emotion`.

1 Introduction

Empathy refers to an individual's capacity to comprehend and express appropriate emotions in response to others' emotions, perspectives and beliefs (Decety and Jackson, 2004). This ability can foster relationships and reduce stress and unhappiness among individuals through interaction. The importance of empathy is evident across a broad range of real-life human interactions, such as patient-doctor (Jani et al., 2012), teacher-student (Aldrup et al., 2022) and human-robot (Spitale et al., 2022) interactions.

The Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis (WASSA) has organised a "Shared Task on Empathy Detection, Emotion Classification and Personality Detection in Interactions" in 2023 (Barriere et al., 2023). The challenge involves predicting empathy, emotion and personality traits from two types of datasets: essay and conversation. The essay-level dataset consists of essays written by study participants in response to news articles involving harm to individuals, groups or other entities. The conversation-level dataset includes textual conversations between participants regarding the news articles. In addition to the textual data (essays and conversations), the datasets also provide demographic and personal information in numerical form. We participated in four tracks of the 2023 challenge, which involves predicting (1) empathy, personality and interpersonal reactivity index from the essay-level dataset and (2) empathy and emotion from the conversation-level dataset.

WASSA 2023 challenge extends from the 2022 challenge (Barriere et al., 2022) that involved predictions from only an essay-level dataset. Participants in 2022 challenge, such as Vasava et al. (2022); Chen et al. (2022); Qian et al. (2022); Del Arco et al. (2022); Lahnala et al. (2022) and Ghosh et al. (2022), employed transformer-based architectures, such as BERT (Devlin et al., 2018). Transformer-based models were also found to be the best-performing model in the WASSA 2021 shared task on empathy prediction (Tafreshi et al., 2021). Apart from WASSA competition, transformer models are also used in predicting empathy in essays written by medical students about simulated patient-doctor interactions (Dey and Girju, 2022).

Transformer models are deemed highly suitable for undertaking text-based empathy prediction owing to their inherent ability to effectively capture long-range dependencies through attention mechanism (Vaswani et al., 2017). Fine-tuning pre-trained transformers harnesses prior knowledge,

---

[1] At the time of writing this paper, official rankings on any tasks and evaluations of several tasks in which we participated have not been published yet.

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 536–541
July 14, 2023 ©2023 Association for Computational Linguistics

leading to enhanced performance while minimising training time. Qian et al. (2022) reported the best performance by just fine-tuning a BERT-based model in their system for the WASSA 2022 shared task. We, therefore, choose to fine-tune pre-trained transformers to predict empathy and personality traits in this challenge. In our prediction pipeline, we utilise numerical information from the datasets, such as participants' demographic information and income, because previous research by Guda et al. (2021) showed demographic information is an important cue in text-based empathy prediction.

Overall, this paper has made the following contributions: (1) we use novel strategies to incorporate numerical demographic and other data in the text-based prediction pipeline, (2) we summarise longer text sequences to fit into the pipeline, and (3) we augment training samples by paraphrasing the textual data.

2 System description

The general prediction system for essay-level tasks is illustrated in Figure 1. In the case of conversation-level tasks, demographic and other personal information are not available in the conversation-level dataset. In that case, our prediction models involve only conversations and summarised articles, followed by paraphrasing to augment the training dataset.

2.1 Number to text mapping

We first discarded data points from the datasets where any component is missing. The data collection process, along with the questionnaires used in the WASSA 2023 datasets, has been detailed in the work of Omitaomu et al. (2022). Based on the reported distribution of demographic information, we have mapped numerical values of gender, education level and race to their corresponding textual information as illustrated in Table 1.

All the textual features were concatenated in the order of appearance, and this combined feature is referred to as the *demographic* feature throughout this paper. We further concatenated the *demographic* feature with the *essay* texts to create the *demographic_essay* feature.

2.2 Article summarisation

The converted article text comprised long sequences with a maximum length of 20,047 characters. In contrast, the *demographic_essay* feature had a maximum of 956 characters, resulting in 236 tokens. Since the BERT tokeniser we used can process a maximum of 512 tokens, the entire article text cannot be processed in its current form. Consequently, we generated summaries of the articles. We employed *flan-t5-base-samsum*[2], which is a fine-tuned variant of the model proposed by Chung et al. (2022).

Numeric feature	**Converted text**
gender	I am <gender>.
age	My age is <age> years.
education level	My education level is <education level>.
race	My race is <race>.
income	My income is <income>.
article_id	I read newspaper article <article_id>.

Table 1: Templates used to transform numerical features into meaningful texts.

The maximum length of the summarised articles was 987 characters. Considering that the *demographic_essay* feature contained 956 characters, resulting in 236 tokens, it seems plausible that incorporating the additional 987 characters of the article summary would be within the limit of BERT's maximum token length of 512.

2.3 Data augmentation

In order to augment the number of training samples, we utilised the *chatgpt_paraphraser_on_T5_base*[3] to paraphrase the *demographic*, *essay* and *article* texts, effectively doubling the size of the dataset.

2.4 Model and hyperparameter tuning

We experimented with different hyperparameter configurations illustrated in Table 2. Specifically, we fine-tuned three transformer models from Huggingface (Wolf et al., 2019). In fine-tuning BERT-based models, weight initialisation plays a critical role (Dodge et al., 2020). Therefore, we also explored various seed values for CPU and GPU. For conversation-level tasks, the length of the conversation texts was comparatively shorter than that of essay-level tasks. Consequently, we investigated larger batch sizes in the range of 2 to 16.

---

[2] `https://huggingface.co/philschmid/flan-t5-base-samsum`

[3] `https://huggingface.co/humarin/chatgpt_paraphraser_on_T5_base`

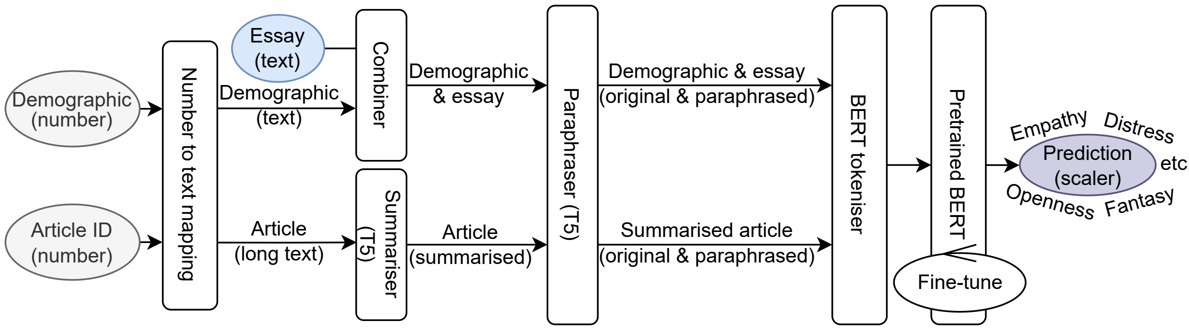


Figure 1: Overall system for essay-level tasks. First, we map numerical features into meaningful text. Next, we leverage a T5-based model to summarise lengthy articles. We use paraphrasing as a data augmentation technique. Finally, we fine-tune a pre-trained BERT model to predict the degree of empathy and other personality traits.

Hyperparameter	Search space
Model	bert-base-uncased, distilbert-base-uncased, albert-base-v2
Learning rate	$[10^{-05} - 10^{-04}]$
Batch size	$[2 - 8]$
Seed	$[1 - 100]$

Table 2: Hyperparameter tuning search space for essay-level tasks.

To tune the hyperparameters, we utilised Optuna (Akiba et al., 2019), with the default tree-structured Parzen estimator as the sampler and the median stopping rule as the pruner. The purpose of the pruner is to stop the tuning process on low-performing hyperparameters early, both to save resources and to enable a greater focus on the best-performing hyperparameters.

The best model, as determined by Optuna, was fine-tuned separately for each of the 14 regression tasks we participated. We employed the Pytorch AdamW optimiser with a default weight decay of 0.01 and betas of 0.9 and 0.999 to optimise the mean-squared-error loss function. To adjust the learning rate, we utilised a linear learning rate scheduler with zero warmup steps. We evaluated the prediction performance of all regression tasks in terms of the official Pearson correlation coefficient metric. We trained all essay-level models for 35 epochs and conversation-level models for 50 epochs. We determined the optimal number of epochs by monitoring the training loss until convergence was reached. We observed that the conversation-level dataset required more epochs for convergence, likely due to its larger size compared to the essay-level dataset.

2.5 Resources

We trained the model on a Tesla V100 32 GB GPU and used the following software packages: Transformers 4.28.1, Datasets 2.12.0, Pytorch 2.0.0, CUDA 11.8, Optuna 3.1.1, Numpy 1.24.3, Pandas 1.5.3, Plotly 5.14.1 with Python 3.10.10.

3 Result & analysis

To determine which feature sets are most effective for predicting empathy, we conducted an experiment in which we combined different features (*essay*, *demographic*, *demographic_essay* and *article*) and trained a DistilBERT (Sanh et al., 2019) model using 5-fold cross-validation for 10 epochs. Huggingface's tokeniser allowed us to tokenise pairs of sequences together by automatically concatenating them with a special [SEP] token. We then used these pairs of features and evaluated their performance, as presented in Table 3.

Features	Average Pearson r
demographic_essay-article (long)	0.819
demographic_essay-article (summary)	0.865
essay-demographic	0.807
essay-article	0.565

Table 3: Five-fold cross-validated (combined training and development set) essay-level empathy prediction performance using various input features in a DistilBERT model. The *demographic_essay* feature refers to manually concatenated pairs of *demographic* and *essay* texts, while the hyphenated features, such as *essay-demographic*, denote automated concatenation by the tokeniser.

Conversion from the longer version of the *article* text to its summarised shorter version improved the performance (Table 3). We speculate that the reason for comparatively lower performance with longer articles is the BERT tokeniser's limitation in accommodating longer texts. The inclusion of *demographic* and *article* features with the *essay* feature improved the model's overall performance. Therefore, we have incorporated *demographic_essay* and *article* features in our final model for the essay-level tasks.

It is worthwhile to note that the use of data augmentation techniques such as paraphrasing can introduce very similar samples in the dataset. It may bias the evaluation metrics, especially when similar samples are present in both the training and validation sets. The cross-validated Pearson correlation coefficient reported in Table 3 includes both the training and development sets with data augmentation (paraphrasing). However, in the process of tuning the model hyperparameters, we only used paraphrasing with the training set and not with the development set to prevent any potential bias caused by the duplication of similar samples.

Among the pre-trained transformer models we experimented with (BERT, DistilBERT and ALBERT), the BERT base model was the best-performing model. Accordingly, we used BERT and tuned the other hyperparameters. We conducted 200 and 100 Optuna trials for essay-based empathy and distress prediction models, respectively. As the best set of hyperparameters is always found within the first 50 trials in the essay-level empathy and distress prediction models, 50 trials were run for other essay-level prediction models. In the case of conversation-level tasks, 100 Optuna trails were run. Table 4 presents the best set of hyperparameters found by the Optuna trials.

Prediction task	**Learning rate**	**Batch size**	**Seed**	**Pearson r**
Empathy	4.27e-05	5	1	0.785
Distress	1.85e-05	6	6	0.726
Conscientiousness	5.98e-05	7	30	0.791
Openness	1.80e-05	2	81	0.776
Extraversion	1.61e-05	7	34	0.681
Agreeableness	5.15e-05	6	65	0.819
Stability	5.36e-05	5	13	0.627
Perspective taking	4.30e-05	2	65	0.837
Personal distress	4.38e-05	7	56	0.788
Fantasy	5.36e-05	5	13	0.895
Empathic concern	4.92e-05	5	1	0.850
Emotional polarity[a]	1.06e-05	10	96	0.763
Emotion[a]	1.44e-05	10	87	0.768
Empathy[a]	1.97e-05	12	68	0.711

[a] Conversation-level

Table 4: Optimal hyperparameters tuned by Optuna and their evaluation (Pearson correlation coefficient) on the original development set without overlapping samples due to augmentation.

We investigated the relative importance of learning rate, seed and batch size (see Appendix A). Our findings are consistent with prior research by Dodge et al. (2020), which highlighted the impact of seed value on the fine-tuning performance of BERT-based models. However, the relative importance of hyperparameters varied across the prediction tasks, indicating the task-specific nature of fine-tuning pre-trained transformer models. It guided us to train separate models for separate tasks.

We observed that text summarisation and data augmentation (paraphrasing) improved model performance on the development set. On the test dataset, the final model achieved Pearson correlation coefficients of 0.750, 0.683 and 0.573 for conversation-level empathy, emotional polarity and emotional intensity prediction, respectively. For essay-level tasks, we achieved Pearson correlation coefficients of 0.187 and 0.344 for empathy and distress predictions, respectively. The average Pearson correlation coefficient for conversation-level tasks was 0.669, while it was 0.266 for essay-level empathy and distress prediction. The test performance of essay-level personality and interpersonal reactivity index predictions, as well as the official rankings, have not been published at the time of writing this paper. Nevertheless, our system achieved its best performance of a Pearson correlation coefficient of 0.750 in predicting conversation-level empathy.

4 Conclusion

Empathy is a vital human attribute to support and care for others. This paper outlines a comprehensive system for predicting empathy, emotion and other personality traits as part of the WASSA 2023 shared task. To this end, we first map the numerical demographic information into meaningful text since individuals' demographic information, such as age, sex and race, may affect their empathic capacity. Our system utilises pre-trained transformers to map numerical information into meaningful text, summarise longer text sequences, paraphrase text sequences to augment smaller training datasets and finally predict the degree of empathy and other personality traits.

Acknowledgements

This research was undertaken with the assistance of resources from the National Computational Infrastructure (NCI Australia), an NCRIS-enabled capability supported by the Australian Government.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631.

Karen Aldrup, Bastian Carstensen, and Uta Klusmann. 2022. Is empathy the key to effective teaching? a systematic review of its association with teacher-student interactions and student outcomes. *Educational Psychology Review*, 34(3):1177–1216.

Valentin Barriere, Shabnam Tafreshi, João Sedoc, and Sawsan Alqahtani. 2022. WASSA 2022 shared task: Predicting empathy, emotion and personality in reaction to news stories. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 214–227, Dublin, Ireland. Association for Computational Linguistics.

Valentin Barriere, Shabnam Tafreshi, João Sedoc, and Salvatore Giorgi. 2023. WASSA 2023 shared task: Predicting empathy, emotion and personality in interactions and reaction to news stories. In *Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*. Association for Computational Linguistics.

Yue Chen, Yingnan Ju, and Sandra Kübler. 2022. IUCL at WASSA 2022 shared task: A text-only approach to empathy and emotion detection. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 228–232, Dublin, Ireland. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Jean Decety and Philip L Jackson. 2004. The functional architecture of human empathy. *Behavioral and cognitive neuroscience reviews*, 3(2):71–100.

Flor Miriam Del Arco, Jaime Collado-Montañez, L. Alfonso Ureña, and María-Teresa Martín-Valdivia. 2022. Empathy and distress prediction using transformer multi-output regression and emotion analysis with an ensemble of supervised and zero-shot learning models. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 239–244, Dublin, Ireland. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Priyanka Dey and Roxana Girju. 2022. Enriching deep learning with frame semantics for empathy classification in medical narrative essays. In *Proceedings of the 13th International Workshop on Health Text Mining and Information Analysis (LOUHI)*, pages 207–217, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*.

Soumitra Ghosh, Dhirendra Maurya, Asif Ekbal, and Pushpak Bhattacharyya. 2022. Team IITP-AINLPML at WASSA 2022: Empathy detection, emotion classification and personality detection. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 255–260, Dublin, Ireland. Association for Computational Linguistics.

Bhanu Prakash Reddy Guda, Aparna Garimella, and Niyati Chhaya. 2021. EmpathBERT: A BERT-based framework for demographic-aware empathy prediction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3072–3079, Online. Association for Computational Linguistics.

Bhautesh Dinesh Jani, David N Blane, and Stewart W Mercer. 2012. The role of empathy in therapy and the physician-patient relationship. *Complementary Medicine Research*, 19(5):252–257.

Allison Lahnala, Charles Welch, and Lucie Flek. 2022. CAISA at WASSA 2022: Adapter-tuning for empathy prediction. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 280–285, Dublin, Ireland. Association for Computational Linguistics.

Damilola Omitaomu, Shabnam Tafreshi, Tingting Liu, Sven Buechel, Chris Callison-Burch, Johannes Eichstaedt, Lyle Ungar, and João Sedoc. 2022. Empathic conversations: A multi-level dataset of contextualized conversations. *arXiv preprint arXiv:2205.12698*.

Shenbin Qian, Constantin Orasan, Diptesh Kanojia, Hadeel Saadany, and Félix Do Carmo. 2022. SURREY-CTS-NLP at WASSA2022: An experiment of discourse and sentiment analysis for the prediction of empathy, distress and emotion. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 271–275, Dublin, Ireland. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Micol Spitale, Sarah Okamoto, Mahima Gupta, HAO Xi, and Maja J Matarić. 2022. Socially assistive robots as storytellers that elicit empathy. *ACM Transactions on Human-Robot Interaction (THRI)*, 11(4):1–29.

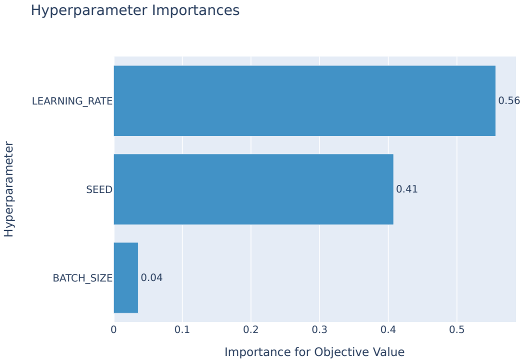
Shabnam Tafreshi, Orphee De Clercq, Valentin Barriere, Sven Buechel, João Sedoc, and Alexandra Balahur. 2021. WASSA 2021 shared task: Predicting empathy and emotion in reaction to news stories. In *Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 92–104, Online. Association for Computational Linguistics.

Himil Vasava, Pramegh Uikey, Gaurav Wasnik, and Raksha Sharma. 2022. Transformer-based architecture for empathy prediction and emotion classification. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 261–264, Dublin, Ireland. Association for Computational Linguistics.

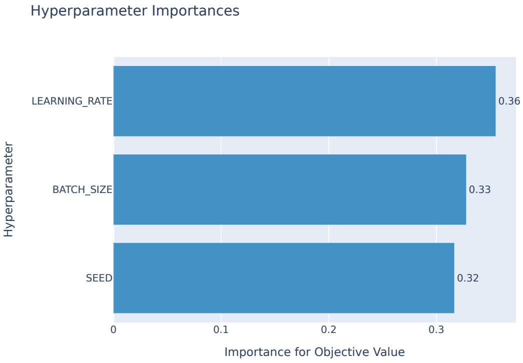
Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. HuggingFace's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

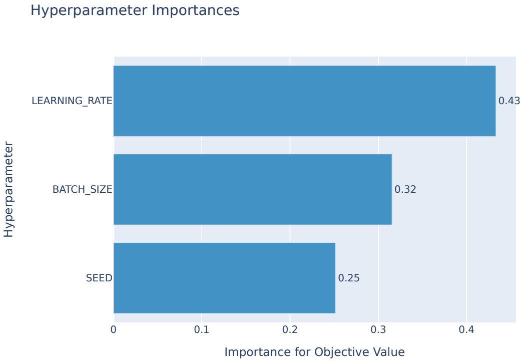
Appendix A Hyperparameter importance



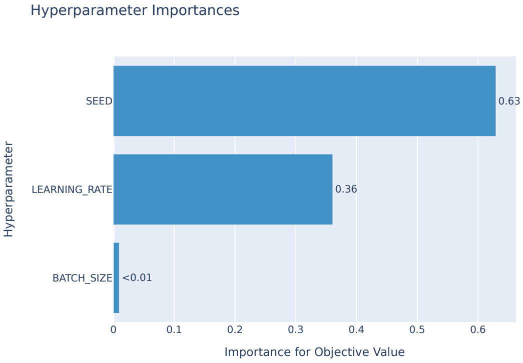
(a) Empathy prediction: learning rate has the highest impact (56% relative importance), followed by the seed value. Batch size is the least impactful (4% relative importance).



(b) Conscientiousness prediction: learning rate, batch size and seed value all have a high impact with a relative importance of around 30%.



(c) Personality distress prediction: learning rate has the highest impact (43% relative importance), followed by batch size (32% relative importance) and seed value (25% relative importance).



(d) Empathic concern prediction: Seed value has the highest impact, followed by the learning rate. Batch size is the least impactful, having less than 1% relative importance.

Figure A1: Relative importance of learning rate, seed value and batch size in various essay-level tasks. Here, the objective value refers to the Pearson correlation coefficient. The variations in hyperparameter importance across tasks indicate the requirements of training separate models for separate tasks.

Team_Hawk at WASSA 2023 Empathy, Emotion, and Personality Shared Task: Multi-tasking Multi-encoder based transformers for Empathy and Emotion Prediction in Conversations

Addepalli Sai Srinivas, Nabarun Barua, Santanu Pal
Wipro AI Research (Lab45), India
`{addepalli.srinivas, nabarun.barua, santanu.pal2}@wipro.com`

Abstract

In this paper, we present Team Hawk's participation in Track 1 of the WASSA 2023 shared task. The objective of the task is to understand the empathy that emerges between individuals during their conversations. In our study, we developed a multi-tasking framework that is capable of automatically assessing empathy, intensity of emotion, and polarity of emotion within participants' conversations. Our proposed core model extends the transformer architecture, utilizing two separate RoBERTa-based encoders to encode both the articles and conversations. Subsequently, a sequence of self-attention, position-wise feed-forward, and dense layers are employed to predict the regression scores for the three sub-tasks: empathy, intensity of emotion, and polarity of emotion. Our best model achieved average Pearson's correlation of 0.7710 (Empathy: 0.7843, Emotion Polarity: 0.7917, Emotion Intensity: 0.7381) on the released development set and 0.7250 (Empathy: 0.8090, Emotion Polarity: 0.7010, Emotion Intensity: 0.6650) on the released test set. These results earned us the 3rd position in the test set evaluation phase of Track 1.

1 Introduction

Empathy involves understanding and sharing others' feelings. In conversation, empathy is demonstrated through active listening, acknowledging emotions, and providing supportive responses. Emotion polarity refers to the positive or negative nature of expressed emotions, while emotion intensity relates to the strength of those emotions.

Computing empathy is an emergent paradigm and become an important component in conversational AI (Mazaré et al., 2018; Roller et al., 2021). Empathy is critical for clinical applications such as automated behavioral therapy (Fitzpatrick et al., 2017). Implementing complex emotional-motivational states and effectively responding in an empathetic manner remains a significant challenge in human-machine interaction.

The dataset shared by the organizers comprises news stories and corresponding brief essays written by participants during conversation sessions. Participants engage in dialogues, assessing each other's conversation turns for empathy, emotion intensity, and polarity. A third-party annotator confirms the emotional dimensions of empathy, intensity, and polarity at the end of the session (Buechel et al., 2018; Sharma et al., 2020; Barriere et al., 2022).

The WASSA 2023 shared task (Barriere et al., 2023) consists of five different tracks: Track 1: Empathy and Emotion Prediction in Conversations (CONV), Track 2: Empathy Prediction (EMP), Track 3: Emotion Classification (EMO), Track 4: Personality Prediction (PER), and Track 5: Interpersonal Reactivity Index Prediction (IRI).

In this paper, we present our submission for Track 1. The provided dataset contains conversations between two participants, along with scores for empathy, emotion polarity, and emotion intensity. The task revolves around two participants who read a news article and initiate a conversation based on that particular news. They are then required to assess each other's empathy, intensity of emotion, and polarity of emotion based on their discussion.

The objective of the task is to develop a system capable of automatically assessing empathy, intensity of emotion, and polarity of emotion within participants' conversations.

Our model consists of two RoBERTa-based encoders to encode article and conversation followed by a 3-layered transformer encoder. The representation is then passed to a sequence of layers which provides regression output of three tasks (cf. Track 1) – Empathy, Emotion Polarity, and Emotion Intensity with a multi-tasking framework (MLT). In Track 1 participation, our best model achieved average Pearson's correlation of 0.771 on the released development set and 0.725 on the released test set. The key findings of this research can be summarized as follows: (i) residual skip connections are

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 542–547
July 14, 2023 ©2023 Association for Computational Linguistics

effective in enhancing the conversation encoder, (ii) including the previous dialogues of both participants in a conversation along with the current dialogue helps in preserving the context of the conversation within the dialogue encoder for a particular session, and (iii) Token interactions between articles and conversations utilizing multi-head self-attention yield significant and informative results.

The rest of this paper is organized as follows. We introduce related work in §2. We discuss problem statement in §3. The proposed model is described in §4 and experiment and result in § 5. Finally, we conclude our paper in §6.

2 Related Work

Recently, transformer based (Vaswani et al., 2017) pre-trained models such as BERT (Devlin et al., 2019), OpenAI GPT (Radford et al., 2018), RoBERTa (Liu et al., 2019) etc. have been shown superior performance in various downstream tasks, including text classification task (Sun et al., 2020; Luo and Wang, 2019; Singh et al., 2021), generation task such as question answering (Garg et al., 2020) and many more. Recent works have shown that using such pre-trained methods can achieve state-of-the-art performance. Towards that end, Sharma et al. (2020) investigated a multi-task RoBERTa-based bi-encoder paradigm for comprehending empathy in text-based health support, Zhou and Jurgens (2020) investigated the link between distress, condolence, and empathy in online support groups using nested regression models. Many research (Abdul-Mageed and Ungar, 2017; Nozza et al., 2017) have given various strategies for emotion recognition. The effectiveness of using transformer encoders for emotion detection was investigated by Adoma et al. (2020). Ghosh et al. (2022) proposed a multi-task deep learning methods to address Empathy Detection, Emotion Classification and Personality Detection. Inspired from (Sharma et al., 2020) and multi-encoder based architectures (Pal et al., 2018, 2019, 2020), we propose a multi-encoder based architecture followed by MLP and a linear layer output layer. Our core architecture is similar to Pal et al. (2018), the difference is we use two RoBERTa (Liu et al., 2019) encoders for inputs.

3 Problem Statement

Our model is based on a multi-task learning based framework (MTL) to force the model to consider three different objectives i.e. three emotional dimensions of empathy (y_1), intensity (y_2), and polarity (y_3). Given a set of conversation for a single session $\mathbf{c}$ for a corresponding article $\mathbf{a_c}$, the output probability $\mathbf{y} \in y_1, y_2, y_3$ in the model setting is calculated as in Equation 1.

$$p(\mathbf{y}) = p(\mathbf{y}|\mathbf{c}, \mathbf{a_c}) \tag{1}$$

The model acts as a regressor, the output head provides regression results. The network consists of a two-layered multi-layered perceptron[1] (MLP) with ReLU activation between layers and output head with a 'linear' activation. Given the output of the regression head, the loss can be calculated as:

$$\mathcal{L}_t = \sum_{(\mathbf{a_c}, \mathbf{c}, \mathbf{y}) \in \mathcal{D}} -logp(\mathbf{y}|\mathbf{c}, \mathbf{a_c}; \theta) \tag{2}$$

Our model is trained for three emotional dimensions of empathy (y_1), intensity (y_2), and polarity (y_3) with corresponding losses $\mathcal{L}_1$, $\mathcal{L}_2$, and $\mathcal{L}_3$ respectively, in an end-to-end fashion that jointly optimizes the loss as in Eq. 3, where α, β, and γ are learnable parameters.

$$\mathcal{L}_{overall} = \alpha * \mathcal{L}_1 + \beta * \mathcal{L}_2 + \gamma * \mathcal{L}_3 \tag{3}$$

4 Model Architecture

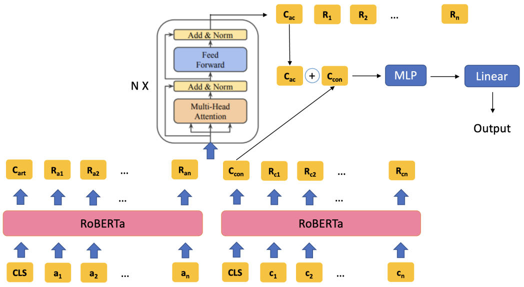


Figure 1: Our proposed model architecture

Our proposed model consists of two Roberta encoders[2] – article encoder (enc_{art}) and conversation encoder (enc_{conv}) based on the inputs from news articles and conversations. In the proposed model, the last hidden state of both encoder representations are merged and passed through a 3-layered transformer encoder (cf. Figure 1). A residual connection is employed between the CLS pooling of the conversation encoder and the output of the 3-layered transformer encoder. The combined CLS

---

[1] Our MLP layer consists of two fully connected feed forward layers and a ReLU activation between them.

[2] For our submission, we use Roberta Large: `https://huggingface.co/roberta-large`

pooling representation is then inputted into MLPs followed by a linear layer. The linear layer generates regression outputs for the three tasks: Empathy, Emotion Polarity, and Emotion Intensity. This model utilizes a multi-tasking framework (MLT) that jointly optimizes each individual loss function (see §3) in an end-to-end manner.

5 Experiments and Results

In this section, we provide a summary of how our systems were trained, tuned, and combined to create the Team Hawk submissions for Track 1 of the WASSA 2023 Shared Task. We evaluated our system using Pearson's correlation, which measures the relationship between the regression outputs of our model and the gold standards. All benchmark evaluation scores are reported based on the development set released by the organizers.

5.1 Dataset

The track 1 dataset (Barriere et al., 2023; Omitaomu et al., 2022) used in this research consists of conversations between two individuals within a given session. The dataset includes several columns, such as conversation-id, turn-id, conversation text, emotional polarity, emotion, empathy, speaker-number, article-id, speaker-id, essay-id, and more. Additionally, an article dataset is also utilized, which contains article-id and article-text.

The overall training data comprises 8,778 labeled data points. To conduct the experiments, the released training set is split into a validation set, consisting of 1,756 data points, and a train set, containing the remaining 7,022 data points.

For evaluation purposes, the released development data consisting of 2,400 data points is used as the in-hand test set.

5.2 Data Pre-processing

To prepare inputs to our model we performed data preprocessing. The enc_{art} takes input as articles with <a_id:article_id>. To the enc_{conv}, we provide the contextual conversations (C) along with Conversation ID, Turn ID, Article ID & Speaker ID represented as <c_id:conversation_id>, <tid:turn_id>, <a_id:article_id> & <s_id:speaker_id> respectively for each conversation (c). For i^{th} training instance for a particular session the contextual conversion means $C = concat(c_{<i}, c_i)$.

5.2.1 Baseline Model

Our baseline model is similar to our proposed model however the model does not have transformer encoder (cf. Figure 2).

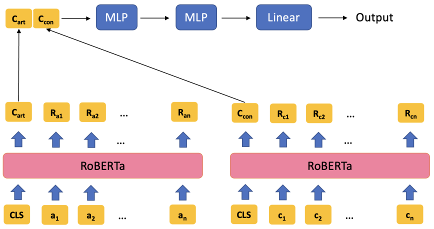


Figure 2: Baseline Model

The baseline model demonstrated an average Pearson correlation of 0.623 on the validation set. However, on the training set, it achieved an average Pearson correlation of approximately 0.977. This significant difference between the performance on the training set and the validation set indicates clear overfitting, as the model has excessively adapted to the training data.

5.2.2 Baseline with residual connection

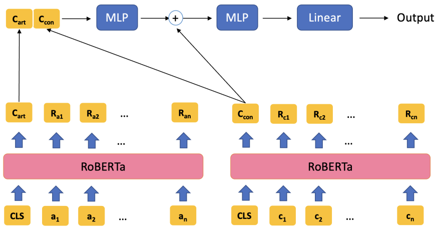


Figure 3: Residual Model

In this experiment, we add a residual connection between MLP representation of combined encoders (concatenation of enc_{art} and enc_{conv}) and the conversation encoder, enc_{conv} (cf. Figure 3). In our model, the combination of both encoders through CLS pooling raises concerns about the possibility of information loss that has been learned by the enc_{conv}. To address this issue, we utilize skip connections, which not only help alleviate the vanishing gradient problem but also improve training efficiency. However, the key question is why we use skip connections in our model.

The use of skip connections ensures that the gradient flows equally to both the article and dialogue encoders. While the article remains constant, empathy, emotion polarity, and emotion intensity are primarily driven by the conversation. Therefore, it is crucial to emphasize the importance of the conversation in our model to provide a diverse range of contexts for the model to learn from.

Furthermore, introducing residual connections offers another avenue for improvement in the output head. With the presence of multiple linear layers, there is a risk of overfitting. The inclusion of residual connections helps mitigate this concern and enhances the model's performance by allowing the flow of information from earlier layers to later layers, enabling better representation learning.

After combining the representations from the two encoders, we proceed with the following steps:

Minimizing the output feature: Instead of directly passing the input through multiple linear layers, we utilize two Multi-Layer Perceptrons (MLPs). Each MLP consists of two linear layers with ReLU activation. The first MLP reduces the output feature dimension from $\mathbf{R}^{1536}$ to $\mathbf{R}^{768}$ (specific to the RoBERTa base). This reduced feature is then passed through Layer Normalization to enhance training stability and speed.

Further feature reduction: The output from Layer Normalization is fed into the second MLP, which reduces the feature dimension from $\mathbf{R}^{768}$ to $\mathbf{R}^{32}$. Layer normalization is once again applied to the output to maintain stability.

Final linear layer: The output from the second MLP is then passed through a linear layer, which further reduces the feature size from $\mathbf{R}^{32}$ to $\mathbf{R}^{1}$.

These steps enhance the model's to an improved output for regression tasks. As a result, we achieved a significant increase in the Pearson correlation, improving it from 0.623 to 0.724.

5.2.3 The proposed model

Here, we introduced a 3-layered transformer encoder (cf. §4). The main difference from the previous experiment is that instead of passing the concatenation of the CLS pooling of enc_{art} and enc_{conv} to the MLP, we pass the combined last hidden state ($\mathbf{R}^{512\times1536}$) of both encoders (for RoBERTa base, the representation of $enc_{art} \in \mathbf{R}^{512\times768}$ and $enc_{conv} \in \mathbf{R}^{512\times768}$) to 3 layered transformer encoder (cf. Figure 1). Similar to the previous experiment, we include a residual connection between the CLS Pooling of the 3-layered transformer encoder ($\mathbf{R}^{768}$) and the CLS Pooling of the conversation encoder ($\mathbf{R}^{768}$). Finally, the combined representation ($\mathbf{R}^{1536}$) is then inputted to an MLP, followed by a Linear Layer that generates the regression output (see steps in §5.2.2).

5.2.4 Result and Discussions

Our submission results are shown in Table 1. As discussed in Section 5.2.1 and 5.2.2, we can see our proposed approach provides best Pearson's scores on development data compared to both baseline and baseline with residuals. We also report testset score evaluated by the organizers. Out submission ranked 3rd in Track 1. In this submission, we use RoBERTa-large for both encoders i.e., enc_{art} and enc_{conv} (cf. Figure 1), with maximum token length 512. Other hyper-parameters include batch size = 6, learning rate = $5e^{-5}$, number of epochs = 30. For our custom transformer encoder, we set number of layers = 3, embedding size = 1024, and head size = 8. All models are trained with mean-squared error loss criteria and optimized with default configuration of Adam optimizer.

Data	Emp	Emo-Pol	Emo-Int	Avg
Dev	0.7843	0.7917	0.7381	0.7710
Test	0.8090	0.7010	0.6650	0.7250

Table 1: Performance of our submission based on Pearson's score on *Development* & *Test sets*. Here, we use RoBERTa-large for enc_{art} and enc_{conv}, with maximum token length 512. Emp: Empathy Emo-Pol: Emotion polarity, Emo-Int: Emotion Intensity, Avg: Average.

6 Conclusion and Future Work

In this paper, we introduced our methodologies for investigating the emergence of empathy during conversations. The task is introduced as part of the WASSA 2023 shared task track 1, which focuses on Empathy and Emotion Prediction in Conversations (CONV). We developed a multi-tasking framework that leverages a core multi-encoder based architecture. This framework enables automatic assessment of empathy, intensity of emotion, and polarity of emotion in participants' conversations. Our systems achieved average Pearson's scores of 0.7710 on the released development set and 0.7250 on the released test set. Our submission ranked 3rd in the shared task. Due to time constraints, we are unable to conduct an exhaustive number of experiments with various architecture variations before reaching a final conclusion. However, for future research, we will conduct a comprehensive analysis involving architecture variations and data preprocessing methods. Additionally, we plan to investigate the influence of other features such as gender, age, etc., on the model's decision-making process.

References

Muhammad Abdul-Mageed and Lyle Ungar. 2017. EmoNet: Fine-grained emotion detection with gated recurrent neural networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 718–728, Vancouver, Canada. Association for Computational Linguistics.

Acheampong Francisca Adoma, Nunoo-Mensah Henry, Wenyu Chen, and Niyongabo Rubungo Andre. 2020. Recognizing emotions from texts using a bert-based approach. In *2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, pages 62–66.

Valentin Barriere, Shabnam Tafreshi, João Sedoc, and Sawsan Alqahtani. 2022. WASSA 2022 shared task: Predicting empathy, emotion and personality in reaction to news stories. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 214–227, Dublin, Ireland. Association for Computational Linguistics.

Valentin Barriere, Shabnam Tafreshi, João Sedoc, and Salvatore Giorgi. 2023. Wassa 2023 shared task: Predicting empathy, emotion and personality in interactions and reaction to news stories. In *Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*.

Sven Buechel, Anneke Buffone, Barry Slaff, Lyle Ungar, and João Sedoc. 2018. Modeling empathy and distress in reaction to news stories. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4758–4765, Brussels, Belgium. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

KK Fitzpatrick, A Darcy, and M Vierhile. 2017. Delivering cognitive behavior therapy to young adults with symptoms of depression and anxiety using a fully automated conversational agent (woebot): A randomized controlled trial. In *JMIR Ment Health.*

Siddhant Garg, Thuy Vu, and Alessandro Moschitti. 2020. TANDA: transfer and adapt pre-trained transformer models for answer sentence selection. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 7780–7788. AAAI Press.

Soumitra Ghosh, Dhirendra Maurya, Asif Ekbal, and Pushpak Bhattacharyya. 2022. Team IITP-AINLPML at WASSA 2022: Empathy detection, emotion classification and personality detection. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 255–260, Dublin, Ireland. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Linkai Luo and Yue Wang. 2019. Emotionx-hsu: Adopting pre-trained bert for emotion classification.

Pierre-Emmanuel Mazaré, Samuel Humeau, Martin Raison, and Antoine Bordes. 2018. Training millions of personalized dialogue agents. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2775–2779, Brussels, Belgium. Association for Computational Linguistics.

Debora Nozza, Elisabetta Fersini, and Enza Messina. 2017. A multi-view sentiment corpus. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 273–280, Valencia, Spain. Association for Computational Linguistics.

Damilola Omitaomu, Shabnam Tafreshi, Tingting Liu, Sven Buechel, Chris Callison-Burch, Johannes Eichstaedt, Lyle Ungar, and João Sedoc. 2022. Empathic conversations: A multi-level dataset of contextualized conversations.

Santanu Pal, Nico Herbig, Antonio Krüger, and Josef van Genabith. 2018. A transformer-based multi-source automatic post-editing system. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 827–835.

Santanu Pal, Hongfei Xu, Nico Herbig, Antonio Krüger, and Josef van Genabith. 2019. Usaar-dfki–the transference architecture for english–german automatic post-editing. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 124–131.

Santanu Pal, Hongfei Xu, Nico Herbig, Sudip Kumar Naskar, Antonio Krüger, and Josef van Genabith. 2020. The transference architecture for automatic post-editing. In *28th International Conference on Computational Linguistics*, 10.18653/v1/2020.coling-main.524, pages 5963–5974. https://www.aclweb.org/anthology/2020.coling-main.524.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Stephen Roller, Emily Dinan, Naman Goyal, Da Ju, Mary Williamson, Yinhan Liu, Jing Xu, Myle Ott, Eric Michael Smith, Y-Lan Boureau, and Jason Weston. 2021. Recipes for building an open-domain chatbot. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 300–325, Online. Association for Computational Linguistics.

Ashish Sharma, Adam Miner, David Atkins, and Tim Althoff. 2020. A computational approach to understanding empathy expressed in text-based mental health support. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5263–5276, Online. Association for Computational Linguistics.

G. Singh, D. Brahma, P. Rai, and A. Modi. 2021. Fine-grained emotion prediction by modeling emotion definitions. In *2021 9th International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 1–8, Los Alamitos, CA, USA. IEEE Computer Society.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2020. How to fine-tune bert for text classification?

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Naitian Zhou and David Jurgens. 2020. Condolence and empathy in online communities. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 609–626, Online. Association for Computational Linguistics.

NCUEE-NLP at WASSA 2023 Empathy, Emotion, and Personality Shared Task: Perceived Intensity Prediction Using Sentiment-Enhanced RoBERTa Transformers

Tzu-Mi Lin, Jung-Ying Chang, and Lung-Hao Lee
Department of Electrical Engineering
National Central University
No. 300, Zongda Rd., Zhongli Dist., Taoyuan City 32001, Taiwan
110521087@cc.ncu.edu.tw, 111521100@cc.ncu.edu.tw, lhlee@ee.ncu.edu.tw

Abstract

This paper describes our proposed system design for the WASSA 2023 shared task 1. We propose a unified architecture of ensemble neural networks to integrate the original RoBERTa transformer with two sentiment-enhanced RoBERTa-Twitter and EmoBERTa models. For Track 1 at the speech-turn level, our best submission achieved an average Pearson correlation score of 0.7236, ranking fourth for empathy, emotion polarity and emotion intensity prediction. For Track 2 at the essay-level, our best submission obtained an average Pearson correlation score of 0.4178 for predicting empathy and distress scores, ranked first among all nine submissions.

1 Introduction

Empathy is the capacity to understand or feel what another person is experiencing from his/her perspectives, which is a cognitive and emotional reaction to observing the situation of others (Omitaomu et al., 2022). Computational detection and prediction of empathy has attracted considerable attention in recent years. Empathy assessment by the writer of a statement was captured and annotated to computationally distinguish between multiple forms of empathy, empathic concerns and personal distress (Buechel et al., 2018). Mixed-Level Feed Forward Network (MLFFN) was proposed to learn word ratings for empathy and distress (Sedoc et al., 2020). Logistic regression models were used to recognize distress and condolences reactions to such distress (Zhou and Jurgens, 2020). A multi-task RoBERTa-based bi-encoder model was developed to identify empathy in conversations (Sharma et al., 2020). A demographic-aware EmpathBERT architecture was presented to infuse demographic information for empathy prediction (Guda et al., 2021).

The Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA) organizes shared tasks for different aspects of affect computation from texts. In WASSA 2021 and 2022 shared tasks that focus on predicting empathy and emotion in reaction to news stories (Barriere et al., 2022; Tafreshi et al., 2021), using transformers based pre-trained language models to achieve promising results. The PVG team proposed a multi-input and multi-task framework based on the RoBERTa transformer for empathy score prediction (Kulkarni et al., 2021). An ensemble of the RoBERTa multi-task model and the vanilla ELECTRA model was used to predict empathy scores (Mundra et al., 2021). The IUCL system fine-tuned two RoBERTa large models, including a regression model for empathy and distress prediction and a classification model for emotion detection (Chen et al. 2022). A multi-output regression model fine-tuned by RoBERTa with additional features, including gender, income and age was used to predict empathy and distress intensity (Arco et al., 2022). The task adapters for a RoBERTa model were trained to predict empathy and distress scores at the essay-level (Lahnala et al., 2022).

WASSA-2023 organizes a similar task with a newly added track on empathy, emotion and self-disclosure detection in conversation at the speech-turn level (Barriere et al., 2023). We participated in the Track 1 for Empathy and Emotion Prediction in Conversations (CONV), aiming to predict perceived empathy, emotion polarity and emotion

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 548–552
July 14, 2023 ©2023 Association for Computational Linguistics

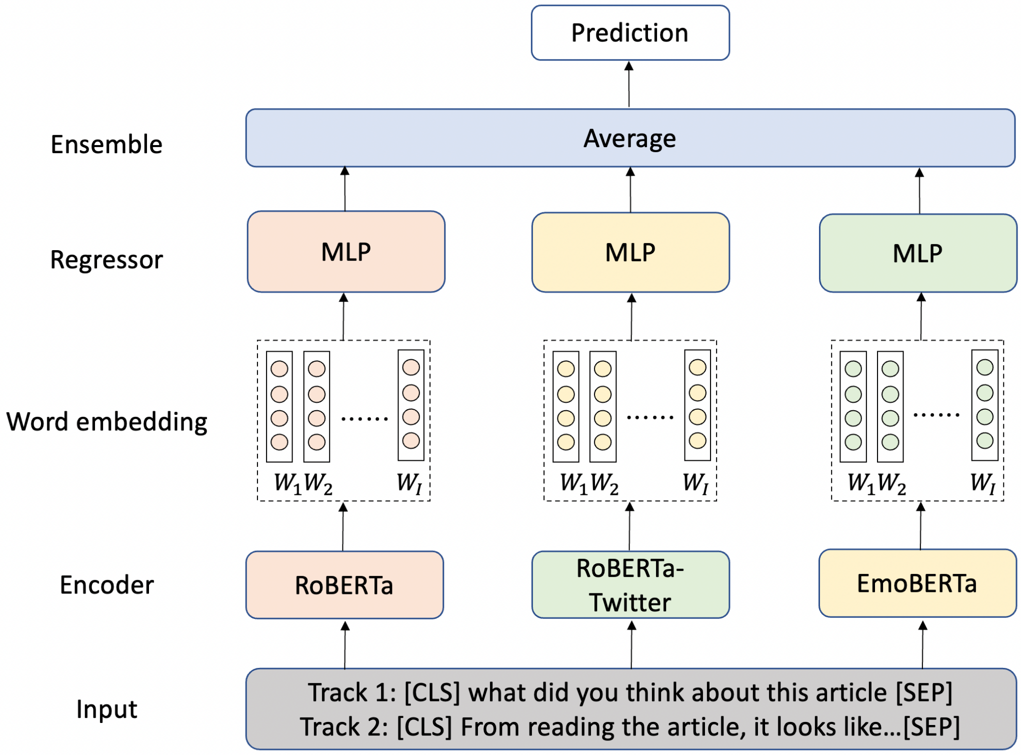


Figure 1: Our NCUEE-NLP system architecture for the Tracks 1 and 2 in the WASSA 2023 task 1.

intensity at the speech-turn-level in a conversation, and Track 2 for Empathy Prediction (EMP) to predict empathy concerns and personal distresses at the essay-level. Both tracks are regression tasks evaluated based on the average of the Pearson correlations. Following the successes of RoBERTa-based models in the previous WASSA shared tasks, we explore the use of sentiment-enhanced RoBERTa models to address the challenges for both tracks in the shared task 1.

This paper describes the **NCUEE-NLP** (**N**ational **C**entral **U**niversity, Dept. of **E**lectrical **E**ngineering, **N**atural **L**anguage **P**rocessing Lab) system for WASSA 2023 shared task 1. A unified framework is used to integrate the original RoBERTa transformer (Liu et al., 2019) with different sentiment-enhanced versions, including RoBERTa-Twitter (Barbieri et l., 2020) and EmoBERTa (Kim and Vossen, 2021) for both tracks. For Track 1, our best submission achieved an average Pearson correlation of 0.7236 and ranked fourth among all participating teams. For Track 2, our best result had an average Pearson correlation of 0.4178, ranking first among all nine submissions.

The rest of this paper is organized as follows. Section 2 describes the NCUEE-NLP system for Tracks 1 and 2 in the WASSA 2023 shared task 1. Section 3 presents the results and performance comparisons. Conclusions are finally drawn in Section 4.

2 The NCUEE-NLP System

We propose a unified architecture of ensemble neural networks to solve Tracks 1 and 2 of WASSA-2023 shared task 1. Figure 1 shows our system architecture for empathy and emotion prediction, which mainly depends on ensemble sentiment-enhanced transformers. We select the following RoBERTa-based transformers to tackle both tracks in this task.

(1) Robust optimized BERT pre-training approach (RoBERTa) (Liu et al., 2019)

RoBERTa is a replication study of BERT (Devlin et al., 2019) pre-training that carefully measures the impact of key parameters and training data size. The model modifications include removing the next sentence predictions,

RoBERTa Transformers	**Track 1: CONV** at the speech-turn level				**Track 2: EMP** at the essay-level		
	Empathy	Emotion Polarity	Emotion Intensity	Average	Empathy	Distress	Average
RoBERTa	0.7715	0.7608	0.6941	0.7421	0.6660	0.5596	0.6128
RoBERTa-twitter	0.7871	0.7671	0.7061	0.7534	0.6000	0.5564	0.5782
EmoBERTa	0.7693	0.7659	0.6899	0.7417	0.6278	0.5454	0.5866
Ensemble	**0.7901**	**0.7751**	**0.7076**	**0.7576**	**0.6702**	**0.5905**	**0.6304**

Table 1: Submission results on the validation set.

dynamically changing the masking pattern applied to the training data, and training with large batches.

(2) RoBERTa-Twitter (Barbieri et al., 2020)

RoBERTa-Twitter is a RoBERTa model trained on nearly 58M tweets and fine-tuned for sentiment analysis using the TweetEval benchmark.

(3) EmoBERTa (Kim and Vossen, 2021)

EmoBERT is a RoBERTa model trained to solve emotion recognition in conversation tasks. EmoBERTa can learn speaker-aware states and contexts to predict the emotion of a current speaker by simply prepending speaker names to utterances and inserting separation tokens between the utterances in a dialogue.

For both tracks in this shared task, we fine-tune these pre-trained RoBERTa-based transformers using the datasets provided by task organizers. For Track 1 on empathy and emotion prediction in conversations, we separately fine-tuned these transformers for empathy, emotion polarity and emotion intensity prediction. For Track 2 on empathy prediction at the essay-level, we respectively trained the transformers for empathy and distress score prediction.

Finally, we use the average ensemble mechanism to combine these individual sentiment-enhanced RoBERTa transformer to produce a desired score output for both tracks.

3 Evaluation

3.1 Datasets

The experimental datasets were provided by task organizers (Barriere et al., 2023). During system development phase, the training and validation sets respectively consisted of 8,776 and 2,400 conversations for Track 1. In addition, the training and validation sets for Track 2 respectively feature 792 and 208 essays. During the evaluation period, the test sets contain 1425 conversations for Track 1 and 100 essays for Track 2.

3.2 Settings

The pre-trained RoBERTa transformers models were download from HuggingFace[1]. The hyper-parameter values for our model implementation were used as follows: epoch 25, batch size 8, learning rate 1e-5, and max sequence 256.

To confirm the average ensemble performance, we also compared individual transformers. The evaluation metric is the Pearson correlation for both tracks. For Track 1, we obtained Pearson correlation coefficients of the empathy, emotion polarity and emotion intensity at the speech-turn level. For Track 2, we had Pearson correlation coefficients for empathy and distress at the essay-level. The official ranking of each participating team was based on the average of the obtained Pearson correlation coefficients.

3.3 Results

Table 1 shows the results on the validation set. For Track 1 at the speech-turn level, RoBERTa-Twitter outperformed the other standalone transformer models for all evaluation metrics, but relatively underperformed for Track 2 at the essay level. The ensemble transformers clearly achieved the best

---

[1] https://huggingface.co/roberta-base
https://huggingface.co/tae898/emoberta-base
https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment

RoBERTa Transformers	**Track 1: CONV** at the speech-turn level				**Track 2: EMP** at the essay-level		
	Empathy	Emotion Polarity	Emotion Intensity	Average	Empathy	Distress	Average
RoBERTa	0.7849	0.6851	0.6384	0.7028	0.3327	0.3819	0.3573
RoBERTa-twitter	0.7898	0.6941	**0.676**	0.72	0.3661	0.3415	0.3538
EmoBERTa	0.772	0.6638	0.6418	0.6925	0.4074	**0.4663**	**0.4368**
Ensemble (Official submission)	**0.8035**	**0.6981**	0.6692	**0.7236**	**0.415**	0.4206	0.4178

Table 2: Submission results on the test set.

performance for both tracks on the validation set. This confirms that the ensemble averaging mechanism works well in integrating multiple models to obtain performance improvement.

Table 2 shows the results on the test set for both tracks. For CONV Track 1, RoBERTa-twitter outperformed the others in the emotion intensity evaluation at the speech-turn level. Our ensemble sentiment-enhanced RoBERTa model achieved the best average Pearson correlation coefficient of 0.7236. For EMP Track 2, EmoBERTa obtained the best distress and average correlation coefficients, while our ensemble transformer model achieved the second-best correlation coefficient of 0.4178.

3.4 Rankings

According to official rankings released by task organizers (Barriere et al., 2023), our final submission from ensemble neural networks of sentiment-enhanced RoBERTa transformers ranked fourth for Track 1 and first for Track 2 among all nine submissions.

4 Conclusions

This study describes the model design, system implementation and performance of the NCUEE-NLP system in the WASSA 2023 Task 1 for empathy and emotion prediction. Our unified architecture used an average ensemble mechanism of three sentiment-enhanced RoBERTa transformers to predict empathy, emotion polarity and emotion intensity for Track 1 at the speech-turn level and empathy and distress scores for Track 2 at the essay-level. Our final submission based on sentiment-enhanced RoBERTa transformers ranked fourth for Track 1 and first for Track 2 among all nine submissions.

Acknowledgments

This study is partially supported by the National Science and Technology Council, Taiwan, under the grant MOST 111-2628-E-008-005-MY3.

References

Flor Miriam Del Arco, Jaime Collado-Montañez, L. Alfonso Ureña, and María-Teresa Martín-Valdivia. 2022. Empathy and distress prediction using transformer multi-output regression and emotion analysis with an ensemble of supervised and zero-shot learning models. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*. Association for Computational Linguistics, pages 239–244. https://doi.org/10.18653/v1/2022.wassa-1.23

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. TweetEval: unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. Association for Computational Linguistics, pages 1644–1650. https://doi.org/10.18653/v1/2020.findings-emnlp.148.

Valentin Barriere, Shabnam Tafreshi, João Sedoc, and Sawsan Alqahtani. 2022. WASSA 2022 shared task: predicting empathy, emotion and personality in reaction to news stories. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Association for Computational Linguistics, pages 214-227. http://dx.doi.org/10.18653/v1/2022.wassa-1.20

Valentin Barriere, Shabnam Tafreshi, João Sedoc, and Salvatore Giorgi. 2023. WASSA 2023 shared task: predicting empathy, emotion and personality in reactions and reaction to news stories. In *Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*.

Sven Buechel, Anneke Buffone, Barry Slaff, Lyle H. Ungar, and João Sedoc. 2018. Modeling empathy and distress in reaction to news stories. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 4758–4765. https://doi.org/10.18653/v1/D18-1507.

Yue Chen, Yingnan Ju, and Sandra Kübler. 2022. IUCL at WASSA 2022 shared task: a text-only approach to empathy and emotion detection. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*. Association for Computational Linguistics, pages 228–232. https://doi.org/10.18653/v1/2022.wassa-1.21

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, pages 4171-4186. http://dx.doi.org/10.18653/v1/N19-1423

Bhanu Prakash Reddy Guda, Aparna Garimella, and Niyati Chhaya. 2021. EmpathBERT: a BERT-based framework for demographic-aware empathy prediction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Association for Computational Linguistics, pages 3072–3079. https://doi.org/10.18653/v1/2021.eacl-main.268.

Taewoon Kim and Piek Vossen. 2021. EmoBERTa: speaker-aware emotion recognition in connersation with RoBERTa. *arXiv preprint*, https://arxiv.org/abs/2108.12009.

Atharva Kulkarni, Sunanda Somwase, Shivam Rajput, and Manisha Marathe. 2021. PVG at WASSA 2021: a multi-input, multi-task, transformer-based architecture for empathy and distress prediction. In *Proceedings of the 11th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Association for Computational Linguistics, pages 105–111.

Allison Lahnala, Charles Welch, and Lucie Flek. 2022. CAISA at WASSA 2022: adapter-tuning for empathy prediction. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*. Association for Computational Linguistics, pages 280–285. https://doi.org/10.18653/v1/2022.wassa-1.31

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: a robustly optimized BERT pretraining approach. *arXiv preprint*, https://arxiv.org/abs/1907.11692.

Jay Mundra, Rohan Gupta, and Sagnik Mukherjee. 2021. WASSA@IITK at WASSA 2021: multi-task learning and transformer finetuning for emotion classification and empathy prediction. In *Proceedings of the 11th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Association for Computational Linguistics, pages 112–116.

Damilola Omitaomu, Shabnam Tafreshi, Tingting Liu, Sven Buechel, Chris Callison-Burch, Johannes Eichstaedt, Lyle Ungar, and João Sedoc. Empathic conversations: a multi-level dataset of contextualized conversations. arXiv Preprint, arXiv:2205.12698. https://doi.org/10.48550/arXiv.2205.12698

João Sedoc, Sven Buechel, Yehonathan Nachmany, Anneke Buffone, and Lyle Ungar. 2020. Learning word ratings for empathy and distress from document-level user responses. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*. European Language Resources Association, pages 1664–1673.

Ashish Sharma, Adam Miner, David Atkins, and Tim Althoff. 2020. A computational approach to understanding empathy expressed in text-based mental health support. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, pages 5263–5276. https://doi.org/10.18653/v1/2020.emnlp-main.425.

Shabnam Tafreshi, Orphee De Clercq, Valentin Barriere, Sven Buechel, João Sedoc, Alexandra Balahur. 2021. WASSA 2021 shared task: predicting empathy and emotion in reaction to news stories. In *Proceedings of the 11th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. pages 92–104.

Naitian Zhou and David Jurgens. 2020. Condolence and empathy in online communities. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, pages 609–626. https://doi.org/10.18653/v1/2020.emnlp-main.45

CAISA at WASSA 2023 Shared Task: Domain Transfer for Empathy, Distress, and Personality Prediction

Fabio Gruschka and **Allison Lahnala** and **Charles Welch** and **Lucie Flek**
Conversational AI and Social Analytics (CAISA) Lab
Department of Mathematics and Computer Science, University of Marburg
`http://caisa-lab.github.io`

Abstract

This research contributes to the task of predicting empathy and personality traits within dialogue, an important aspect of natural language processing, as part of our experimental work for the WASSA 2023 Empathy and Emotion Shared Task. For predicting empathy, emotion polarity, and emotion intensity on turns within a dialogue, we employ adapters trained on social media interactions labeled with empathy ratings in a stacked composition with the target task adapters. Furthermore, we embed demographic information to predict Interpersonal Reactivity Index (IRI) subscales and Big Five Personality Traits utilizing BERT-based models. The results from our study provide valuable insights, contributing to advancements in understanding human behavior and interaction through text. Our team ranked 2nd on the personality and empathy prediction tasks, 4th on the interpersonal reactivity index, and 6th on the conversational task.

1 Introduction

Empathy, a fundamental component of interpersonal communication, emerges in broad spectrum of conversational and discourse settings, ranging from informal dialogues to purpose-driven conversations. With an invaluable role for cooperative interactions, modeling empathetic language is a growing area in natural language processing (NLP) research, enabling the improvement of dialogue agent experiences, analysis of online supportive interactions, and the development of educational tools (Rashkin et al., 2019; Lin et al., 2020; Majumder et al., 2020). Despite its significance, predicting empathy poses an immense challenge due to the scarcity of resources and the complexities involved in establishing a gold standard for this nuanced phenomenon (Omitaomu et al., 2022).

A major hindrance to empathy prediction in the NLP field is the lack of accessible, high-quality datasets. Many studies are conducted on sensitive data, which cannot be disclosed publicly (Lahnala et al., 2022b). Though some publicly available datasets exist, built on social media platforms or through specific data collection tasks, they are sparse and each comes with its inherent limitations due to the challenges in the data collection and annotation process (Omitaomu et al., 2022).

Defining empathy in a concrete, measurable way for consistent and relevant gold standard annotations is another formidable challenge. Empathy definitions vary greatly across psychological research, and NLP datasets are often annotated by third parties, rather than the individuals experiencing or receiving empathy. This approach captures specific language aspects identified by an external observer but fails to provide insight into how particular empathetic experiences influence language (Buechel et al., 2018).

In parallel to empathy, personality traits are fundamental to interpersonal relationships and social interactions. The Big Five Personality model (PER) (McCrae and Costa, 1992), a widely studied framework, is instrumental in understanding human social behavior. A comprehensive understanding of human behavior can be achieved by combining empathy, gauged by the Interpersonal Reactivity Index (IRI) (Davis, 1980), and personality traits.

This paper presents our experimental work on the WASSA 2023 Empathy and Emotion Shared Task (Barriere et al., 2023). We predict perceived empathy, emotion polarity, and emotion intensity at the turn-level in a conversation (the CONV task), and empathy and distress at the essay level (the EMP task). We explore the use of adapters, which provide greater efficiency compared to complete fine-tuning, and an adjusted approach derived by Sharma et al. (2020) at the essay level. Additionally, we embed demographic information to predict IRI subscales (Perspective Taking, Personal Distress, Fantasy, and Empathic Concern) and the Big

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 553–557
July 14, 2023 ©2023 Association for Computational Linguistics

Five Personality Traits (Conscientiousness, Openness, Extraversion, Agreeableness, and Stability) using BERT-based models (Devlin et al., 2019).

The paper is structured as follows: Section 2 reviews the task and dataset, provides an overview of the IRI, PER constructs, and empathy in conversations, emphasizing their significance in understanding human behavior. Section 3 describes our implementation. Section 4 presents the results and discussion of our findings. Finally, Section 5 concludes the paper and proposes directions for future research.

2 Task and Dataset

In this section, we describe the dataset and tasks employed in our research aimed at predicting empathy, distress, personality traits, and IRI subscales using the dataset provided by Omitaomu et al. (2022).

The dataset utilized in our experiments comprises empathetic reactions captured in essays and conversation responses to news articles involving harm to a person, group, or other entities. These reactions are documented in essays that range between 300 and 800 characters in length, as well as in conversations that consist of an average of 23 speech turns. The dataset also includes the original news articles and demographic information at the person-level, such as age, gender, ethnicity, income, and education level.

Each essay in the dataset is supplemented with Batson et al. (1987)'s empathic concern and personal distress scores, providing an insight into the individual's empathetic response after engaging with the news article. In addition, the dataset provides McCrae and Costa (1992)'s Big Five personality scores and Inter-Personal Reactivity Index (IRI) scores for the respective user, further enhancing our understanding of their empathetic capacity.

The Big Five of Costa and McCrae (1992) was shown to predict many traits about people, their behavior, and relationships. Each dimension can be rated on a continuous scale, where a person has more or a lesser degree of the qualities associated with that dimension. The following facets are from John et al. (1999):

1. **Neuroticism** - Anxiety, angry hostility, depression, self-consciousness, impulsiveness, vulnerability
2. **Extraversion** - Warmth, gregariousness, assertiveness, activity, excitement seeking, positive emotions
3. **Openness** - Fantasy, aesthetics, feelings, actions, ideas, values
4. **Agreeableness** - Trust, straightforwardness, altruism, compliance, modesty, tendermindedness
5. **Conscientiousness** Competence, order, dutifulness, achievement striving, self-discipline, deliberation

The IRI index is discussed in Davis (1983), who constructed a 28-item survey to measure four aspects of empathy using a 5-point Likert scale. The items (directly taken from the paper) are as follows:

1. **Perspective Taking** – the tendency to spontaneously adopt the psychological point of view of others.
2. **Fantasy** – taps respondents' tendencies to transpose themselves imaginatively into the feelings and actions of fictitious characters in books, movies, and plays.
3. **Empathic Concern** – assesses "other-oriented" feelings of sympathy and concern for unfortunate others.
4. **Personal Distress** – measures "self-oriented" feelings of personal anxiety and unease in tense interpersonal settings.

In the case of conversations, each speech turn has been annotated by a third person for perceived empathy, emotion polarity, and emotion intensity. This offers a comprehensive view of the interaction, enabling a detailed examination of the empathetic exchanges within the conversation.

The tasks of our research involve predicting empathy and emotion polarity & intensity on conversational turns (CONV), empathy and distress scores (EMP), personality traits (PER), and IRI subscale values (IRI). Systems are evaluated based on the Pearson's r correlation between the predicted and actual values in a test set, similar to the approach adopted in the previous edition of the shared task (Barriere et al., 2022).

3 System Description

3.1 Essay-Level Prediction

Domain Adapted Model. In our approach to predict empathy and distress at the essay-level, we adapt the classification model proposed by Sharma et al. (2020) to a regression model. Their original model, was designed for empathy classification, while our goal is to predict empathy and distress in essay texts. To achieve this, we make several modifications to the model, allowing it to handle essay-level predictions.

First, we normalize the labels for empathy and distress scores in the range of 0 to 1. This transformation enables the model to predict continuous values rather than categorical labels.

Next, we modify the model's architecture to accommodate the regression task. We replace the classification layer with a regression layer, which predicts continuous values instead of class probabilities. To train the modified model, we use the mean squared error (MSE) loss function, which measures the average squared differences between the predicted and true empathy and distress scores.

Finally, we fine-tune the adapted model on the Omitaomu et al. (2022) datasets, which contains essay texts along with their corresponding empathy and distress scores. The model learns to predict empathy and distress scores by leveraging the pretrained model's understanding of natural language and adjusting its weights based on the specific context of empathy and distress in essays.

Demographic Embeddings. The demographic embedding layer takes in one-hot encoded demographic information (i.e. gender, education, race, age) as well as income as a single number and concatenates it to the text encoding during the forward pass. This allows the model to utilize demographic features for each individual, which is particularly beneficial for tasks requiring personalized predictions. The demographic embedding layer is initialized using Xavier initialization and is updated during training (Glorot and Bengio, 2010) and has a dimension of 135. The BERT encodings of the article and essay are projected down to the same dimension before being concatenated and passed to the final classification layer.

3.2 Conversation-Level Emotion Prediction

Adapter-Tuning Framework. For our implementation, we use AdapterHub (Pfeiffer et al., 2020), a straightforward framework built on the HuggingFace transformers. We train adapters to predict a conversation's emotional polarity, emotional intensity, and empathy.

We employ a method inspired by the EPITOMEFUSION method implemented by Lahnala et al. (2022a). We used the model of Sharma et al. (2020), which is based on RoBERTa (Liu et al., 2019) to predict empathetic reactions, explorations, and interpretations. We finetuned separate adapters to categorize each of these aspects in the EPITOME dataset. Later, these adapters are merged using the AdapterFusion composition technique (Pfeiffer et al., 2021). An adapter for the prediction of empathy and distress in conversation was trained on top of this, with learning rate $1e^{-4}$, for 10 epochs. This configuration allows the combination of the knowledge from each of the pretrained adapters for the EPITOME tasks and their application in the conversation-level prediction tasks.

Attribute	**Pearson Correlation**
Personality (PER)	
Conscientiousness	0.3229
Openness	0.3273
Extraversion	-0.1966
Agreeableness	0.2900
Stability	0.1999
Interpersonal Reactivity Index (IRI)	
Perspective Taking	0.1582
Personal Distress	-0.1875
Fantasy	-0.0556
Empathic Concern	0.1804
Overall	0.0239
Empathy Prediction (EMP)	
Empathic Concern	0.3478
Personal Distress	0.4197
Overall	0.3840
Empathy & Emotion in Conversations (CONV)	
Emotion Polarity	0.7832
Emotion Intensity	0.6858
Empathy	0.6523
Overall	0.7071

Table 1: Pearson correlations for the personality, IRI, and empathy prediction tasks post-phase essay-level results with the task embedding model.

4 Results and Discussion

Our submissions to the post-evaluation phase on the test dataset have yielded promising results, as showcased in Table 1.

In the domain of personality prediction, as displayed in the top portion of Table 1, our task embedding model has performed particularly well in predicting certain aspects of the Big Five Personality traits. Specifically, it has demonstrated strong predictive power for the traits of Conscientiousness (r=0.3229), Openness (r=0.3273), and Agreeableness (r=0.29). However, the model has shown a negative correlation for Extraversion (r=-0.1966), indicating a need for further refinement in this area.

Turning to the Interpersonal Reactivity Index (IRI) prediction, as seen in the middle of Table 1, the performance of our model has been more varied. While the prediction of the Perspective Taking subscale showed a modest positive correlation (r=0.1582), Personal Distress exhibited a negative correlation (r=-0.1875). This might suggest that the model currently struggles with accurately capturing the nuances of distress experienced by individuals. The model also demonstrated a low correlation for the Fantasy (r=-0.0556) subscale, though our best performance was on the Empathic Concern (r=0.1804) subscale.

At the essay level, our approaches have shown encouraging results for empathy and distress prediction, as evidenced by the second to last portion of Table 1. The Domain Adapted (Sharma) approach, in particular, has excelled in this task, yielding an average Pearson correlation of 0.3478 for Empathy and a notable 0.4197 for Distress. These results underline the efficacy of this approach in gauging empathy and distress from written texts.

Lastly, as we move to the conversation level prediction in bottom of Table 1, our adapter approach has demonstrated satisfactory performance. The model has been particularly successful in predicting emotional polarity (r=0.7832), emotional intensity (r=0.6858), and empathy (r=0.6523) in the conversation. These results affirm the potential of our adapter approach in effectively capturing the empathetic and emotional dynamics within conversational exchanges. We believe there is much room for improvement at the conversation level and through the use of adapters. Our model was relatively simple and future work should explore other adapters and architectures to more effectively transfer knowledge from related tasks. We only took individual turns in the conversation into account and future work would benefit from providing the model with additional context from the conversation history.

5 Conclusion

In this paper, we have presented our methodologies and findings from predicting a range of empathy-related features in text, specifically in essays and conversation responses from the Omitaomu et al. (2022) dataset.

We developed and evaluated a series of models, each addressing unique aspects of the prediction tasks. At the essay level, we employed a domain-adapted model based on the work of Sharma et al. (2020), modified to perform regression instead of classification, effectively predicting empathy and distress scores.

Our results across different measures are encouraging. The demographic-embedding approach performed quite well in predicting the conscientiousness, openness, and agreeableness aspects of the Big Five Personality traits. In contrast, the performance on the Interpersonal Reactivity Indices was less impressive. The domain-adapted model excelled in predicting empathy and distress at the essay level. At the conversation level, our adapter approach achieved satisfactory results in predicting emotional polarity, emotional intensity, and empathy.

Code for our systems and experiments are publicly available at `https://github.com/caisa-lab/wassa-shared-task-2023`.

References

Valentin Barriere, Shabnam Tafreshi, João Sedoc, and Sawsan Alqahtani. 2022. Wassa 2022 shared task: Predicting empathy, emotion and personality in reaction to news stories. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 214–227.

Valentin Barriere, Shabnam Tafreshi, João Sedoc, and Salvatore Giorgi. 2023. Wassa 2023 shared task: Predicting empathy, emotion and personality in interactions and reaction to news stories. In *Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*.

C. Daniel Batson, Jim Fultz, and Patricia A. Schoenrade. 1987. Distress and empathy: two qualitatively distinct vicarious emotions with different motivational consequences. *Journal of personality*, 55 1:19–39.

Sven Buechel, Anneke Buffone, Barry Slaff, Lyle Ungar, and João Sedoc. 2018. Modeling empathy and distress in reaction to news stories. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4758–4765, Brussels, Belgium. Association for Computational Linguistics.

Paul T Costa and Robert R McCrae. 1992. Neo pi-r professional manual: Revised neo personality inventory (neo pi-r) and neo five-factor inventory (neo-ffi). *Odessa, FL: Psychological Assessment Resources*.

Mark H Davis. 1980. A multidimensional approach to individual differences in empathy. *JSAS Catalog of Selected Documents in Psychology*, 10:85.

Mark H Davis. 1983. Measuring individual differences in empathy: Evidence for a multidimensional approach. *Journal of personality and social psychology*, 44(1):113.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings.

Oliver P John, Sanjay Srivastava, et al. 1999. The big-five trait taxonomy: History, measurement, and theoretical perspectives.

Allison Lahnala, Charles Welch, and Lucie Flek. 2022a. CAISA at WASSA 2022: Adapter-tuning for empathy prediction. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 280–285, Dublin, Ireland. Association for Computational Linguistics.

Allison Lahnala, Charles Welch, David Jurgens, and Lucie Flek. 2022b. A critical reflection and forward perspective on empathy and natural language processing. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2139–2158, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zhaojiang Lin, Peng Xu, Genta Indra Winata, Farhad Bin Siddique, Zihan Liu, Jamin Shin, and Pascale Fung. 2020. Caire: An empathetic neural chatbot.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Navonil Majumder, Pengfei Hong, Shanshan Peng, Jiankun Lu, Deepanway Ghosal, Alexander Gelbukh, Rada Mihalcea, and Soujanya Poria. 2020. MIME: MIMicking emotions for empathetic response generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8968–8979, Online. Association for Computational Linguistics.

Robert R McCrae and Paul T Costa. 1992. The five-factor model of personality and its relevance to personality disorders. *Journal of Personality Disorders*, 6(4):343–359.

Damilola Omitaomu, Shabnam Tafreshi, Tingting Liu, Sven Buechel, Chris Callison-Burch, Johannes Eichstaedt, Lyle Ungar, and João Sedoc. 2022. Empathic conversations: A multi-level dataset of contextualized conversations.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2021. AdapterFusion: Non-destructive task composition for transfer learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 487–503, Online. Association for Computational Linguistics.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. AdapterHub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 46–54, Online. Association for Computational Linguistics.

Hannah Rashkin, Eric Michael Smith, Margaret Li, and Y-Lan Boureau. 2019. Towards empathetic open-domain conversation models: A new benchmark and dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5370–5381, Florence, Italy. Association for Computational Linguistics.

Ashish Sharma, Adam Miner, David Atkins, and Tim Althoff. 2020. A computational approach to understanding empathy expressed in text-based mental health support. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5263–5276, Online. Association for Computational Linguistics.

Converge at WASSA 2023 Empathy, Emotion and Personality Shared Task: A Transformer-based Approach for Multi-Label Emotion Classification

Aditya Paranjape, Gaurav Kolhatkar, Yash Patwardhan, Omkar Gokhale,
Shweta Dharmadhikari
Pune Institute of Computer Technology, Pune, India
`adifeb24@gmail.com`, `gauravk403@gmail.com`, `yash23pat@gmail.com`,
`omkargokhale2001@gmail.com`, `scdharmadhikari@pict.edu`

Abstract

In this paper, we highlight our approach for the "WASSA 2023 Empathy and Emotion Shared Task". We present our approach for track 3 of the shared task which aims to identify emotions from text. Each sample in the dataset has one or more labels making it a multi-label classification task. We compared multiple transformer-based models by fine-tuning them for multi-label classification. Oversampling was used to overcome the class imbalance in the dataset. Ensembling techniques were used to improve the performance of the system. We obtained a macro F1-score of 0.5649 using XLNet on the test dataset in the official phase and secured rank 6 on the official leaderboard. During the post-competition phase, a threshold-based voting mechanism was performed on three models (Longformer, BERT, BigBird) that yielded the highest overall macro F1-score of 0.6605.

1 Introduction

With a rapid increase in the technological and scientific advancements seen in the domains of Machine Learning and Deep Learning, machines can now easily perform complex tasks at a degree of proficiency comparable to that of humans. However, one aspect where machines fall short in performing human-like tasks is those which require the understanding and contextualization of emotions. Emotions can be broadly divided into primary and secondary emotions (Rodríguez-Torres et al., 2005). Primary emotions include but are not limited to joy, sadness, and anger; while secondary emotions are emotions that are caused by other emotions.

Emotion Classification is an approach that helps in identifying the emotional context of textual data. This classification can serve as a concise summary for the readers. Applications such as recommendation systems also benefit greatly when used in combination with emotion classification approaches. Based on the classification of the user query, potential recommendations (Barrière and Kembellec, 2018) are narrowed down for the user and help the model in finding the best response. Emotion classification plays a crucial role in bridging the gap between human-computer interaction.

Through this paper, we intend to examine the efficacy of several transformer-based models for producing competitive results for emotion classification. The texts on which the models are trained and tested are essays that are responses to news articles. The models perform multi-label classification to identify the emotions expressed in the essays.

2 Related Work

Ekman and Friesen (1986) suggested that there are a set of universal emotions, which include happiness, sadness, anger, fear, disgust, and surprise, that are expressed by all humans through specific facial expressions regardless of their cultural background.Darwin and Prodger (1998)'s investigation into the expression of emotion on the face and through body gestures in both humans and animals marked a pioneering moment in the science of emotion recognition and analysis. Emotions can be recognized primarily through three categories: facial expressions (Goldman and Sripada, 2005), voice (Koolagudi and Rao, 2012), and text (Thakur et al., 2018). The process of automatically tagging a text with an emotion from a list of predetermined emotion labels is known as emotion recognition in text.

Early research concentrated on a lexicon-based methodology (Pradhan et al., 2023) which establishes polarity or sentiment to classify emotions from a text as positive, negative, or neutral. This was followed by the introduction of keyword-based methodology (Tao, 2004; Ma et al., 2005) that involves locating keyword occurrences in a text and tagging each one with an emotion from an emotion dictionary. Subsequently, based on rule-based techniques, rule-based models (Lee et al., 2010; Udochukwu and He, 2015) were presented

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 558–563
July 14, 2023 ©2023 Association for Computational Linguistics

in which the rules for emotion detection were extracted from the preprocessed dataset and the best rule among them was selected for emotion labeling.

With the emergence of machine learning approaches that categorize text into multiple emotion categories, it has been observed that SVM (Desmet and Hoste, 2013) and Bayesian networks (Liew and Turtle, 2016) consistently produce good results. Several classification algorithms were evaluated for multi-label emotion recognition (Xu et al., 2018) and it was discovered that logistic regression produced the best results on the provided features. As research in the field of deep learning gained traction, various models for multi-label emotion recognition that used CNNs (Wang et al., 2016), DNNs (Du and Nie, 2018), LSTMs (Li et al., 2018) and Bi-LSTMs (Baziotis et al., 2018) were proposed. In addition to other deep learning ideas, transformer models like BERT (Devlin et al., 2018) were employed in a variety of applications to improve performance. The most popular deep learning methods, nevertheless, were those based on LSTM and its subtypes.

In order to produce accurate results for emotion detection tasks, numerous hybrid models (Park et al., 2018; Seol et al., 2008; Shaheen et al., 2014; De Bruyne et al., 2018) combining various strategies were proposed from the pool of methods developed for text-based emotion analysis. In this paper, we compare various transformer-based models for emotion classification and perform experiments on the same.

3 Dataset Description

The dataset provided for this task (Omitaomu et al., 2022; Barriere et al., 2023) comprised essays that were written in response to news articles. The essays vary in length, ranging from 300 characters to 800 characters. The training data had 792 samples of such essays, the development data contained 208 samples, and the test data comprised 100 samples. The training data contained features like the essay, article-id, speaker-id, gender, education, etc. This shared task problem falls under the category of multi-label classification. There are 8 base emotions or labels (Anger, Hope, Sadness, Neutral, Disgust, Surprise, Joy, Fear) and each essay in the dataset is assigned one or more of these labels. The class of 'Sadness' had the highest number of samples in the training data, with 297 samples. Whereas, the class 'Joy' had the least number of samples in the training data, with only 5 samples.

4 Methodology

First, we evaluate and compare the performance of different models on the test dataset based on their Macro F1-score and Micro F1-score metrics. These models are listed and explained below. We finetune these models on the training dataset using the standard procedure for multi-label classification. We use a threshold value of 0.37 to decide whether a label should be assigned to a particular example. If the probability output for a certain label is greater than the threshold, then that label is selected. All the models were trained for 12 epochs (except for Longformer, which was trained for 10 epochs) with a learning rate of 4e-5. The results obtained in the post-competition phase have been showcased in Table 1. The official phase score for XLNet is also mentioned in Table 1.

4.1 Longformer

Longformer (Beltagy et al., 2020) is a transformer-based model that is useful for tasks that require processing long sequences of text. Longformer uses a modified attention mechanism that scales linearly with the input size, as opposed to the quadratic time taken by the traditional attention mechanism. It achieves this by using a combination of local and global attention.

4.2 BERT

BERT is a language representation model. It is used to obtain bidirectional representations of text input, which yield state-of-the-art results on many NLP tasks.

4.3 XLNet

XLNet (Yang et al., 2019)is an autoregressive pretraining technique that improves on the deficiencies of BERT. XLNet uses a Permutation Language Modelling objective, to help understand the bidirectional context. The model outperforms BERT on several NLP tasks.

4.4 BigBird

BigBird (Zaheer et al., 2020) is a BERT-like model that is useful for longer input sequences. It replaces the self-attention mechanism in BERT with a combination of sparse, global, and random attention. This requires much lesser computational power while giving a comparable performance.

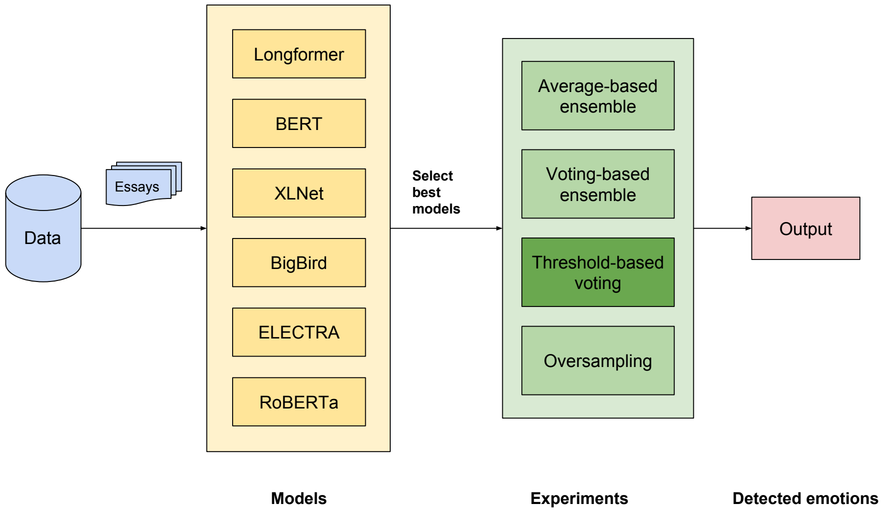


Figure 1: Methodology

4.5 ELECTRA

ELECTRA (Clark et al., 2020) is a pre-training method that aims to use significantly fewer compute resources than an MLM pre-training method. The pre-training stage involves training two transformer models: the generator and the discriminator. The discriminator model is further used on downstream tasks.

4.6 RoBERTa

RoBERTa (Liu et al., 2019) improves on the BERT model by making some important tweaks to the hyperparameters. It removes the next sentence prediction pre-training objective and uses much larger mini-batch sizes and learning rates.

Model name	Macro F1 score	Micro F1 score
XLNet* (Official)	0.5649	0.7009
XLNet (Post-Competition)	0.5927	0.7018
RoBERTa	0.5716	0.6937
BERT	0.6308	0.7039
BigBird	0.6281	0.7074
Electra	0.5860	0.7167
LongFormer	**0.6360**	**0.7289**

Table 1: Vanilla Model outputs (Post-Competition)
* Official result was submitted on the official leaderboard and was trained with a higher learning rate.

5 Experiments

5.1 Ensemble

Based on our results on the test data, we ensemble the top models by using three strategies as shown in Figure 1.

5.1.1 Voting

We calculate the outputs for each sample using all 3 models. We then take a vote between the models to determine the actual output. If all three models give different outputs, preference is given to the top model. In this case, the top 3 models are Longformer, Bigbird, and BERT, with the highest preference given to Longformer. We repeat this process for the top 5 models which are Longformer, Bigbird, BERT, XLNet, and RoBERTa.

5.1.2 Averaging

We average the individual probability values for each class obtained from the top 3 models and then determine the output label/labels for each sample based on the 0.37 threshold mentioned in 4. We repeat this process with the top 5 models and compare the results.

5.1.3 Threshold-based voting

We observed that the previous voting strategies seem to fail for samples having ground truths that consist of multiple labels. To counter this we implement a threshold-based strategy. This strategy is implemented on an ensemble of the top 3 as well as the top 5 models.

Top 3 models:

We implement voting with an extra stipulation that if a model predicts a label with a confidence higher than 0.55, then its label is retained irrespective of whether it wins or loses the vote.

Top 5 models:

In this ensemble, we add two stipulations to the voting process. First, if two models predict the same label with a confidence higher than 0.5 then that label is retained. Second, if a single model predicts a label with confidence higher than 0.75, then that label is retained.

Experiment	**Models used**	**Macro F1 score**	**Micro F1 score**
Average-based ensemble	Top 3	0.5695	0.6953
Voting-based ensemble	Top 3	0.5683	0.6926
Threshold based voting	Top 3	**0.6605**	**0.7236**
Average-based ensemble	Top 5	0.6098	0.7094
Voting-based ensemble	Top 5	0.561	0.693
Threshold based voting	Top 5	0.6104	0.6917
Oversampling	Long-former	0.4653	0.6784

Table 2: Post-competition results in the test dataset (Top 3: Longformer, BERT, BigBird, Top 5: Top 3 + XLNet, ELECTRA

5.2 Oversampling

There is a significant class imbalance in the data. To counter this we implement oversampling. Here, we duplicate samples from classes having less number of samples. The end goal is to have an equal number of samples for each class. In our dataset, class 'Sadness' has 292 samples which is the highest number of samples. So, we oversample the other classes such that each class has 292 samples.

6 Results

We hereby analyze the results shown in Table 2. We make some key observations regarding the results as follows:

Longformer is the best standalone model:

Out of all the vanilla models we trained, we see that 'Longformer' performs best with a macro F1-score of 0.6360. In the provided dataset, the average number of words per essay is 86. Since Longformer works well for long input sequences, as is seen in the provided training dataset, it outperforms the other models.

Oversampling yields no performance improvement:

We observed that oversampling leads to a significant decrease in macro F1-score, obtaining a score of 0.4653. Further investigation is required to explain this discrepancy.

Ensembling significantly improves results:

Both the approaches provided competitive results, however threshold-based voting with three models(Longformer, BERT, BigBird) gives the best overall macro F1-score score of 0.6605. Average-based Ensemble with five models(Longformer, BERT, BigBird, XLNet, ELECTRA) also provides good results with a macro F1-score of 0.6098.

7 Conclusion

In this paper, we compared the performance of six transformer-based models (Longformer, BERT, BigBird, XLNet, ELECTRA, RoBERTa) for emotion classification on the test dataset. Our official macro F1-score in the official phase was 0.5649, which was obtained on XLNet. Further, many improvements were made in the scores in the post-competition phase. It was observed that Longformer outperformed all other models with a macro F1-score of 0.636. We conducted multiple experiments by employing ensembling and oversampling techniques which concluded that the threshold-based voting method yields the best performance with a macro F1-score of 0.6605. In the future, we plan to improve our oversampling score and combine it with threshold-based voting.

References

Valentin Barrière and Gérald Kembellec. 2018. Short review of sentiment-based recommender systems. In *Proceedings of the 1st International Conference on Digital Tools & Uses Congress*, pages 1–4.

Valentin Barriere, Shabnam Tafreshi, Jo ao Sedoc, and Salvatore Giorgi. 2023. Wassa 2023 shared task: Predicting empathy, emotion and personality in interactions and reaction to news stories. In *Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment Social Media Analysis*.

Christos Baziotis, Nikos Athanasiou, Alexandra Chronopoulou, Athanasia Kolovou, Georgios Paraskevopoulos, Nikolaos Ellinas, Shrikanth Narayanan, and Alexandros Potamianos. 2018. Ntua-slp at semeval-2018 task 1: Predicting affective content in tweets with deep attentive rnns and transfer learning. *arXiv preprint arXiv:1804.06658*.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Charles Darwin and Phillip Prodger. 1998. *The expression of the emotions in man and animals*. Oxford University Press, USA.

Luna De Bruyne, Orphée De Clercq, and Véronique Hoste. 2018. Lt3 at semeval-2018 task 1: A classifier chain to detect emotions in tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 123–127.

Bart Desmet and Véronique Hoste. 2013. Emotion detection in suicide notes. *Expert Systems with Applications*, 40(16):6351–6358.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Pan Du and Jian-Yun Nie. 2018. Mutux at semeval-2018 task 1: exploring impacts of context information on emotion detection. In *Proceedings of the 12th international workshop on semantic evaluation*, pages 345–349.

Paul Ekman and Wallace V Friesen. 1986. A new pan-cultural facial expression of emotion. *Motivation and emotion*, 10:159–168.

Alvin I Goldman and Chandra Sekhar Sripada. 2005. Simulationist models of face-based emotion recognition. *Cognition*, 94(3):193–213.

Shashidhar G Koolagudi and K Sreenivasa Rao. 2012. Emotion recognition from speech: a review. *International journal of speech technology*, 15:99–117.

Sophia Yat Mei Lee, Ying Chen, and Chu-Ren Huang. 2010. A text-driven rule-based system for emotion cause detection. In *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*, pages 45–53.

Meng Li, Zhenyuan Dong, Zhihao Fan, Kongming Meng, Jinghua Cao, Guanqi Ding, Yuhan Liu, Jiawei Shan, and Binyang Li. 2018. Isclab at semeval-2018 task 1: Uir-miner for affect in tweets. In *Proceedings of the 12th international workshop on semantic evaluation*, pages 286–290.

Jasy Suet Yan Liew and Howard R Turtle. 2016. Exploring fine-grained emotion detection in tweets. In *Proceedings of the NAACL student research workshop*, pages 73–80.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Chunling Ma, Helmut Prendinger, and Mitsuru Ishizuka. 2005. Emotion estimation and reasoning based on affective textual interaction. In *Affective Computing and Intelligent Interaction: First International Conference, ACII 2005, Beijing, China, October 22-24, 2005. Proceedings 1*, pages 622–628. Springer.

Damilola Omitaomu, Shabnam Tafreshi, Tingting Liu, Sven Buechel, Chris Callison-Burch, Johannes Eichstaedt, Lyle Ungar, and Jo ao Sedoc. 2022. Empathic conversations: A multi-level dataset of contextualized conversations. *arXiv preprint arXiv:2205.12698*.

Ji Ho Park, Peng Xu, and Pascale Fung. 2018. Plusemo2vec at semeval-2018 task 1: Exploiting emotion knowledge from emoji and# hashtags. *arXiv preprint arXiv:1804.08280*.

Anima Pradhan, Manas Ranjan Senapati, and Pradip Kumar Sahu. 2023. Comparative analysis of lexicon-based emotion recognition of text. In *Machine Learning, Image Processing, Network Security and Data Sciences: Select Proceedings of 3rd International Conference on MIND 2021*, pages 671–677. Springer.

Ramón Rodríguez-Torres, Jacques Philippe Leyens, Armando Rodríguez Pérez, Verónica Betancor Rodriguez, María N Quiles del Castillo, Stéphanie Demoulin, and Brezo Cortés. 2005. The lay distinction between primary and secondary emotions: A spontaneous categorization? *International Journal of Psychology*, 40(2):100–107.

Yong-Soo Seol, Dong-Joo Kim, and Han-Woo Kim. 2008. Emotion recognition from text using knowledge-based ann. In *ITC-CSCC: International Technical Conference on Circuits Systems, Computers and Communications*, pages 1569–1572.

Shadi Shaheen, Wassim El-Hajj, Hazem Hajj, and Shady Elbassuoni. 2014. Emotion recognition from text based on automatically generated rules. In *2014 IEEE International Conference on Data Mining Workshop*, pages 383–392. IEEE.

Jianhua Tao. 2004. Context based emotion detection from text input. In *Eighth International Conference on Spoken Language Processing*.

Priyanka Thakur, Dr Rajiv Shrivastava, and A DR. 2018. A review on text based emotion recognition system. *International Journal of Advanced Trends in Computer Science and Engineering*, 7(5).

Orizu Udochukwu and Yulan He. 2015. A rule-based approach to implicit emotion detection in text. In *Natural Language Processing and Information Systems: 20th International Conference on Applications of Natural Language to Information Systems, NLDB 2015, Passau, Germany, June 17-19, 2015, Proceedings 20*, pages 197–203. Springer.

Yaqi Wang, Shi Feng, Daling Wang, Ge Yu, and Yifei Zhang. 2016. Multi-label chinese microblog emotion classification via convolutional neural network. In *Web Technologies and Applications: 18th Asia-Pacific Web Conference, APWeb 2016, Suzhou, China, September 23-25, 2016. Proceedings, Part I*, pages 567–580. Springer.

Huimin Xu, Man Lan, and Yuanbin Wu. 2018. Ecnu at semeval-2018 task 1: Emotion intensity prediction using effective features and machine learning models. In *Proceedings of the 12th international workshop on semantic evaluation*, pages 231–235.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297.

PICT-CLRL at WASSA 2023 Empathy, Emotion and Personality Shared Task: Empathy and Distress Detection using Ensembles of Transformer Models

Tanmay Chavan[*] , **Kshitij Deshpande**[*] and **Sheetal Sonawane**[*]
Pune Institute of Computer Technology, Pune
`{chavantanmay1402, kshitij.deshpande7}@gmail.com, sssonawane@pict.edu`

Abstract

This paper presents our approach for the WASSA 2023 Empathy, Emotion and Personality Shared Task. Empathy and distress are human feelings that are implicitly expressed in natural discourses. Empathy and distress detection are crucial challenges in Natural Language Processing that can aid our understanding of conversations. The provided dataset consists of several long-text examples in the English language, with each example associated with a numeric score for empathy and distress. We experiment with several BERT-based models as a part of our approach. We also try various ensemble methods. Our final submission has a Pearson's r score of 0.346, placing us third in the empathy and distress detection subtask.

1 Introduction

Empathy and distress are important attributes in natural language processing which allow a better understanding of the nature of human interactions. However, they are difficult to quantify and as a result, are not as deeply researched as fields like hate and sentiment detection. Nevertheless, they are very beneficial in comprehending useful information. Thus, there is a huge scope for work to be done in this domain. Empathy is an emotion that enables us to grasp the emotional and mental state of others and thus is very crucial to conversations. It helps foster deeper social connections and promotes amicability. Hence, precisely identifying empathy is very beneficial. On the other hand, distress is an emotion that acts as a vital sign that suggests a possible threat or harm (Viertiö et al., 2020). It thus helps identify discomfort and thereby makes efforts to allay any suffering that may have resulted. Thus, accurate distress detection helps promote well-being and peace in society.

The WASSA 2023 Empathy, Emotion and Personality Shared Task Barriere et al. (2023) was to perform Empathy Detection, Emotion Classification, and Personality Detection in Interactions. Our team, PICT-CLRL, participated under the Codalab username *earendil* in the shared task. 5 tracks, namely Empathy and Emotion Prediction in Conversations (CONV), Empathy Prediction (EMP), Emotion Classification (EMO), Personality Prediction (PER), and Interpersonal Reactivity Index Prediction (IRI) were offered. This paper demonstrates our work on Task 2 Empathy Prediction (EMP). Specifically, Task 2 was to predict the empathy and distress levels at an essay level. The essays supplied in the dataset are between 300 and 800 characters in length. These essays are written in response to news stories about people, groups, or other entities facing some sort of plights. Various attributes related to an individual demographic like age, gender, ethnicity, income, and education level are also provided. The average Pearson correlation for empathy and distress was considered the official metric for evaluation.

Recently, transformer-based models Vaswani et al. (2017) have achieved great success in several NLP-related tasks. BERT Devlin et al. (2019) has achieved State-of-the-art results on several benchmarks. Furthermore, several BERT-based models with additional pre-training have also produced excellent performance. The selection of pre-training data can aid domain-specific tasks as well. We experiment with several such models. In addition to this, we also try ensemble-based approaches. Ensembling can produce better results than individual models by combining model outputs in an effective manner. We evaluate these approaches and present the results and observations.

2 Related Work

Litvak et al. (2016) identifies that monitoring social and linguistic behavior through empathy monitoring hasn't gained much attention and that there is a huge scope for further research. To determine

---

[*] Equal contribution

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 564–568
July 14, 2023 ©2023 Association for Computational Linguistics

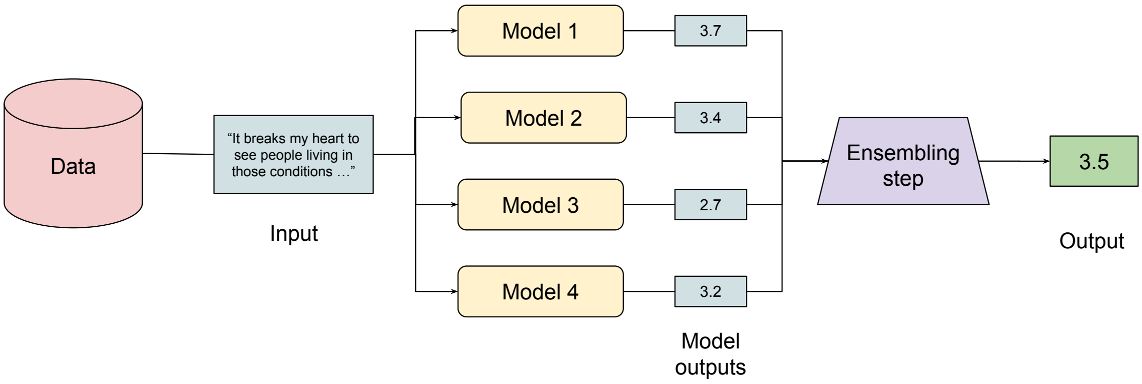


Figure 1: Our ensembling system. We use the same configuration for empathy and distress.

how social and linguistic behavior relates to the attribute of empathy, a Poisson regression model has been utilized. To better comprehend empathy, Davis' Interpersonal Reactivity Index (IRI), which takes into account 4 factors (namely fantasy, personal distress, empathetic concern, and perspective taking) has been used.

Gibson et al. (2016) utilizes empathy in addiction counseling. The transcripts of the session conducted are used to train a model and predict empathy. Naturally, high empathy is desirable from a counselor toward the client. The model is trained in two parts, firstly a Recurrent Neural Network is trained on a group of certain behavioral acts and then this is used to train the final Deep Neural Network. This approach is shown to have produced better results than training the Deep neural network all at once.

Hosseini and Caragea (2021) identifies that it can be difficult to annotate data to identify empathy when working on a large scale. To integrate knowledge from the available resources and detect empathy from the natural language in several domains, this study uses multi-task training with knowledge distillation. Results on the TwittEmp dataset are shown to produce significantly better results using this approach.

Saleem et al. (2012) recognizes that psychological distress is seldom sufficiently identified. It offers ways to detect distress indicators and assess the severity of the distress. Text from online forum posts where individuals discuss their thoughts more freely is used. SVMs are used to identify distress indicators.

Guda et al. (2021) utilizes user demographic to create an EMPATH-BERT framework for empathy detection. Internally it uses the BERT model, the framework is shown to surpass existing machine learning techniques. This paper allows us to understand the important role of demographic information in empathy detection.

Barriere et al. (2022) summarizes the previous edition of the shared task and covers several approaches for the problem of empathy and distress prediction.

3 Data

We use the *Empathic Conversations* dataset presented by Omitaomu et al. (2022) for the task. The original data was split into 72%, 18.9%, and 9.1% for train, dev, and test respectively. The data splits are summarized in Table 1. Compared to the conversations dataset, the essay dataset is relatively smaller. The data present consists of 24 attributes. The training dataset has 792 examples. The validation set provided has 208 samples. Lastly, the testing dataset has 100 samples. The data consists of various individual demographic attributes like age, gender, and education along with essays that were composed in response to news stories about people experiencing hardships.

4 System Overview

The shared task consists of a regression problem, where we have to predict a numerical value given an essay. BERT-based approaches are very successful at these types of problems. We try out several

Dataset	Number of examples
Training	792
Validation	208
Testing	100

Table 1: The dataset's training, validation, and test splits.

such LLMs. We also utilize various ensembling techniques. We briefly summarize our efforts in the following section.

4.1 BERT-based Models

We experiment with several pre-trained BERT-based models for the task. We evaluate their performances and select the models with the best performance on our dataset. RoBERTa[1] (Liu et al., 2019) is a BERT-based model with additional pre-training. It is pre-trained on 5 different English datasets, totaling a size of almost 160 GB. This is a vast improvement over BERT, which is pre-trained on two datasets totaling a size of about 16 GB. The authors claim that the additional pre-training results in improved performance, which is heavily supported by empirical evaluations.

We also use Twitter-RoBERTa-emotion[2] (Barbieri et al., 2020). This model is a RoBERTa model which is pre-trained on roughly 58 million tweets. The model is also fine-tuned on the TweetEval benchmark datasets for emotion classification.

Twitter-RoBERTa-sentiment[3] is another RoBERTa-based model used in our experiments. It pre-trains the roBERTa-base model on Twitter corpora composed of around 124 million tweets. The model is additionally fine-tuned on the TweetEval benchmark for sentiment analysis. The pre-training data for the model consists of tweets posted over a span of roughly 4 years, thus encompassing data spanning over a significant period of time.

The unsupervised SimCSE[4] (Gao et al., 2022) model uses sentence embeddings instead of token embeddings. It is trained in an unsupervised format and makes predictions on input sentences with a contrastive learning framework. Sentence-embedding-based models can potentially perform very well on long text-document classification tasks.

All of the above models are freely available on HuggingFace. We have tagged the model names with their links in the footnotes.

4.2 Ensemble Methods

Ensembling involves combining predictions by several individual models using various statistical and non-statistical-based approaches to enhance results. Ensembling can often result in better performances than individual models despite using the same data. We explore several ensembling approaches and elaborate on them.

The simplest ensembling method is to calculate average individual predictions and present them as the final output. Although this method might not utilize specific trends within the data, it generates stable predictions with low variance.

We also use machine learning algorithms for ensembling outputs generated by the four models. We try linear regression and Support Vector Regression, as implemented in the sklearn module. We also use the XGBoost algorithm, an efficient variant of gradient boosting. Figure 1 illustrates our ensemble system.

5 Results

The results of our experiments are discussed in the following section. We report the results of the models and the ensembles in table 2. The official scoring metric of the shared task for the EMP track is Pearson's r. The final rank is determined by the average value of Pearson's correlations of empathy and distress.

We can see that the RoBERTa-base model has the highest score for predicting empathy. The unsupervised SimCSE performs best at predicting distress. Overall, the unsupervised SimCSE model has the best performance, with an average Pearson correlation of 0.352. The excellent performance of the SimCSE model is suggestive of the benefits of using sentence embeddings for long-text documents, in addition to better pre-training. It can also be observed that the Twitter-RoBERTa models do not exhibit significantly better performance than RoBERTA-base. This may be due to the fact that the Twitter pre-training data consisting of casually-written tweets, is significantly different in nature than the dataset responses which are properly and

---

[1] Model link: https://huggingface.co/roberta-base

[2] Model link: https://huggingface.co/cardiffnlp/twitter-roberta-base-emotion

[3] Model link: https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment-latest

[4] Model link: https://huggingface.co/princeton-nlp/unsup-simcse-roberta-base

Model	**Averaged Pearson Correlation**	**Empathy Pearson Correlation**	**Distress Pearson Correlation**
Twitter-RoBERTa-emotion	0.3189	0.3389	0.2991
Twitter-RoBERTa-sentiment	0.294	0.3128	0.2753
Unsupervised SimCSE RoBERTa	**0.35285**	0.3311	**0.3746**
RoBERTa-base	0.29075	**0.3444**	0.2371
Ensembles			
Mean	**0.34619**	0.3585	**0.3339**
Linear Regression	0.3285	0.3349	0.3221
SVR	0.3221	**0.3837**	0.2605
XGBoost	0.2898	0.3502	0.2294

Table 2: The results of our methods along with their scores. We use Pearson's *r* as the scoring metric.

Rank	**Codalab ID**	**Score**
1	ltm11	0.4178
2	Gruschka	0.3837
3	earendil	0.3462
4	zex	0.3419
5	luxinxyz	0.3416

Table 3: The top 5 teams at the EMP track. The score reported is averaged Pearson's correlation values of empathy and distress. Our team participated under the username *earendil*.

more mindfully composed.

Amongst the ensembling approaches, calculating the average value of individual model predictions generates the best result. It also outperforms other techniques at predicting distress. Support Vector Regression has the best results for empathy prediction. Although some BERT-based models perform better than others, supervised learning algorithms like linear regression and XGBoost fail to utilize this information and perform poorly. The poor performance can also be attributed to relatively less training data.

Our team finished 3rd in the EMP track at the shared task. Our final submission has an average Pearson correlation score of 0.346. The top five participants along with their scores are reported in Table 3.

6 Conclusion

The approach of Empathy Detection and Emotion Classification is proposed as part of the WASSA 2023 Empathy, Emotion and Personality Shared Task. Various methods are explored for the task. We have implemented several BERT-based models and evaluated them. We observed that the unsupervised SimCSE model has the best performance among the models we evaluated. It can also be seen that averaging the results of individual models generates the best results among the ensembling methods. Our final submission, with a Pearson's *r* score of 0.346, is the third-best score in the EMP (Empathy Prediction) track at the shared task. Several improvements and future lines of work can be identified. Additional models pre-trained on relevant data can potentially boost methods. Furthermore, other ensembling techniques can also be explored and evaluated for better results.

Acknowledgment

We thank the Pune Institute of Computer Technology's Computational Linguistics Research Lab for providing the necessary help and guidance for the work. We are grateful for their support.

Limitations

Although our models perform efficiently on the provided dataset, they might not be feasible in real-world scenarios due to the amount of computation they require. The development of more efficient models will vastly improve the deployability of such systems.

References

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650, Online. Association for Computational Linguistics.

Valentin Barriere, Shabnam Tafreshi, Jo ao Sedoc, and Sawsan Alqahtani. 2022. Wassa 2022 shared task: Predicting empathy, emotion and personality in reaction to news stories. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 214–227.

Valentin Barriere, Shabnam Tafreshi, João Sedoc, and Salvatore Giorgi. 2023. Wassa 2023 shared task: Predicting empathy, emotion and personality in interactions and reaction to news stories. In *Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2022. Simcse: Simple contrastive learning of sentence embeddings.

James Gibson, Dogan Can, Bo Xiao, Zac E Imel, David C Atkins, Panayiotis Georgiou, and Shrikanth Narayanan. 2016. A deep learning approach to modeling empathy in addiction counseling. *Commitment*, 111(2016):21.

Bhanu Prakash Reddy Guda, Aparna Garimella, and Niyati Chhaya. 2021. Empathbert: A bert-based framework for demographic-aware empathy prediction. *arXiv preprint arXiv:2102.00272*.

Mahshid Hosseini and Cornelia Caragea. 2021. Distilling knowledge for empathy detection. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3713–3724.

Marina Litvak, Jahna Otterbacher, Chee Siang Ang, and David Atkins. 2016. Social and linguistic behavior and its correlation to trait empathy. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Damilola Omitaomu, Shabnam Tafreshi, Tingting Liu, Sven Buechel, Chris Callison-Burch, Johannes Eichstaedt, Lyle Ungar, and Jo ao Sedoc. 2022. Empathic conversations: A multi-level dataset of contextualized conversations. *arXiv preprint arXiv:2205.12698*.

Shirin Saleem, Rohit Prasad, Shiv Vitaladevuni, Maciej Pacula, Michael Crystal, Brian Marx, Denise Sloan, Jennifer Vasterling, and Theodore Speroff. 2012. Automatic detection of psychological distress indicators and severity assessment from online forum posts. In *Proceedings of COLING 2012*, pages 2375–2388.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Satu Viertiö, Olli Kiviruusu, Maarit Piirtola, Jaakko Kaprio, Tellervo Korhonen, Mauri Marttunen, and Jaana Suvisaari. 2020. Factors contributing to psychological distress in the working population, with a special reference to gender difference - a population-based study.

Team Bias Busters at WASSA 2023 Empathy, Emotion and Personality Shared Task: Emotion Detection with Generative Pretrained Transformers

Andrew Nedilko
agnedil@gmail.com

Yi Chu
yirosie@gmail.com

Abstract

This paper describes the approach that we used to take part in the multi-label multi-class emotion classification as Track 3 of the WASSA 2023 Empathy, Emotion and Personality Shared Task at ACL 2023. The overall goal of this track is to build models that can predict 8 classes (7 emotions + neutral) based on short English essays written in response to news article that talked about events perceived as harmful to people. We used OpenAI generative pretrained transformers with full-scale APIs for the emotion prediction task by fine-tuning a GPT-3 model and doing prompt engineering for zero-shot / few-shot learning with ChatGPT and GPT-4 models based on multiple experiments on the dev set. The most efficient method was fine-tuning a GPT-3 model which allowed us to beat our baseline character-based XGBoost Classifier and rank 2nd among all other participants by achieving a macro F1 score of 0.65 and a micro F1 score of 0.7 on the final blind test set.

1 Introduction and Related Works

Emotion prediction by a machine is a challenging task because emotions are inherently a human quality, and as everything human they are quite subjective - different people from different cultures may interpret emotions in very different ways. Even if it is the same culture, similar text in different contexts can be understood as different emotions or lack thereof. Due to this high variability, it may be not easy to get accurately annotated text for emotions because the annotators may disagree as to the precise emotions expressed in the same text.

Another aspect of emotions is that they can be interpreted using extra-linguistic information, such as the voice tone/pitch, intonation, the presence of a smile or other facial expressions, etc. But these features are absent when text is the only information available for emotion detection.

Despite all these difficulties, the modern AI systems such as customer-facing chatbots or automated phone systems can definitely benefit greatly from an improved ability to detect emotions, because this will mean better customer service. And as we are seeing the rise in the use of such AI systems (Plaza et al., 2022), the task of emotion detection becomes more and more important.

In this regard, Barriere et al. (2022) presents an overview of the most recent emotion studies and describes the results of the similar shared task for 2022. Tafreshi et al. (2021) also provides an overview of emotion studies and talks about the results of the similar shared task for 2021. Omitaomu et al. (2022) describes the process of creating the dataset of empathy conversations for the current shared task.

Alvarez-Gonzalez et al. (2021) utilizes two large emotion classification corpora, designs a benchmark and evaluates several machine learning algorithms including two novel BERT models.

Acheampong et al. (2021) talks about the importance of extracting contextual information for NLP including emotion recognition from text and discusses such transformer-based models as generative pre-trained transformers (GPT), XLM, and BERT in the light of the text-based emotion detection.

Yang et al. (2023) evaluates the use of the latest LLMs such as ChatGPT for emotional reasoning on multiple datasets across several tasks and analyzes the effects of various emotion-based prompting strategies in the context of mental health analysis.

2 Dataset and Task

The WASSA 2023 Empathy, Emotion and Personality Shared Task includes 5 tracks for empathy and emotion prediction in conversations, essays, emotion classification and personality / interpersonal reactivity prediction. We participated in **Track 3 Emotion Classification (EMO)** which involves

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 569–573
July 14, 2023 ©2023 Association for Computational Linguistics

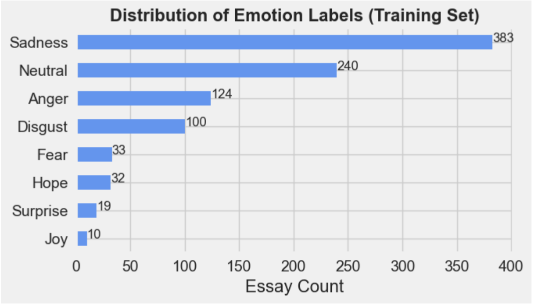


Figure 1: Distribution of Emotion Labels - Training Set

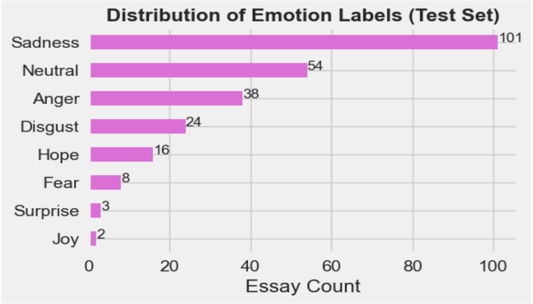


Figure 2: Distribution of Emotion Labels - Test Set

predicting emotions at the essay level.

The dataset for Track 3 Emotion Classification (EMO) consists of short essays written in response to news articles describing some events that were harmful to people (Omitaomu et al., 2022). Each essay has one or two emotion categories which makes this task the multi-label multi-class text classification. The emotion categories include Sadness, Anger, Disgust, Hope, Neutral, Surprise, and Fear. There are 792 essays in the training set, 208 essays in the development set (hereinafter, dev set), and 100 essays in the final blind test set. The essays contain a lot of spelling errors.

The essays required some light text cleaning. Some essays didn't have any punctuation at all. There are mostly no missing values in the dataset except for several cases in non-textual feature columns such as gender, education, race, age, income.

A blind test set where labels were not provided was used for testing the model that had the best performance on the dev set.

When it comes to the distribution of emotion categories, both training set and test set have a similar, very imbalanced distribution. The most overrepresented classes are sadness and neutral, and the least represented ones – joy and surprise.

3 System Description

3.1 Baseline Model

The initial metrics were set by the baseline model - XGBoost Classifier with character ngram counts as features (Countvectorizer). The ngram range was (1,7). Using word counts or tf-idf scores for words or characters proved to be less efficient. The initial macro F1 score was below 0.5, but with some improvements, such as a combination of downsampling / oversampling and data augmentation, we managed to achieve the final baseline macro F1 score of 0.56 and the micro F1 score of 0.62 (see Table 1 below). This baseline turned out to be quite hard to beat.

See subsection 4.2 below for a description of the data augmentation process. For oversampling, we decided not to oversample all classes up to the number of data points in the majority class, sadness, which is quite a big number – 383. Instead, we randomly downsampled sadness to 240 data points, as in the neutral class, and then randomly oversampled other classes (with replacement, if necessary) to 240 data points each. This proved to be more efficient than not oversampling at all or oversampling to 383.

3.2 GPT: Iterative Prompt Engineering vs. Fine-Tuning

Using transformer models and their ensembles (Kshirsagar et al., 2022) was proved to be efficient for sequence classification, but the macro F1 score for emotion detection was still below 0.55. We all have witnessed the recent rise of autoregressive models with the generative pretrained transformer (GPT) architecture and the fact that they demonstrate "human-level performance on various professional and academic benchmarks" (OpenAI, 2023). Therefore, we decided to evaluate whether the GPT series models can help solve the task of emotion classification in a more efficient way.

For this purpose, we used a suite of OpenAI models because they have a full-scale commercial API that allows multiple ways to interact with pretrained models. First of all, we utilized the ChatGPT and GPT-4 APIs with prompt engineering - generating dozens of different prompts in order to run full-scale experiments on the dev set to maximize the macro F1 score. We used the zero-shot and the few-shot approaches. The training set was used only to concatenate together examples for the few-shot learning.

As we were not able to beat our baseline model using these APIs, we tried fine-tuning an older GPT series model in an attempt to improve the metrics. Fine-tuning is currently not available for either ChatGPT or GPT-4. Only the original GPT-3 base models that do not have any instruction following training and are smaller than ChatGPT can be fine-tuned. We selected the largest one – DaVinci. Doing this allowed us to beat our own baseline model and to get the best results among all our models. We used the standard OpenAI API to fine-tune the model without changing the predefined hyperparameters.

Overall, we ranked 2nd and achieved the macro F1 score of 0.65 and the micro F1 score of 0.7 on the final blind test set.

4 Analysis of Results

4.1 ChatGPT vs. GPT-4

The idea behind using the zero-shot learning was based on the fact that the names of the 8 labels (7 emotions + neutral) are self-explanatory and can be well understood by such a pre-trained model as ChatGPT or GPT-4. However, the best macro F1 score achieved using this method for both models was only 0.46 which is lower than the baseline XGBoost Classifier (0.51-0.56).

Therefore, next we selected the few-shot method to enhance the zero-shot classification results. Since the context window size is limited (4096 for ChatGPT and 8192 for GPT-4), we had to select only a limited number of essay + label examples from the training set.

Most efficient prompt contained step-by-step instructions for ChatGPT describing the task, the categories, the actions to be taken, especially the fact that the second category must be added only if it is absolutely necessary.

Alternatively, we excluded the step-by-step instructions and used only the concatenated essay + label examples from the training set with a question about the category of the last unlabeled essay to be classified.

Both methods seemed to be equally efficient. Sometimes, the first method performed better because of the step-by-step instructions, sometimes the second method was better because one can squeeze in more training set examples since the instructions don't take up space.

The two methods used to select the existing example from the training set were: 1) selecting N random example from the training set; when doing this, each essay to be classified was getting different random examples so that eventually all the training examples were used with an equal frequency, 2) using N examples from the training set that would be the most similar to the essay to be classified. N was determined experimentally to stay within the context window size. To determine the similarity, we used the OpenAI embeddings (the text-embedding-ada-002 model). For this particular task, the random sampling outperformed the most similar approach.

Here are some of the interesting facts about comparing the performance of ChatGPT and GPT-4: zero-shot results for GPT-4 were less accurate than for ChatGPT. Reason: GPT-4 is too eager to output the second emotion category, even when it is not required and even when the model's temperature setting is 0. This led to a situation when almost all dev set data points had two categories predicted, even when the ground truth contained just one category.

We used several prompts trying to discourage GPT-4 from including the second emotion, such as: "Do not add the second category unless it is absolutely necessary" - and this still didn't help.

As for few-shot learning, the GPT-4 results were close to those of ChatGPT, with some slight advantage of ChatGPT. One other aspect to remember is that the GPT-4 API is a lot more expensive than ChatGPT - very quickly our experiments started costing us 3-digit amounts while the ChatGPT experiments cost approximately a few dozen dollars.

4.2 Data Augmentation

Some of the essays have two few labels in the training data (e.g. anger/sadness), and there are multiple cases when one of the two labels is neutral while it is hard to imagine that the same text can be both emotional and neutral at the same time. As an experiment we removed the neutral label in such cases. It was somewhat useful for the baseline classifier, but the final fine-tuned GPT model actually benefited from the presence of the second neutral label.

We attempted to use non-textual feature columns such as gender, education, race, age, income. Using these features alone we achieved a macro F1 score of 0.37 (micro F1 score = 0.52). However, the non-textual features did not provide any benefits when we combined them with the text features.

To add more examples to the minority classes, we conducted data augmentation for the smallest categories: hope, surprise, joy, fear. A total of 165 new examples were added using the following technique - GPT-4 was given some examples of the essays in a certain class and then the model was asked to generate 20-50 more examples in the same manner and style and using semantically similar vocabulary. This technique helped to train a better baseline model and eventually the final winning model.

We also tried to generate other types of augmented data. For example, GPT-4 was asked to come up with a good title and a meaningful summary for each essay, but this approach did not provide any significant uplift in the final results.

4.3 Model Comparison

The official competition metric for emotion prediction is the macro F1 score with the secondary metrics being micro Jaccard score, micro F1 score, micro precision, micro recall, macro precision, macro recall. Table 1 below lists only the macro and micro F1 scores for our models to save space. All the scores in Table 1 are for the dev set. The best performing model shown in the last line of Table 1 scored 0.6469 (macro F1) and 0.6996 (micro F1) on the final blind test set which allowed our solution to rank 2nd among all other participants.

It is worth noting that, as the zero-shot learning method was always outperformed by the few-shot learning, we observed two evident **limitations related to few-shot learning**:

- ChatGPT has a relatively small context window size (4k tokens) which doesn't allow it to fit in all examples from the training set.
- GPT-4 has a larger context window of 8k tokens, but is considerably more expensive (cost constraint) - several rounds of few-shot learning when we tried to show the model as many training set examples as possible lead to the costs in the 3-digit range for the GPT-4 API.

5 Conclusions

We have come to a conclusion that ChatGPT and GPT-4 seem unpredictable in their behavior to a certain degree. This volatility makes it harder to find a consistently working configuration for them - it is more difficult to control them. It is not surprising that the task of emotion prediction using the zero-shot and few-shot methods on this particularly difficult dataset turned out to be too hard even for such state-of-the-art models.

Classifier	**Macro F1**	**Micro F1**
Baseline XGBClassifier	0.5057	0.6053
Improved baseline XGBClassifier	0.5638	0.6162
Zero-shot ChatGPT	0.4620	0.5720
Few-shot ChatGPT (random examples)	0.4744	0.5992
Few-shot ChatGPT (most similar examples)	0.4237	0.5906
Zero-shot GPT-4	0.4285	0.5505
Few-shot GPT-4 (random examples)	0.4657	0.6300
Few-shot GPT-4 (most similar examples)	0.4325	0.5940
Fine-tuned DaVinci	0.5811	0.6877
Fine-tuned DaVinci w/augmented data	0.5916	0.6800

Table 1: Performance of Various Classifiers on Development Set

The largest OpenAI fine-tunable model DaVinci, which is older and smaller than ChatPGT and does not have any instruction following training, proved to be much more efficient for this task. This fine-tuned model outputs class probabilities which is very useful for the current multi-label multi-class classification task because we had to make a decision about when to add the second class. This decision was based on probability cutoffs.

In addition, the ability to fine-tune a model helped us solve both few-shot learning limitations mentioned in subsection 4.3 because the model being fine-tuned sees all the training set examples and at inference you pay only for the tokens in the one example to be classified. Also, as this experiment showed, fine-tuning is a very powerful text classification technique when it is used with GPT models.

References

Francisca Adoma Acheampong, Henry Nunoo-Mensah, and Wenyu Chen. 2021. Transformer models for text-based emotion detection: a review of bert-based approaches. *Artificial Intelligence Review*, 54:5789—-5829.

Nurudin Alvarez-Gonzalez, Andreas Kaltenbrunner, and Vicenc Gomez. 2021. Uncovering the limits of text-based emotion detection. arXiv:2109.01900.

Valentin Barriere, Shabnam Tafreshi, João Sedoc, and Sawsan Alqahtani. 2022. Wassa 2022 shared task: Predicting empathy, emotion and personality in reaction to news stories. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 214–227.

Valentin Barriere, Shabnam Tafreshi, João Sedoc, and Salvatore Giorgi. 2023. Wassa 2023 shared task: Predicting empathy, emotion and personality in interactions and reaction to news stories. In *Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*.

Yue Chen, Yingnan Ju, and Sandra Kubler. 2022. Iucl at wassa 2022 shared task: A text-only approach to empathy and emotion detection. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 228–232.

Soumitra Ghosh, Dhirendra Maurya, Asif Ekbal, and Pushpak Bhattacharyya. 2022. Team iitp-ainlpml at wassa 2022: Empathy detection, emotion classification and personality detection. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 255–260.

Atharva Kshirsagar, Shaily Desai, Aditi Sidnerlikar, Nikhil Khodake, and Manisha Marathe. 2022. Leveraging emotion-specific features to improve transformer performance for emotion classification. arXiv:2205.00283.

Himanshu Maheshwari and Vasudeva Varma. 2022. An ensemble approach to detect emotions at an essay level. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 276–279.

Damilola Omitaomu, Shabnam Tafreshi, Tingting Liu, Sven Buechel, Chris Callison-Burch, Johannes Eichstaedt, Lyle Ungar, and Joao Sedoc. 2022. Empathic conversations: A multi-level dataset of contextualized conversations. arXiv:2205.12698.

OpenAI. 2023. Gpt-4 technical report.

Miroslaw Plaza, Slawomir Trusz, Justyna Keczkowska, Ewa Boksa, Sebastian Sadowski, and Zbigniew Koruba. 2022. Machine learning algorithms for detection and classifications of emotions in contact center applications. *https://www.mdpi.com/journal/sensors*.

Shenbin Qian, Constantin Orasan, Diptesh Kanojia, Hadeel Saadany, and Felix do Carmo. 2022. Surrey-cts-nlp at wassa2022: An experiment of discourse and sentiment analysis for the prediction of empathy, distress and emotion. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 271–275.

Shabnam Tafreshi, Orphee De Clercq, Valentin Barriere, Sven Buechel, João Sedoc, and Alexandra Balahur. 2021. Wassa 2021 shared task: Predicting empathy and emotion in reaction to news stories. In *Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 92–104.

Kailai Yang, Shaoxiong Ji, Tianlin Zhang, Qianqian Xie, Ziyan Kuang, and Sophia Ananiadou. 2023. Towards interpretable mental health analysis with chatgpt.

HIT-SCIR at WASSA 2023: Empathy and Emotion Analysis at the Utterance-Level and the Essay-Level

Xin Lu*, **Zhuojun Li***, **Yanpeng Tong***, **Yanyan Zhao**†, **Bing Qin**
Research Center for Social Computing and Information Retrieval
Harbin Institute of Technology, China
`{xlu, zjli, yptong, yyzhao, qinb}@ir.hit.edu.cn`

Abstract

This paper introduces the participation of team HIT-SCIR to the WASSA 2023 Shared Task on Empathy Detection and Emotion Classification and Personality Detection in Interactions. We focus on three tracks: Track 1 (Empathy and Emotion Prediction in Conversations, CONV), Track 2 (Empathy Prediction, EMP) and Track 3 (Emotion Classification, EMO), and designed three different models to address them separately. For Track 1, we designed a direct fine-tuning DeBERTa model for three regression tasks at the utterance-level. For Track 2, we designed a multi-task learning RoBERTa model for two regression tasks at the essay-level. For Track 3, we designed a RoBERTa model with data augmentation for the classification task at the essay-level. Finally, our team ranked 1st in the Track 1 (CONV), 5th in the Track 2 (EMP) and 3rd in the Track 3 (EMO) in the evaluation phase.

1 Introduction

In the field of human-computer interaction systems, a discernible trend is the increased focus on the emotion and empathy status of users and the facilitation of emotional exchanges with them. This approach significantly contributes to enhancing service quality and boosting user satisfaction.

However, analyzing the emotion and empathy status of users is still a challenging problem, which requires researchers to conduct thorough exploration and in-depth study. The WASSA 2023 Shared Task 1 (Barriere et al., 2023) provides a unified evaluation benchmark, on the basis of which we have conducted corresponding work.

We have participated in three of five tracks, which are:

Track 1: Empathy and Emotion Prediction in Conversations (CONV), which consists in predicting the perceived empathy, emotion polarity and emotion intensity at the utterance-level in a dialog.

Track 2: Empathy Prediction (EMP), which consists in predicting both the empathy concern and the personal distress at the essay-level.

Track 3: Emotion Classification (EMO), which consists in predicting the emotion at the essay-level.

We conducted analyses and experiments on these three tracks concurrently. In Section 2, we present the methodologies designed for different tasks, the dataset features used in our design and our ensembling method. In Section 3, we introduce the experimental results of our proposed methods, along with corresponding result analyses. In Section 4, we provide our conclusions and summarize our methodologies. The implementation details can be found in Appendix A.

2 System Description

2.1 Track 1: Empathy and Emotion Prediction in Conversations (CONV)

The training set is initially analyzed, revealing an average dialogue length of 23 turns, with each utterance averaging 18 tokens. More details of how this dataset was designed can be found in Omitaomu et al. (2022). To encode the context information of each utterance effectively, we employ a concatenation approach to encode the dialogue information without significant loss. Given the strong contextual relevance of emotion polarity, emotion intensity, and empathy in dialogues, each turn is assigned a context window, and through comprehensive experimentation, we determine the optimal window size for each metric. Our approach involves direct fine-tuning of the DeBERTa (He et al., 2020) model for regression tasks (more details can be found on Appendix A), resulting in a

---

* Equal Contribution.
† Email Corresponding.

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 574–580
July 14, 2023 ©2023 Association for Computational Linguistics

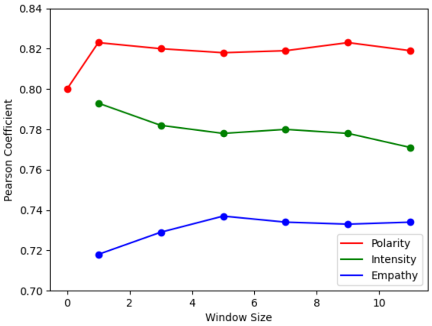


Figure 1: Pearson Correlation Coefficient of different window sizes using the deberta-xl model in the official development dataset.

collection of models that exhibited favorable performance. Then employing a model selection method, unstable models are filtered out, and the remaining models in the collection are ensembled for the final results.

2.1.1 Model Architecture

The model architecture is illustrated in Figure 3a. For a given utterance, denoted as u_k, the corresponding input is constructed as follows.

$$< s > u_{k-w}...u_{k-1} < s > u_k < /s > u_{k+1}...u_{k+w}$$

Here, $u_k = w_1, w_2, ..., w_n$, where n represents the number of tokens in the k-th utterance. The input is fed into the encoder, and the output corresponding to the first <s> token is taken as the contextual representation for the k-th utterance. It is then passed through an MLP to obtain the corresponding output regression value. We apply the same data processing to the validation set. The method of processing the input text without altering the model architecture is quite simple and effective . We adopt DeBERTa as the contextual encoders.

2.1.2 Contextual Window

We conduct extensive experiments on different models and different context window sizes, and the results are shown in Figure 1. It can be observed that for emotion intensity, the trend indicates that the metric decreases as the window size increases. This may be because the expression of emotion intensity is often highly correlated with the expression of the current sentence and does not depend on context too far away. Therefore, we abandon windows larger than 3 for this task. For emotion polarity, we find that the performance is relatively similar for window sizes larger than 0, so we discard the windows with relatively worse results, such as 5 and 11. For empathy, we discard models with window sizes less than 5. All of the results above are achieved by deberta-xl on the official development set. We also train deberta-xxl and select some models based on similar criteria to form a model set χ, which participate in the final model ensemble. We also find that for emotion polarity prediction tasks, models with window sizes greater than 0 are significantly better than those with a window size of 0 (single-sentence prediction). However, when the window size is too large (9 or 11), the metric decrease (as experiment results on xxl proved). This is consistent with the intuition that emotion polarity depends on context but not on irrelevant context.

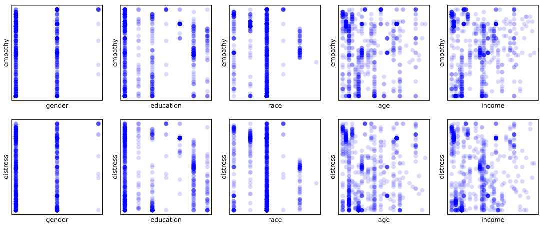


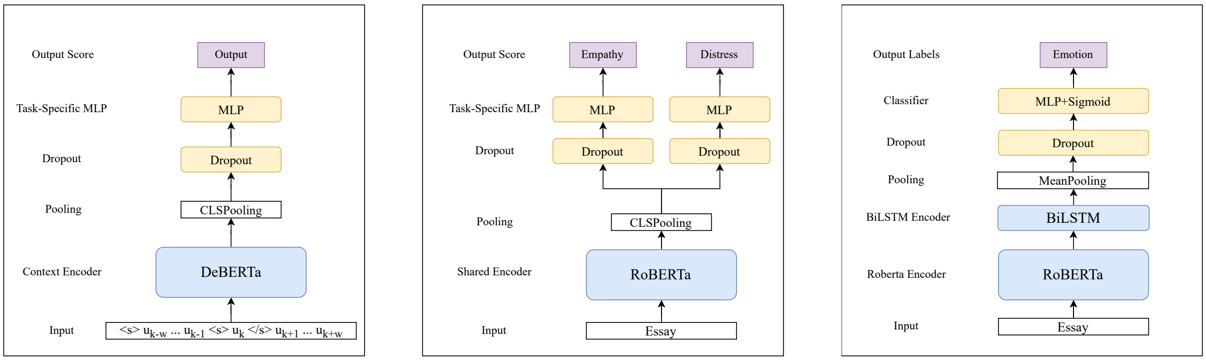
Figure 2: Scatter plot of the bivariate distribution of empathy/distress ratings and demographic features.

2.2 Track 2: Empathy Prediction (EMP)

Initially, we perform a correlation analysis on the train set, examining the relationship between empathy/distress ratings and demographic features. Our findings indicate no significant correlation between the demographic features and empathy/distress ratings. Additionally, building upon Batson's Empathy Theory (Batson et al., 1987) and considering the high Pearson correlation score observed between empathy and distress in a previous study (Buechel et al., 2018), we proceed to investigate the correlation between empathy and distress within the train set. This subsequent analysis reveals a strong correlation between these two variables. Consequently, we employ a multi-task learning approach to effectively model both the empathy and distress subtask.

2.2.1 Data Analysis

In Figure 2, we display the bivariate distribution of empathy/distress ratings and demographic features, indicating a lack of significant correlation between them. Additionally, based on previous researches (Lahnala et al., 2022; Chen et al., 2022; Ghosh et al., 2022), most models have achieved good re-



(a) CONV track model architecture. (b) EMP track model architecture. (c) EMO track model architecture.

Figure 3: Track 1 (CONV), Track 2 (EMP) and Track 3 (EMO) model architectures.

sults without incorporating these features. To avoid introducing more noise into the model, we choose to follow approach by Chen et al. (2022) and construct a model by fine-tuning of the RoBERTa (Liu et al., 2019) model. Inspired by Buechel et al. (2018), we compute the Pearson correlation between empathy and distress in the training set, which results in a high score of 0.63. This finding suggests that a multi-task learning approach, which simultaneously models both empathy and distress, is a suitable choice.

2.2.2 Multi-task Learning Model

In Figure 3b, we select RoBERTa for encoding the essays in the EMP task. To represent the entire sentence, we use the CLS token and applied a single-layer MLP with dropout to predict the empathy and distress outputs. For the multi-task model, we share the RoBERTa encoding layer and equally weight the losses of both subtasks during fine-tuning.

2.3 Track 3: Emotion Classification (EMO)

An initial analysis of the distribution of data labels in the training set reveals a small dataset size and an uneven distribution. To address this issue, we employ data augmentation techniques, attempting various methods, including EDA (Wei and Zou, 2019), GoEmotions (Demszky et al., 2020), and ChatGPT rephrasing. Our experiments ultimately show that ChatGPT rephrasing produce the best results. Additionally, after testing different model architectures, we select a structure that is both effective and robust.

2.3.1 Data Analysis

We analyze the distribution of emotion labels in the training set. The number of instances for the "Fear", "Hope", "Joy", and "Surprise" is significantly lower compared to the other labels. On the other hand, the "Sadness" and "Neutral" labels have a relatively larger number of instances. Additionally, we have computed the distribution of single-label instances for each category, and it is found that the proportion of single-label instances for "Fear", "Hope", "Joy", and "Surprise" is consistently lower than 50%. This indicates that the classification of these labels is prone to be influenced by other labels, posing a significant challenge for modeling them. More detailed statistics are shown in Appendix B.

2.3.2 Data Augmentation with ChatGPT

Developed by OpenAI and released in November 2022, ChatGPT is an artificial intelligence chatbot that achieves strong instruction-following abilities through fine-tuning and reinforcement learning on large language models such as GPT 3.5 and GPT4. Leveraging ChatGPT's powerful language modeling capabilities, our objective is to perform data augmentation on imbalanced samples in order to mitigate the potential biases. We use it to rephrase the original essay for data augmentation, and more details are shown in Appendix C.

To tackle any potential data imbalance and improve our model's performance, we generate over 200 additional instances for each of the categories, except for “Sadness” and “Neutral” due to their relative abundance of data. To ensure that the expanded data did not introduce excessive noise, we apply a sorting process based on ROUGE-L scores and prioritized sentences with higher scores.

2.3.3 Emotion Classification Model

In Figure 3c, we use the RoBERTa model as the essay encoder. The encoded vectors are then processed through a BiLSTM layer to capture long-

Team	Avg	Polarity	Intensity	Empathy
HIT-SCIR	0.758	0.852	0.714	0.708
YNU-HPCC	0.730	0.824	0.693	0.674
Team Hawk	0.725	0.809	0.701	0.665

Table 1: Test dataset results (Pearson correlations) for Track 1 (CONV) in the evaluation phase.

distance word dependencies within the essay. Afterwards, the BiLSTM outputs are averaged, followed by a dropout operation. Finally, an 8-dimensional vector is obtained through a single-layer MLP, using the sigmoid function as the activation function for multi-label classification.

2.4 Ensembling Method

On the official essay-level development set, speaker information such as gender, education level, race, and age are available. Using these attributes, we divide the speakers into 21 groups and then partition the samples in development datasets according to the speaker groups. This results in 21 datasets with different distributions. We consider that a model with strong generalization ability should not have too much variation in performance across these 21 different development subsets. Therefore, we further filter the initial model set χ based on variance and obtained the final model set $\chi\prime$. Using these models, we can further achieve model ensemble. Especially, for regression models, we directly average the regression values output by each model in the set to obtain the ensembled regression value. The division details can be found in Appendix D.

3 Results and Discussions

3.1 Results for Track 1 (CONV)

The results presented in Table 1 indicate that our final ensembled model achieved the top rank on the official test set. Specifically, our model outperforms the second-ranked model by almost 3 points in predicting emotional polarity and empathy, and by 2 points in predicting intensity. This remarkable performance demonstrates the superior generalization ability of our final ensembled model on the test set, which can be attributed to our effective model ensemble strategy and our context window selection.

3.2 Results for Track 2 (EMP)

Table 2 presents the results of our systems on the test set of the EMP task. We also provide our results on the dev set, where it outperformed all known results. However, we observe a significant drop in performance on the test set for both our systems and note that other teams experience similar performance drops. This suggests the data distribution between the dev and test sets may differ significantly, leading to overfitting to the dev set and poor generalization performance on the test set.

Team	Average	Empathy	Distress
NCUEE-NLP	0.4178	0.4150	0.4206
CAISA	0.3838	0.3478	0.4197
earendil	0.3462	0.3585	0.3339
zex	0.3420	0.2933	0.3906
HIT-SCIR	0.3416	0.3287	0.3545
HIT-SCIR (Dev set)	0.6571	0.6662	0.6480

Table 2: Test dataset results (Pearson correlations) for Track 2 (EMP) in the evaluation phase.

Team	Macro F1	Micro F1
adityapatkar	0.701	0.750
Bias Busters	0.647	0.700
HIT-SCIR	0.644	0.720

Table 3: Test dataset results (Macro F1 & Micro F1) for Track 3 (EMO) in the evaluation phase.

3.3 Results for Track 3 (EMO)

Table 3 presents the test results of Top-3 systems in this Task, and our system ranks 3rd. Upon analyzing the error logs in Codalab, we find that none of the instances in the test set were labeled as "Hope", "Joy", or "Surprise", which are precisely the three least represented labels in the training set. We hypothesize that our model introduce a trade-off between the underrepresented and overrepresented categories, which may have led to the slight decrease in performance on the test set. More ablation studies can be found in Appendix E.

4 Conclusion

Our team HIT-SCIR participated in the WASSA 2023 Shared Task on Empathy Detection and Emotion Classification and Personality Detection in Interactions. We focused on empathy and emotion analysis and participated in three of five tracks. We analyzed the features of each task and designed different methodologies for them. Finally, our team ranked 1st in the Track 1 (CONV), 5th in the Track 2 (EMP) and 3rd in the Track 3 (EMO) in the evaluation phase.

Acknowledgements

This work was supported by the National Key RD Program of China via grant 2021YFF0901602 and the National Natural Science Foundation of China (NSFC) via grant 62176078.

References

Valentin Barriere, Shabnam Tafreshi, João Sedoc, and Salvatore Giorgi. 2023. Wassa 2023 shared task: Predicting empathy, emotion and personality in interactions and reaction to news stories. In *Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*.

C Daniel Batson, Jim Fultz, and Patricia A Schoenrade. 1987. Distress and empathy: Two qualitatively distinct vicarious emotions with different motivational consequences. *Journal of personality*, 55(1):19–39.

Sven Buechel, Anneke Buffone, Barry Slaff, Lyle Ungar, and João Sedoc. 2018. Modeling empathy and distress in reaction to news stories. In *EMNLP 2018*.

Yue Chen, Yingnan Ju, and Sandra Kübler. 2022. IUCL at WASSA 2022 shared task: A text-only approach to empathy and emotion detection. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 228–232, Dublin, Ireland. Association for Computational Linguistics.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A dataset of fine-grained emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4040–4054, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Soumitra Ghosh, Dhirendra Maurya, Asif Ekbal, and Pushpak Bhattacharyya. 2022. Team IITP-AINLPML at WASSA 2022: Empathy detection, emotion classification and personality detection. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 255–260, Dublin, Ireland. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Allison Lahnala, Charles Welch, and Lucie Flek. 2022. CAISA at WASSA 2022: Adapter-tuning for empathy prediction. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 280–285, Dublin, Ireland. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Damilola Omitaomu, Shabnam Tafreshi, Tingting Liu, Sven Buechel, Chris Callison-Burch, Johannes Eichstaedt, Lyle Ungar, and João Sedoc. 2022. Empathic conversations: A multi-level dataset of contextualized conversations. *arXiv preprint arXiv:2205.12698*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

A Implementation Details

We train the model using the Pytorch (Paszke et al., 2019) on the NVIDIA A100 GPU and use the hugging-face (Wolf et al., 2020) framework. The optimizer used for model training is AdamW (Loshchilov and Hutter, 2017) optimizer which is a fixed version of Adam (Kingma and Ba, 2014) with weight decay, and set β_1 to 0.9, β_2 to 0.99 for the optimizer. All experiments select the best parameters in the valid set and then use the ensembling method in 2.4 . Below is the details to the custom parameter settings for different tracks.

For CONV track, we use both the DeBERTa-v2-xl and DeBERTa-v2-xxl as our pre-trained models. For each metric of this track (polarity, intensity and empathy), we fine-tune on both models. Specifically, during the process of choosing the best window size, for each window size, we fine-tune on both models. The DeBERTa model comes with 24(48) layers and a hidden size of 1536. The total parameters are 900M(1.5B), and it is trained with 160GB raw data. DeBERTa improves the BERT and RoBERTa models using two novel techniques, the disentangled attention mechanism and the enhanced mask decoder. We use the learning rate {3e-6, 4e-6}, total training batch size 32, training epoch 6 for DeBERTa-xl and DeBERTa-xxl separately. We conduct distributed training on 4 NVIDIA A100-80GB GPUs and use fp16 training. We set the maximum length of 1024, and delete the excess.

For EMP track, we use the roberta-base as our pre-trained model, and fine-tune the model. RoBERTa (Liu et al., 2019) extends BERT (Devlin et al., 2019) by incorporating techniques like dynamic masking and removing the next sentence prediction pre-training objective. We conduct grid search with the learning rate varying in {1e-5, 2e-5, 3e-5}, batch size varying in {8, 16, 32}, and dropout rate varying in {0, 0.3}. We set the maximum length of 512, and delete the excess.

For EMO track, we use the roberta-large as our pre-trained model, and fine-tune the model. We conduct grid search with the same parameter search range as that for the EMP track. Additionally, the hidden dimension of the BiLSTM layer is 256.

B Data Statistics for Track 3 (EMO)

Table 4 presents the distribution of emotion labels in the training set.

Emotion	All Instances	Single-Label Instances
Anger	124	67
Disgust	100	44
Fear	33	10
Hope	32	10
Joy	10	5
Neutral	240	202
Sadness	383	297
Surprise	19	9

Table 4: Data distribution over emotion classes in the origin train dataset. "Single label instances" refers to the number of data instances that contain only one label.

C Our Prompt for ChatGPT

We use the following text as a prompt to provide to ChatGPT in order to rephrase the original essay for data augmentation.

> Prompt: You are a helpful assistant that rephrase text and make sentence smooth. Besides, you should keep the emotion in the text unchanged. Please rephrase the following text, it's written by participants after reading news articles where there is harm to a person, group or other. Pay attention to retain emotion of `{emotions}` in the source text, keep the word count in 300-800 characters. Text: `{content}`

The `{content}` field pertains to the essay in the training set that requires rephrasing, and `{emotions}` represents the emotion labels associated with that essay.

D Division Details

The division rules are shown in Table 5.

Attribution	Set1	Set2	Set3	Set4
age	<30	other	-	-
education	<5	other	-	-
income	<35000	other	-	-
race	1	2	3	5
gender	1	2	-	-

Table 5: Rules used to divide speakers into 21 groups.

E Ablation Study for Track 3 (EMO)

Table 6 shows the results of the ablation study on the dev set of the EMO Task. We use several different settings to demonstrate the effectiveness of our proposed methods. The ensembling strategy significantly improve the performance of our system, and adding BiLSTM and data augmentation methods also contributed to the improvement of the model's performance. Moreover, our final results exceed all known results on the dev set, but show a slight decrease on the test set.

Methods	**Macro F1**
RoBEERTa-large finetune	0.5798
+ BiLSTM	0.6117
+ BiLSTM + data augmentation	0.6178
+ BiLSTM + data augmentation + ensemble	0.6630

Table 6: Ablation study on dev set of Track 3 (EMO).

VISU at WASSA 2023 Shared Task: Detecting Emotions in Reaction to News Stories Using Transformers and Stacked Embeddings

Vivek Kumar
University of Cagliari, Italy
`vivek.kumar@unica.it`

Prayag Tiwari
Halmstad University, Sweden
`prayag.tiwari@hh.se`

Sushmita Singh
Liverpool John Moores University, UK
`sushmitafordata@gmail.com`

Abstract

Our system, VISU, participated in the WASSA 2023 Shared Task (3) of Emotion Classification from essays written in reaction to news articles. Emotion detection from complex dialogues is challenging and often requires context/domain understanding. Therefore in this research, we have focused on developing deep learning (DL) models using the combination of word embedding representations with tailored prepossessing strategies to capture the nuances of emotions expressed. Our experiments used static and contextual embeddings (individual and stacked) with Bidirectional Long short-term memory (BiLSTM) and Transformer based models. We occupied rank tenth in the emotion detection task by scoring a Macro F1-Score of 0.2717, validating the efficacy of our implemented approaches for small and imbalanced datasets with mixed categories of target emotions.

1 Introduction

Digitalization and ease of access to internet-based intelligent and interactive technologies have led to an unprecedented amount of textual data generation from social media, customer reviews, and online forums. Therefore, the need to accurately understand and extract emotions and sentiments from text has become imperative for two reasons; first, due to their various crucial applications such as sentiment analysis (Gupta et al., 2023), chatbots, mental health assessment(Wu et al., 2020), social media monitoring, market research, brand management, and customer feedback analysis and second to reduce the human efforts, time and resource requirements. The Shared Task on *Empathy Detection, Emotion Classification and Personality Detection in Interactions* of WASSA 2023[1] aims to develop models to predict various targets, including emotion, empathy, personality, and interpersonal-index, from textual data (Barriere et al., 2023). The shared task consists of five tracks, of which we participated in *Track 3: Emotion Classification* (EMO), which targets emotion classification at the essay level. This work presents two systems to capture the subtle notion of emotions expressed through texts: a) BiLSTM-based (Graves and Schmidhuber, 2005) DL model using static, contextual, and the combination of static and contextual (stacked) embeddings and b) Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019). Stacked embeddings (Bhandari et al., 2022) are fast-to-train, powerful but underutilized representations; therefore, to reckon their efficacy compared to the transformer model, we have used them in this work. Our proposed systems have performed competitively and got the tenth rank[2] in the evaluation phase of the *Track 3: EMO* task.

The remainder of the paper is structured as follows: Section 2 presents the notable research works on emotion detection. Section 3 presents the problem statement, dataset description, and the preprocessing strategies applied. In section 4, we present our different classification systems and the experimental setup. Section 5 presents the evaluation results of our proposed systems and comparison with other participating teams of the shared task. Finally, section 6 provides the conclusion and discusses the future research directions.

2 Literature Survey

The significance of accurate emotion detection and sentiment analysis extends beyond understanding textual data. Recent research has brought the machines one step closer to mimicking humans' innate ability to understand emotional cues from and text and different modalities. Works such as (Acheampong et al., 2020; Chatterjee et al., 2019) explore the emotion detection form texts; (Zhong

---

[1] `https://2023.aclweb.org/program/workshops/`

[2] The rank is solely based on the submissions done before the deadline of the shared task

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 581–586
July 14, 2023 ©2023 Association for Computational Linguistics

et al., 2019; Acheampong et al., 2021; Adoma et al., 2020) explores the variants of transformer models useful for emotion detection from texts. Some notable works such as (Wu et al., 2023, 2022; Bostan et al., 2020; Bostan and Klinger, 2018; Buechel and Hahn, 2017; Sosea and Caragea, 2020) have created the novel datasets from textual and conversational settings to address the scarce data challenges in complex domains for emotion detection.

3 Problem Statement, Dataset Description and Data Preprocessing

In this section, we have mentioned the problem statement tackled, the dataset description, and the data-prepossessing techniques implemented for our experiments.

3.1 Problem Statement

In this work, we tackled a multiclass classification problem to predict emotions from essay-level texts. The target labels consist of thirty-one categories of emotions, including individual and mixed sets of emotion categories, as follows: Hope/Sadness, Anger, Sadness, Neutral, Disgust/Sadness, Anger/Disgust, Fear/Sadness, Joy, Hope, Joy/Neutral, Disgust, Neutral/Sadness, Neutral/Surprise, Anger/Neutral, Hope/Neutral, Surprise, Anger/Sadness, Fear, Anger/Joy, Disgust/Fear, Fear/Neutral, Fear/Hope, Joy/Sadness, Anger/Disgust/Sadness, Anger/Surprise, Disgust/Neutral, Anger/Fear, Sadness/Surprise, Disgust/Surprise, Anger/Hope, and Disgust/Hope.

3.2 Dataset Description

The experimental dataset contains long essays of length between 300 and 800 (Omitaomu et al., 2022). The dataset includes news articles and person-level demographic information (empathy, distress, age, race, income, gender, education level, emotion labels, etc.). The dataset was made available as training, development (dev), and test sets where the target labels were shared only for the training and development sets for the evaluation phase. The overall distribution of the dataset is shown in Table 1 and the distribution of each emotion class of the train and dev sets is shown in Table 1.

Dataset Split Distribution			
Train	*Dev*	*Test*	*Total*
792	208	100	1000

Table 1: Train, dev, and test set distribution.

3.3 Dataset Preprocessing

As evident from Figure 1, the dataset is small and imbalanced and several emotion categories have only one data point. Also, the mixed categories of emotions in the target class made the task more challenging. Therefore, to overcome these constraints, we have applied a tailored preprocessing strategy along with standard NLP techniques to prepare the input dataset (Dessì et al., 2020; Kumar et al., 2021; Uysal and Gunal, 2014). The preprocessing steps are as follows. The input texts are converted to lowercase to make the dataset uniform in terms of representation (e.g., Emotion and emotion are represented by a common token, emotion). Punctuation, stopwords, newlines, whitespaces, and extra spaces are removed from the text. We have removed the special characters, symbols, and elements which are not part of the standard English language. We have expanded the contractions such as *didn't –> did not*. We performed stemming and lemmatization alternatively for experiments but observed a slight decline in the model's performance. Therefore, we have not considered them for preprocessing the input dataset for the final submission of *Track 3: EMO* shared task.

4 Methodology

This section describes our different systems (classification models) based on the BiLSTM and transformer model implemented for the emotion classification task.

4.1 BiLSTM Based DL Model

Our first system is a DL model using two BiLSTM layers. More precisely, this model's architecture consists of an embedding layer, followed by two BiLSTM layers, a dense layer, and an output layer at the end for the multi-class classification. The embedding layer is initialized by input_dim (size of the vocabulary); output_dim: (word vector length), embedding matrix, and sequences length. For ease of understanding, we have summed up the parameters and combination of embeddings used for our experiments in Table 2.

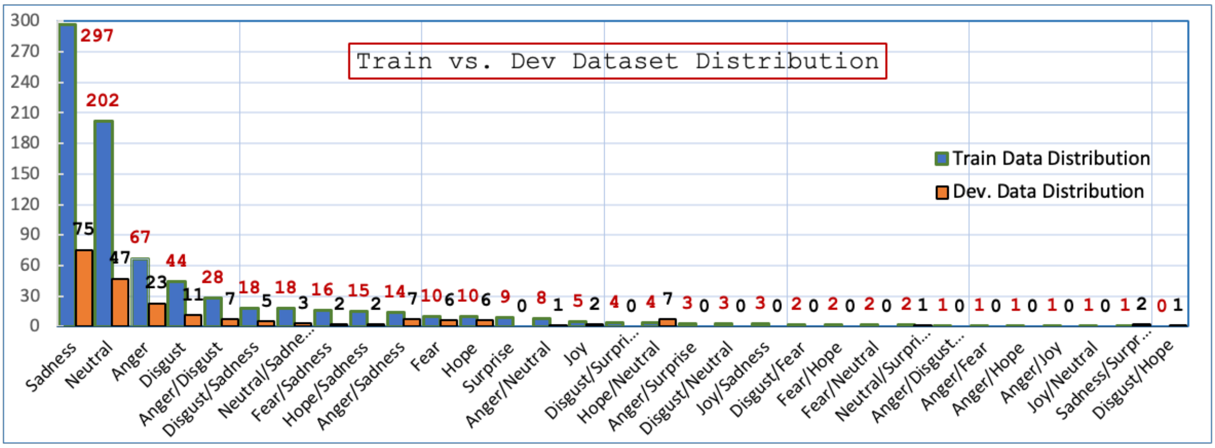


Figure 1: Plot showing the skewed distribution of training and development dataset.

4.2 BERT

The second system is a transformer-based model *BERT*, created using *Keras*[3] and Tensorflow[4]. Our *BERT* model comprises two input layers, a *BERT* model layer, and two dense layers of 768 embedding dimensions with the Adam optimizer. The parameters used for fine-tuning the model are listed in Table 2.

4.3 Features Representation

We have used pre-trained static and contextual word embeddings for our experiments to generate the feature vectors discussed below.

GloVE (Global Vectors for Word Representation): *GloVE*[5] is an unsupervised learning algorithm that generates word embeddings as dense vector representations of words in a high-dimensional space. It leverages co-occurrence statistics from a large text corpus to capture semantic relationships between words. *GloVe* embeddings are trained by factorizing a matrix representing the word co-occurrence statistics (Pennington et al., 2014).

fastText: *fastText*[6] was developed by Facebook's AI Research (FAIR) team (Bojanowski et al., 2017; Joulin et al., 2016). *fastText* extends the traditional word embeddings by representing each word as a bag of character n-grams, where n can range from 1 to a maximum specified length. This approach allows *fastText* to capture morphological information and handle out-of-vocabulary words effectively.

BERT: *BERT*[7] embeddings are a type of word representation that captures contextual information in the text. Unlike traditional word embeddings like *Word2Vec* or *GloVe*, *BERT* embeddings take into account the surrounding words when representing a word. This means that the meaning of a word can vary depending on its context.

FLAIR (FastText and Language-Independent Representations): *FLAIR*[8] embedding is a state-of-the-art word representation model that captures contextual information and word semantics by combining the strengths of two powerful techniques: *FastText* and contextual string embeddings. By combining these techniques, *FLAIR* embedding provides a robust and language-independent representation of words. It considers both the local context of a word and its global context within a sentence or document (Akbik et al., 2019).

5 Results and Analysis

Table 3 presents the results of our two systems. The BERT base system has significantly outperformed the BiLSTM-based system using the combination of *GloVe*, *fastText* and *BERT* embeddings. Therefore, we have submitted the BERT base result for the shared task evaluation phase. The evaluation of *Track 3: EMO* shared task is based on the macro F1-score and Micro Jaccard-score, Micro F1-score, Micro Precision, Micro Recall, Macro Precision and Macro Recall are supporting metrics. Our BERT base system has achieved a Macro F1-score of 2.717 and stood tenth[9] among participants. Table 4 presents the official results of all the qualifying teams.

---

[3] https://keras.io/

[4] https://t fhub.dev/google/collections/bert

[5] https://nlp.stanford.edu/projects/glove/

[6] https://fasttext.cc/

[7] https://pypi.org/project/bert-embedding/

[8] https://pypi.org/project/flair/

[9] These rankings were provided by the shared task organizing team on 11/05/2023.

Models	*Embedding Dimension*	*Sequence Length*	*Batch Size*	*Epoch*	*Learning Rate*
BiLSTM + GloVe	100	74	32	3	0.001
BiLSTM + fastText	300	74	32	4	0.001
BiLSTM + (GloVe & fastText)	400	128	32	3	0.001
BiLSTM + (GloVe & BERT)	868	128	32	3	0.001
BiLSTM + (fastText & BERT)	1068	152	32	7	0.001
BiLSTM + (GloVe, fastText & BERT)	1168	152	32	5	0.001
BERT	768	152	32	5	2e-5

Table 2: Experimental settings of proposed systems.

DL Model	**Static Embedding**		**Stacked Embedding**				**Contextual**
Metrics	*GloVe*	*fastText*	*GloVe + fastText*	*GloVe + BERT*	*fastText + BERT*	*GloVe+ fastText +BERT*	*BERT*
Micro F1-Score	0.213	0.213	0.213	0.213	0.204	0.213	**0.593**
Macro F1-Score	0.075	0.075	0.075	0.075	0.073	0.075	**0.284**
Micro Jaccard	0.119	0.119	0.119	0.119	0.113	0.119	**0.421**
Micro Precision	0.230	0.230	0.230	0.230	0.220	0.230	**0.640**
Macro Precision	0.046	0.046	0.046	0.046	0.045	0.046	**0.282**
Micro Recall	0.198	0.198	0.198	0.198	0.19	0.198	**0.552**
Macro Recall	0.192	0.192	0.192	0.192	0.183	0.192	**0.318**

Table 3: The results of our implemented models for static and contextual embeddings.

Rank	**Team ID**	**Macro F1 Score**	**Micro Recall**	**Micro Precision**	**Micro F1 Score**	**Macro Recall**	**Macro Precision**	**Micro Jaccard**
1	adityapatkar	0.7012	0.7241	0.7778	0.750	0.6773	0.8105	0.600
2	anedilko	0.6469	0.7931	0.6259	0.6996	0.7305	0.6305	0.538
3	luxinxyz	0.644	0.6983	0.7431	0.72	0.6314	0.7207	0.5625
4	zex	0.6426	0.7069	0.7321	0.7193	0.637	0.6992	0.5616
5	lazyboy.blk	0.6125	0.6638	0.77	0.713	0.6005	0.7764	0.554
6	gauravk	0.5649	0.7069	0.6949	0.7009	0.5605	0.5955	0.5395
7	amsqr	0.533	0.6293	0.7228	0.6728	0.4793	0.7521	0.5069
8	surajtc	0.522	0.7586	0.5269	0.6219	0.6679	0.4626	0.4513
9	alili_wyk	0.5142	0.6724	0.7358	0.7027	0.5022	0.575	0.5417
10	**kunwarv4**	0.2717	0.5517	0.64	0.5926	0.3012	0.2571	0.4211
11	Cordyceps	0.202	0.4138	0.3664	0.3887	0.2356	0.1905	0.2412
12	Sidpan	0.1497	0.4138	0.4848	0.4465	0.2111	0.2948	0.2874
13	mimmu3302	0.126	0.3966	0.46	0.4259	0.2	0.092	0.2706

Table 4: The official results of the evaluation phase of *Track 3: EMO* task. Our system VISU (Team ID kunwarv4 attained the tenth rank.)

6 Conclusion

Our system, VISU, participated in the shared task *Track 3: EMO* of emotion classification tasks of the WASSA 2023, and our BERT base system scored tenth rank. Our experiments conclude that although *FLAIR* are powerful word representations built to capture *out-of-vocabulary* words, they are not as effective as contextual embeddings when used for small and imbalanced datasets. Our future research aims to address the data imbalance and scarce data challenges (Kumar. et al., 2023) by incorporating novel augmentation techniques of domain adaptation(Kumar et al., 2022) to interpret better the emotions expressed in text.

References

Francisca Adoma Acheampong, Henry Nunoo-Mensah, and Wenyu Chen. 2021. Transformer models for text-based emotion detection: a review of bert-based approaches. *Artificial Intelligence Review*, pages 1–41.

Francisca Adoma Acheampong, Chen Wenyu, and Henry Nunoo-Mensah. 2020. Text-based emotion detection: Advances, challenges, and opportunities. *Engineering Reports*, 2(7):e12189.

Acheampong Francisca Adoma, Nunoo-Mensah Henry, and Wenyu Chen. 2020. Comparative analyses of bert, roberta, distilbert, and xlnet for text-based emotion recognition. In *2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, pages 117–121. IEEE.

Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. Flair: An easy-to-use framework for state-of-the-art nlp. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics (demonstrations)*, pages 54–59.

Valentin Barriere, Shabnam Tafreshi, João Sedoc, and Salvatore Giorgi. 2023. Wassa 2023 shared task: Predicting empathy, emotion and personality in interactions and reaction to news stories. In *Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*.

Aakash Bhandari, Vivek Kumar, Pham Thi Thien Huong, and Dang NH Thanh. 2022. Sentiment analysis of covid-19 tweets: Leveraging stacked word embedding representation for identifying distinct classes within a sentiment. In *Artificial Intelligence in Data and Big Data Processing: Proceedings of ICABDE 2021*, pages 341–352. Springer.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146.

Laura Ana Maria Bostan, Evgeny Kim, and Roman Klinger. 2020. GoodNewsEveryone: A corpus of news headlines annotated with emotions, semantic roles, and reader perception. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1554–1566, Marseille, France. European Language Resources Association.

Laura-Ana-Maria Bostan and Roman Klinger. 2018. An analysis of annotated corpora for emotion classification in text. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2104–2119, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Sven Buechel and Udo Hahn. 2017. Readers vs. writers vs. texts: Coping with different perspectives of text understanding in emotion annotation. In *Proceedings of the 11th Linguistic Annotation Workshop*, pages 1–12, Valencia, Spain. Association for Computational Linguistics.

Ankush Chatterjee, Kedhar Nath Narahari, Meghana Joshi, and Puneet Agrawal. 2019. SemEval-2019 task 3: EmoContext contextual emotion detection in text. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 39–48, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Danilo Dessì, Rim Helaoui, Vivek Kumar, Diego Reforgiato Recupero, and Daniele Riboni. 2020. TF-IDF vs word embeddings for morbidity identification in clinical notes: An initial study. In *Proceedings of the First Workshop on Smart Personal Health Interfaces co-located with 25th International Conference on Intelligent User Interfaces, SmartPhil@IUI 2020, Cagliari, Italy, March 17, 2020*, volume 2596 of *CEUR Workshop Proceedings*, pages 1–12. CEUR-WS.org.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural networks*, 18(5-6):602–610.

Shelley Gupta, Archana Singh, and Vivek Kumar. 2023. Emoji, text, and sentiment polarity detection using natural language processing. *Information*, 14(4).

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Vivek Kumar., Simone Balloccu., Zixiu Wu., Ehud Reiter., Rim Helaoui., Diego Recupero., and Daniele Riboni. 2023. Data augmentation for reliability and fairness in counselling quality classification. In *Proceedings of the 1st Workshop on Scarce Data in Artificial Intelligence for Healthcare - SDAIH,*, pages 23–28. INSTICC, SciTePress.

Vivek Kumar, Diego Reforgiato Recupero, Daniele Riboni, and Rim Helaoui. 2021. Ensembling classical machine learning and deep learning approaches for morbidity identification from clinical notes. *IEEE Access*, 9:7107–7126.

Vivek Kumar, Diego Reforgiato Recupero, Rim Helaoui, and Daniele Riboni. 2022. K-lm: Knowledge augmenting in language models within the scholarly domain. *IEEE Access*, 10:91802–91815.

Damilola Omitaomu, Shabnam Tafreshi, Tingting Liu, Sven Buechel, Chris Callison-Burch, Johannes Eichstaedt, Lyle Ungar, and João Sedoc. 2022. Empathic conversations: A multi-level dataset of contextualized conversations. *arXiv preprint arXiv:2205.12698*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Tiberiu Sosea and Cornelia Caragea. 2020. Canceremo: A dataset for fine-grained emotion detection. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8892–8904.

Alper Kursat Uysal and Serkan Gunal. 2014. The impact of preprocessing on text classification. *Information Processing & Management*, 50(1):104–112.

Zixiu Wu, Simone Balloccu, Vivek Kumar, Rim Helaoui, Diego Reforgiato Recupero, and Daniele Riboni. 2023. Creation, analysis and evaluation of annomi, a dataset of expert-annotated counselling dialogues. *Future Internet*, 15(3).

Zixiu Wu, Simone Balloccu, Vivek Kumar, Rim Helaoui, Ehud Reiter, Diego Reforgiato Recupero, and Daniele Riboni. 2022. Anno-mi: A dataset of expert-annotated counselling dialogues. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6177–6181.

Zixiu Wu, Rim Helaoui, Vivek Kumar, Diego Reforgiato Recupero, and Daniele Riboni. 2020. Towards detecting need for empathetic response in motivational interviewing. In *Companion Publication of the 2020 International Conference on Multimodal Interaction*, pages 497–502.

Peixiang Zhong, Di Wang, and Chunyan Miao. 2019. Knowledge-enriched transformer for emotion detection in textual conversations. *arXiv preprint arXiv:1909.10681*.

Findings of WASSA 2023 Shared Task: Multi-Label and Multi-Class Emotion Classification on Code-Mixed Text Messages

Iqra Ameer[1], Necva Bölücü[2], Hua Xu[3], and Ali Al Bataineh[4]

[1]Division of Science and Engineering, The Pennsylvania State University at Abington, Pennsylvania, USA
[2]Data61, CSIRO, Sydney, Australia
[3]Section of Biomedical Informatics and Data Science, Yale School of Medicine, USA
[4]Electrical and Computer Engineering Norwich University, USA

Abstract

We present the results of the WASSA 2023 Shared-Task 2: Emotion Classification on code-mixed text messages (Roman Urdu + English), which included two tracks for emotion classification: multi-label and multi-class. The participants were provided with a dataset of code-mixed SMS messages in English and Roman Urdu labeled with 12 emotions for both tracks. A total of 5 teams (19 team members) participated in the shared task. We summarized the methods, resources, and tools used by the participating teams. We also made the data freely available for further improvements to the task.

1 Introduction

In recent times, the growing number of Internet users and the proliferation of diverse online platforms have led to a significant surge in individuals expressing their opinions and attitudes on government websites, microblogs, and other social media platforms. Consequently, there is growing interest in effectively extracting people's sentiments and emotions towards events from such subjective information. To address this, *Natural Language Processing* (NLP) employs emotion analysis called *Emotion Classification*. Emotion Classification is one of the most challenging NLP tasks, in which a given text is assigned to the most appropriate emotion(s) that best reflect the author's mental state of mind (Tao and Fang, 2020), where emotions can be anger, joy, sadness, surprise, etc. People freely express their feelings, arguments, opinions, and thoughts on social media. Therefore, this task plays a pivotal role in uncovering valuable insights from user-generated content, and more and more attention is being paid to automatic tools for classifying users' emotion(s) from written text. Emotion classification has applications in several domains, including financial marketing (Zhang et al., 2016; Yang et al., 2020; Lysova and Rasskazova, 2019), medicine (Lin et al., 2016; Saffar et al., 2022; Huang et al., 2023), education (Huang and Zhang, 2019; Zhang et al., 2020b; Carstens et al., 2019), etc.

There are two different views on the classification of emotions. Ameer et al. (2020) stated that emotions are dependent; one emotional expression can be linked to multiple emotions (Deng and Ren, 2020). Therefore, the emotion classification problem should be defined as *Multi-Label Emotion Classification* (MLEC). MLEC is the task of assigning all possible emotions for a written text that best presents the author's mental state. The other view is that written data is associated with only one emotion (Ameer et al., 2022), which defines the problem as a *Multi-class Emotion Classification* (MCEC) problem. MCEC is the task of assigning one most dominating emotion to the given piece of text that best represents the mental state of an author.

In this paper, we present the WASSA 2023 Shared Task: Multi-Label and Multi-Class Emotion Classification on Code-Mixed Text Messages. We used the same dataset provided by (Ameer et al., 2022) composed of code-mixed (English + Roman Urdu) SMS messages originally collected for MLEC. Each SMS message is annotated for the absence/presence of 12 multiple emotions (anger, anticipation, disgust, fear, joy, love, optimism, pessimism, sadness, surprise, trust, and neutral (no emotion)) provided by SemEval-2018 Task 1: Affect in Tweets (Mohammad et al., 2018) (see Section 3 for more details). The shared task consists of two tracks:

- Track 1 - MLEC: The formulation of this track is to predict all possible emotion labels from code-mixed SMS messages.
- Track 2 - MCEC: The formulation of this track is to predict a single most dominating emotion from code-mixed SMS messages.

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 587–595
July 14, 2023 ©2023 Association for Computational Linguistics

7 teams participated in this shared task: 3 teams submitted results to *MLEC* and 7 teams submitted results to *MCEC* tracks[1]. The tracks were designed using CodaLab[2], allowing teams to submit one official result during the evaluation phase and multiple results during the training phase. During the evaluation phase, each team was allowed to submit their results by a certain deadline, after which the final submission was considered for ranking. The best result for *Track 1 - MLEC* was Multi-Label Accuracy = 0.9782, and the best result for *Track 2 - MCEC* was Macro F_1 = 0.9329.

The rest of the paper is structured as follows: Section 2 provides an overview of related work. Section 3 presents the details of the datasets for both tracks. The task description is outlined in Section 4, while the official results are presented in Section 5. Section 6 provides a discussion of the various systems that participated in both tracks. Finally, our work is concluded in Section 7.

2 Related Work

In recent years, extensive research has been conducted on emotion classification (Ren et al., 2017; Tang et al., 2019; Zhang et al., 2020a). Among supervised machine learning techniques, Random Forest, Logistic Regression, Naïve Bayes, Support Vector Machine, Bagging, AdaBoost, and Decision Tree are widely used for emotion classification problems (Ameer et al., 2020, 2022; Hadwan et al., 2022; Edalati et al., 2022).

The success of deep learning models in various NLP tasks, including Neural Machine Translation (NMT) (Wang et al., 2017; Song et al., 2019) and Semantic Textual Similarity (STS) (Wu et al., 2021; Zhang and Lan, 2021), has led them to be applied to the emotion classification problem as well. Notably, deep learning models, LSTM (Baziotis et al., 2018; Gee and Wang, 2018), CNN (Kim et al., 2018), GRU (Eisner et al., 2016; Alswaidan and Menai, 2020), GNN (Ameer et al., 2023b) and Transformers (e.g., BERT, XLNet, DistilBERT, and RoBERTa) (Ameer et al., 2020; Ding et al., 2020; Ameer et al., 2022, 2023a) have been utilized in this context.

There have been several efforts in the literature to construct benchmark corpora for emotion classification tasks (Illendula and Sheth, 2019; Demszky et al., 2020; Xu et al., 2015; Saputra et al., 2022; Ashraf et al., 2022; Ilyas et al., 2023). However, the existing efforts have primarily focused on monolingual datasets. In particular, SemEval has organized a number of international competitions (Mohammad et al., 2018; Strapparava and Mihalcea, 2007) that have published monolingual benchmark corpora for MLEC, which serve as valuable resources for developing, comparing, and evaluating approaches. Regarding the code-mixed task, a few benchmark corpora have been developed for MLEC (Vijay et al., 2018; Sinha et al., 2021; Sasidhar et al., 2020; Lee and Wang, 2015; Tan et al., 2020; Plaza-del Arco et al., 2020).

Vijay et al. (2018) developed a Hindi-English code-mixed corpus by collecting 2,866 tweets from the past eight years. The corpus was annotated with Ekman's six emotion labels, including anger, disgust, fear, happiness, sadness, and surprise. Each tweet in the corpus was labeled with its source language and the causal language of the expressed emotion. Another effort by Sinha et al. (2021) involved the development of a Hindi-English code-mixed corpus of 15,997 Facebook status updates. These updates were annotated with emotions such as joy, sadness, anger, fear, trust, disgust, surprise, anticipation, and love. Similarly, Sasidhar et al. (2020) created a Hindi-English code-mixed corpus for single-label emotion classification. This corpus consisted of 12,000 texts gathered from Twitter, Instagram, and Facebook posts. It was manually annotated with three basic emotion labels: happy, sad, and anger.

For Chinese-English code-mixed corpora, Lee and Wang (2015) compiled a multilingual corpus by collecting code-switching data from `Weibo.com`, a popular Chinese social networking website. The corpus contained 2,313 posts annotated with five basic emotions: anger, fear, happiness, sadness, and surprise. The posts covered various domains such as life, finance, service, celebrities, products, and politics, with happiness being the most dominant emotion.

In the context of Malaysian code-mixed corpora, Tan et al. (2020) developed a large Twitter corpus consisting of 295,817 Tweets in the Malaysian language (Malay, Malaysian slang, and English). The corpus was annotated with six basic emotion classes: anger, fear, happiness, love, sadness, and surprise. Additionally, Plaza-del Arco et al. (2020) compiled a multi-label and code-mixed emotion

---

[1] Only 5 of the teams submitted system description papers.

[2] Details of task descriptions, datasets, and results are in CodaLab `https://codalab.lisn.upsaclay.fr/competitions/10864`

corpus based on events in April 2019. The corpus included 7,303 English tweets and 8,409 Spanish tweets. Each tweet was assigned one of Ekman's fundamental emotions, such as anger, surprise, disgust, enjoyment, fear, and sadness, or labeled as neutral or other emotions.

While existing code-mixed corpora mainly focused on English combined with Spanish, Malaysian, Hindi, and other languages for tweets, a benchmark code-mixed (English + Roman Urdu) dataset with proposed models to solve the problem for the MLEC task was lacking. To address this gap, the code-mixed dataset developed by Ameer et al. (2022) for MLEC was used for the shared task by extending the problem for MLEC and MCEC problems.

3 Dataset Compilation Process

The dataset–CM-MEC-21 corpus–utilized for the shared task is developed for the MLEC task and consists of code-mixed (English + Roman Urdu) SMS messages (Ameer et al., 2022). In this section, we first provide the details of the original dataset and then describe the dataset preparation process for the *MCEC* track of the shared task.

The dataset contains code-mixed (English + Roman Urdu) SMS messages which are manually selected from SMS-AP-18 corpus (Fatima et al., 2018) and annotated by three annotators for the presence/absence of 12 emotions as in SemEval-2018 (Mohammad et al., 2018) for the MLEC task. Therefore, we used the dataset for the MLEC track of the shared task since it is already annotated for the MLEC using a set of 12 emotions: anger, anticipation, disgust, fear, joy, love, optimism, pessimism, sadness, surprise, trust, and neutral (no emotion).

For the MCEC track, the annotators annotated each code-mixed (English + Roman Urdu) SMS message with the most dominating emotion among all the labels assigned for MLEC. In cases where a code-mixed SMS message did not convey any particular emotion, only the “neutral" label was assigned.

We randomly split the MLEC and MCEC track datasets into train (80%), development (10%), and test (10%) sets. Table 1 represents the train, development, and test splits. The distributions of emotions for MLEC and MCEC tracks for each set are presented in Tables 2 and 3, respectively. The dataset used in the shared task is publicly available[3].

Track	Train	Dev	Test	Total
MLEC	9530	1191	1191	11912
MCEC	9530	1191	1191	11912

Table 1: Statistical details of train, development, and test set for MLEC and MCEC tracks.

Emotion	Train	Dev	Test
Anger	271	41	35
Anticipation	1046	135	134
Disgust	955	134	124
Fear	522	58	51
Joy	1213	144	142
Love	265	34	34
Neutral	3247	404	394
Optimism	1065	133	121
Pessimism	219	26	29
Sadness	638	65	85
Surprise	281	27	34
Trust	1185	145	160

Table 2: Distribution of emotion labels in the MLEC track.

Emotion	Train	Dev	Test
Anger	226	35	26
Anticipation	832	94	97
Disgust	687	113	98
Fear	453	52	55
Joy	1022	131	123
Love	187	17	24
Neutral	3262	388	399
Optimism	880	110	103
Pessimism	178	29	35
Sadness	486	62	69
Surprise	199	35	28
Trust	1118	125	134

Table 3: Distribution of emotion labels in the MCEC track.

4 Task Description

We set up the tracks in CodaLab[4]. Section 4.1 describe the tracks of the shared task and dataset, resources, and evaluation metrics are explained in Section 4.2.

---

[3] https://github.com/wassa23codemixed/codemixed

[4] https://codalab.lisn.upsaclay.fr/competitions/10864

4.1 Tracks

Track 1 - Multi-Label Emotion Classification (MLEC): The problem of this task is to classify each code-mixed SMS message as “neutral or no emotion" or as one or more of eleven given emotions (anger, anticipation, disgust, fear, joy love, optimism, pessimism, sadness, surprise, trust) that best represent the mental state of the author.

Track 2 - Multi-Class Emotion Classification (MCEC): The problem of this task is to predict an emotion label from the emotion set, as well as *no emotion tag (neutral)* for each code-mixed SMS message.

4.2 Setup

Dataset: Participants are provided with the dataset described in Section 3. Participants are allowed to use external datasets in the training phase or use data augmentation techniques to improve their systems.

Team	Accuracy	Micro F_1	Macro F_1
YNU-HPCC	**0.9782**	**0.9854**	**0.9869**
CTcloud	0.9723	0.9815	0.9833
wsl&zt	0.9110	0.9407	0.9464
baseline	0.7321	0.8514	0.8347

Table 4: Results of the teams participating in the MLEC track.

Emotion	YNU-HPCC	CTcloud	wsl&zt
Anger	86.67	97.80	80.00
Anticipation	90.49	88.81	81.69
Disgust	95.00	95.54	93.12
Fear	97.67	96.00	94.36
Joy	92.13	98.97	86.83
Love	91.97	90.70	90.00
Optimism	96.46	88.95	82.44
Pessimism	89.25	80.00	84.55
Sadness	95.17	98.91	95.75
Surprise	93.33	97.19	97.40
Trust	85.90	85.43	85.44

Table 5: Class-wise MLEC results (*100) of the teams participating in the MLEC track.

Resources and Systems Restrictions: The organizers allowed participants to use any third-party tools, lexical resources, additional train data, or synthetic datasets generated by AI models for the tasks, nor did they apply any restrictions on the participants.

System Evaluation: The official competition evaluation script for MLEC was multi-label accuracy (or Jaccard index), and Macro F_1 was used for MCEC. In addition to the official evaluation metrics, Micro and Macro F_1 scores for MLEC and Accuracy, Macro Precision, and Macro Recall for MCEC were also used as secondary evaluation metrics to provide a different perspective on the results.

5 Results and Discussion

5.1 Multi-Label Emotion Classification

Table 4 presents the main results for the MLEC track. 3 teams submitted their results (2 of them submitted their papers). *YNU-HPCC* ranked first in MLEC track (Multi-label Accuracy = 0.9782), which is very close to team *CTcloud* (Multi-label Accuracy = 0.9723), which ranked second. Table 5 provides the class-wise Macro F_1 results for the teams participating in the MLEC track.

5.2 Multi-Class Emotion Classification

Table 6 presents the main results for the MCEC track. 7 teams submitted their results (5 of them submitted their system description papers), and the best-performing team was *YNU-HPCC* (Macro F_1 = 0.9329).

We also provided class-wise Macro F_1 results of the teams participating in the MCEC track in Table 7 to get more insights. Due to the high frequency in the training set of the dataset, the submitted systems achieved higher Macro F_1 scores for Neutral, Trust, Joy, and Optimism labels compared to other emotion labels.

6 Summary of Participating Systems

WASSA 2023 Shared Task on Multi-Label and Multi-Class Emotion Classification on Code-Mixed Text Messages received 5 system description papers. The results of the systems are represented in Tables 4 and 6 for MLEC and MCEC tracks, respectively. Only two five systems attempted the MLEC and MCEC tasks, while the others did not submit results for the MLEC task.

6.1 Machine Learning Architectures

All systems submitted results to the shared task applied deep learning models for MLEC and MCEC tracks. Table 8 provides a high-level summary of the frequency of architectures and techniques used by multiple systems. There are similarities between the four systems based on transformer-based language models. One system deviated from the others using ChatGPT with prompt tuning for

Team	Macro F_1	Accuracy	Macro Precision	Macro Recall
YNU-HPCC	**0.9329**	**0.9488**	**0.9488**	**0.9488**
CTcloud	0.8917	0.9219	0.9219	0.9219
wsl&zt	0.7359	0.7699	0.7699	0.7699
anedilko	0.7038	0.7313	0.7313	0.7313
baseline	0.7014	0.7298	0.7298	0.7298
PrecogIIITh	0.6061	0.6734	0.6734	0.6734
BpHigh	0.3764	0.5642	0.5642	0.5642

Table 6: Results of the teams participating in the MCEC track.

Emotion	YNU-HPCC	CTcloud	wsl&zt	anedilko	PrecogIIITh	BpHigh
Anger	90.20	80.85	66.67	65.45	53.06	0.00
Anticipation	92.55	87.70	68.11	56.11	58.88	35.64
Disgust	91.63	89.20	67.80	69.32	57.00	37.17
Fear	96.49	91.07	75.73	75.25	60.00	26.53
Joy	94.17	93.23	88.16	80.00	82.20	82.03
Love	91.30	77.27	75.00	72.34	57.89	45.71
Neutral	97.48	95.31	80.09	79.14	73.15	71.93
Optimism	94.34	93.72	74.37	70.94	67.94	58.45
Pessimism	94.29	93.94	67.80	67.69	55.56	0.00
Sadness	94.96	91.04	75.71	77.61	67.16	42.67
Surprise	97.27	84.00	65.22	60.87	28.57	0.00
Trust	94.81	92.72	78.46	69.80	65.93	51.56

Table 7: Class-wise MCEC results (*100) of the teams participating in the MCEC track.

Technique / Model	Submission Count
BERT	1
MBERT	1
RoBERTa	1
XLM-RoBERTa	3
IndicBERT	1
MuRIL	1
XGBClassifier	1
Prompt Tuning	1
Prompt Engineering	1

Table 8: Summary of techniques and architectures used in submissions.

the shard task tracks. Three of the systems applied pre-processing (using an emoticon dictionary (*CTcloud*), English translation of code-mixed sentences using ChatGPT (*PrecogIIITh*), converting multi-class labels to multi-label labels with one hot encoding (*YNU-HPCC*)). Only one of the systems used data augmentation in the training phase (*BpHigh*).

With increasing attention to prompt tuning and prompt engineering for extracting knowledge from language models, two of the five systems attempted prompt tuning and engineering for the tasks.

6.2 Features and Resources

For a given code-mixed text, emotion(s) classification is a challenging task in the NLP domain. Teams were allowed to use external resources, which can be data, a lexicon, or contextual embeddings that can improve the performance of systems. Table 9 provides the details of features and resources used in the submitted system description papers.

Features	# of team	MLEC	MCEC
Emotion lexicon	1	✓	✓
ChatGPT	2		✓
External dataset	1		✓
Framework	2	✓	✓

Table 9: Features and resources used in the submitted system description papers.

The emotion lexicon is created by gathering the icons in the training set and collecting more icons from the Internet [5] (*CTcloud*). ChatGPT is

---

[5] https://en.wikipedia.org/wiki/List$_{o}f_{e}moticonsLastvisited$:

Team Name	# of Authors	MCEC	MLEC	Algorithm
YNU-HPCC	5	✓	✓	Finetune PLM
CTcloud	5	✓	✓	Prompt Tuning
wsl&zt	-	✓	✓	
anedilko	1	✓		Prompt Engineering
Arenborg	-	✓		Finetune PLM
PrecogIIITh	4	✓		Finetune PLM
BpHigh	1	✓		Finetune PLM & Contrastive Learning

Table 10: Summary of all the teams that reported their results

used in the submitted system description papers for translation (*PrecogIIITH*) and prompt engineering (*anedilko*).

Moreover, participants used external datasets in the shared task, such as HS-RU-20 (Khan et al., 2021), Roman Urdu Hate Speech (Rizwan et al., 2020), and Hing-Corpus (Nayak and Joshi, 2022). These datasets are used to train the transformer model with contrastive learning (*BpHigh*).

SetFit[6] (Tunstall et al., 2022) (*BpHigh*) and OpenPrompt[7] (Ding et al., 2021) (*CTcloud*) are used as frameworks in the systems. While SetFit is a framework to build a robust sentence classifier for small datasets that helps finetune sentence transformers on the dataset with contrastive learning, Openprompt is a framework to adapt pre-trained language models (PLMs) to downstream NLP tasks.

6.3 System Specifies

YNU-HPCC, the team ranked first, developed a model using a hybrid dataset approach–combined MLEC and MCEC datasets with a unified multilingual pre-trained model. They applied pre-processing step in the training phase to convert multi-class labels to multi-label labels with one hot encoding. They applied Kullback-Leibler (KL) (Eguchi and Copas, 2006) to obtain mixed annotation labels, combining two tracks and fine-tuning XLM-RoBERTA (Conneau et al., 2019). In inference, they separately obtained the results for two tracks with fine-tuned XLM-RoBERTa.

CTcloud, the team ranked second, applied pre-processing before the training phase, mapping emoticons to textual form using icon-emotion and Unicode-short name mapping to leverage their rich emotional information for the problem. They applied prompt tuning with zero-shot and few-shot approaches for GPT-3. They also applied soft-prompt following Zhu et al. (2022) with manual and soft verbalizer using XLM-RoBERTa (Conneau et al., 2019). The best results are obtained with soft prompts and soft verbalizers. They built their system using OpenPrompt (Zhu et al., 2022). In the experiments, they test base and large versions of XLM-RoBERTa as well as the fine-tuned XLM-ROBERTa for the problem. It is found that when the fine-tuned model is used, only a small amount of prompt tuning is required to obtain satisfactory results. On the other hand, XLM-RoBERTa requires more prompt tuning.

anedilko developed a system for MCEC track with prompt engineering on Chat-GPT API. For the prompts, they chose 100 samples from the training set in terms of the cosine similarity of the samples in the training and development sets using embedding API[8]. They also apply XGB Classifier (Chen and Guestrin, 2016), which used character n-grams as features as the baseline model.

PrecogIIITh fine-tuned multi-lingual transformer-based models, XLM-RoBERTa (Conneau et al., 2019) and IndicBERT (Doddapaneni et al., 2022) for MCEC track. As a third experiment, they used ChatGPT interface[9] to translate code-mixed sentences into English and fine-tuned XLM-RoBERTa with the translated sentences.

BpHigh applied SimCSE (Gao et al., 2021), which uses contrastive learning to obtain sentence embeddings using MuRIL–a transformer-based BERT architecture that supports 17 Indic languages, including English. To train SimCSE, they combined 3 datasets, such as HS-RU-20 (Khan et al., 2021), Roman Urdu Hate Speech (Rizwan et al., 2020), and Hing-Corpus dataset (Nayak and

---

06 – 08 – 2023.

[6] `https://github.com/huggingface/setfit` Last visited: 06-08-2023.

[7] `https://github.com/thunlp/OpenPrompt` Last visited: 06-08-2023.

[8] https://platform.openai.com/docs/guides/embeddings

[9] `https://openai.com/blog/chatgpt`

Joshi, 2022).

Table 10 presents the details of the submitted systems to the shared task.

7 Conclusions

This paper presents a shared task on multi-label and multi-class emotion classification for code-mixed (English and Roman Urdu) SMS messages. We provide a comprehensive overview of the task, including its design, data, evaluation process, results, and participating systems. Through the analysis of the systems, we find that most of them employ fine-tuned pre-trained language models for the task of multi-class emotion classification. While these models have shown success in this domain, our observations indicate the need for additional information to fully leverage their potential. Furthermore, prompt tuning emerges as a prominent area of research, holding great promise for multi-label and multi-class classification tasks, particularly in the context of code-mixed datasets and challenging domains like emotion classification. Finally, prompt engineering emerges as an area that demands further investigation to effectively address the challenges posed by these problems.

References

Nourah Alswaidan and Mohamed El Bachir Menai. 2020. A survey of state-of-the-art approaches for emotion recognition in text. *Knowledge and Information Systems*, pages 1–51.

Iqra Ameer, Noman Ashraf, Grigori Sidorov, and Helena Gómez Adorno. 2020. Multi-label emotion classification using content-based features in twitter. *Computación y Sistemas*, 24(3):1159–1164.

Iqra Ameer, Necva Bölücü, Muhammad Hammad Fahim Siddiqui, Burcu Can, Grigori Sidorov, and Alexander Gelbukh. 2023a. Multi-label emotion classification in texts using transfer learning. *Expert Systems with Applications*, 213:118534.

Iqra Ameer, Necva Bölücü, Grigori Sidorov, and Burcu Can. 2023b. Emotion classification in texts over graph neural networks: Semantic representation is better than syntactic. *IEEE Access*.

Iqra Ameer, Grigori Sidorov, Helena Gomez-Adorno, and Rao Muhammad Adeel Nawab. 2022. Multi-label emotion classification on code-mixed text: Data and methods. *IEEE Access*, 10:8779–8789.

Noman Ashraf, Lal Khan, Sabur Butt, Hsien-Tsung Chang, Grigori Sidorov, and Alexander Gelbukh. 2022. Multi-label emotion classification of Urdu tweets. *PeerJ Computer Science*, 8:e896.

Christos Baziotis, Nikos Athanasiou, Alexandra Chronopoulou, Athanasia Kolovou, Georgios Paraskevopoulos, Nikolaos Ellinas, Shrikanth Narayanan, and Alexandros Potamianos. 2018. NTUA-SLP at SemEval-2018 Task 1: Predicting Affective Content in Tweets with Deep Attentive RNNs and Transfer Learning.

Alta Carstens, Vanessa Hockly, Maria Petronella Koen, and Elizabeth Johanna Pretorius. 2019. An investigation into the use of emotional intelligence for learning analytics. *International Journal of Educational Technology in Higher Education*, 16(1):1–15.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A dataset of fine-grained emotions. *arXiv preprint arXiv:2005.00547*.

Jie Deng and Feng Ren. 2020. Multi-label emotion detection via emotion-specified feature extraction and emotion correlation learning. *IEEE Transactions on Affective Computing*, 11(3):360–373.

Fei Ding, Xin Kang, Shun Nishide, Zhijin Guan, and Fuji Ren. 2020. A fusion model for multi-label emotion classification based on BERT and topic clustering. In *International Symposium on Artificial Intelligence and Robotics 2020*, volume 11574, pages 98–111. SPIE.

Ning Ding, Shengding Hu, Weilin Zhao, Yulin Chen, Zhiyuan Liu, Hai-Tao Zheng, and Maosong Sun. 2021. Openprompt: An open-source framework for prompt-learning. *arXiv preprint arXiv:2111.01998*.

Sumanth Doddapaneni, Rahul Aralikatte, Gowtham Ramesh, Shreya Goyal, Mitesh M Khapra, Anoop Kunchukuttan, and Pratyush Kumar. 2022. IndicXTREME: A Multi-Task Benchmark For Evaluating Indic Languages. *arXiv preprint arXiv:2212.05409*.

Maryam Edalati, Ali Shariq Imran, Zenun Kastrati, and Sher Muhammad Daudpota. 2022. The potential of machine learning algorithms for sentiment classification of students' feedback on MOOC. In *Intelligent Systems and Applications: Proceedings of the 2021 Intelligent Systems Conference (IntelliSys) Volume 3*, pages 11–22. Springer.

Shinto Eguchi and John Copas. 2006. Interpreting kullback–leibler divergence with the neyman–pearson lemma. *Journal of Multivariate Analysis*, 97(9):2034–2040.

Ben Eisner, Tim Rocktäschel, Isabelle Augenstein, Matko Bošnjak, and Sebastian Riedel. 2016. emoji2vec: Learning Emoji Representations from their Description.

Mehwish Fatima, Saba Anwar, Amna Naveed, Waqas Arshad, Rao Muhammad Adeel Nawab, Muntaha Iqbal, and Alia Masood. 2018. Multilingual SMS-based author profiling: Data and methods. *Natural Language Engineering*, 24(5):695–724.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Grace Gee and Eugene Wang. 2018. psyML at SemEval-2018 Task 1: Transfer learning for sentiment and emotion analysis. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 369–376.

Mohammed Hadwan, Mohammed Al-Sarem, Faisal Saeed, and Mohammed A Al-Hagery. 2022. An improved sentiment classification approach for measuring user satisfaction toward governmental services' mobile apps using machine learning methods with feature engineering and smote technique. *Applied Sciences*, 12(11):5547.

Chih-Wei Huang, Bethany CY Wu, Phung Anh Nguyen, Hsiao-Han Wang, Chih-Chung Kao, Pei-Chen Lee, Annisa Ristya Rahmanti, Jason C Hsu, Hsuan-Chia Yang, and Yu-Chuan Jack Li. 2023. Emotion recognition in doctor-patient interactions from real-world clinical video database: Initial development of artificial empathy. *Computer Methods and Programs in Biomedicine*, 233:107480.

Yali Huang and Jinhua Zhang. 2019. A study of the effectiveness of emotion recognition for student learning outcomes in e-learning environments. *Interactive Learning Environments*, 27(7):1019–1032.

Anurag Illendula and Amit Sheth. 2019. Multimodal emotion classification. In *companion proceedings of the 2019 world wide web conference*, pages 439–449.

Abdullah Ilyas, Khurram Shahzad, and Muhammad Kamran Malik. 2023. Emotion Detection in Code-Mixed Roman Urdu-English Text. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 22(2):1–28.

Muhammad Moin Khan, Khurram Shahzad, and Muhammad Kamran Malik. 2021. Hate speech detection in roman urdu. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 20(1):1–19.

Yanghoon Kim, Hwanhee Lee, and Kyomin Jung. 2018. AttnConvnet at SemEval-2018 Task 1: Attention-based Convolutional Neural Networks for Multi-label Emotion Classification.

Sophia Lee and Zhongqing Wang. 2015. Emotion in code-switching texts: Corpus construction and analysis. In *Proceedings of the Eighth SIGHAN workshop on chinese language processing*, pages 91–99.

Kai Lin, Fuzhen Xia, Wenjian Wang, Daxin Tian, and Jeungeun Song. 2016. System Design for Big Data Application in Emotion-Aware Healthcare. *IEEE Access*, 4:6901–6909.

Ekaterina I Lysova and Elena I Rasskazova. 2019. Emotions in financial decision-making: A systematic review. *Journal of Behavioral and Experimental Finance*, 24:100–113.

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. Semeval-2018 task 1: Affect in tweets. In *Proceedings of the 12th international workshop on semantic evaluation*, pages 1–17.

Ravindra Nayak and Raviraj Joshi. 2022. L3Cube-HingCorpus and HingBERT: A Code Mixed Hindi-English Dataset and BERT Language Models. *arXiv preprint arXiv:2204.08398*.

Flor Miriam Plaza-del Arco, Carlo Strapparava, L Alfonso Urena Lopez, and M Teresa Martín-Valdivia. 2020. EmoEvent: A multilingual emotion corpus based on different events. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1492–1498.

Han Ren, Yafeng Ren, Xia Li, Wenhe Feng, and Maofu Liu. 2017. Natural logic inference for emotion detection. In *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*, pages 424–436. Springer.

Hammad Rizwan, Muhammad Haroon Shakeel, and Asim Karim. 2020. Hate-speech and offensive language detection in roman Urdu. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 2512–2522.

Alieh Hajizadeh Saffar, Tiffany Katharine Mann, and Bahadorreza Ofoghi. 2022. Textual emotion detection in health: Advances and applications. *Journal of Biomedical Informatics*, page 104258.

Karen Etania Saputra, Galih Dea Pratama, Andry Chowanda, et al. 2022. Emotion dataset from Indonesian public opinion. *Data in Brief*, 43:108465.

T Tulasi Sasidhar, B Premjith, and KP Soman. 2020. Emotion detection in hinglish (hindi+ english) code-mixed social media text. *Procedia Computer Science*, 171:1346–1352.

S Sinha, K Saxena, and N Joshi. 2021. Detecting Multi-label emotions from code-mixed Facebook Status Updates. *Indian Journal of Science and Technology*, 14(31):2542–2549.

Linfeng Song, Daniel Gildea, Yue Zhang, Zhiguo Wang, and Jinsong Su. 2019. Semantic neural machine translation using AMR. *Transactions of the Association for Computational Linguistics*, 7:19–31.

Carlo Strapparava and Rada Mihalcea. 2007. Semeval-2007 task 14: Affective text. In *Proceedings of the fourth international workshop on semantic evaluations (SemEval-2007)*, pages 70–74.

Kathleen Swee Neo Tan, Tong Ming Lim, and Yee Mei Lim. 2020. Emotion analysis using self-training on Malaysian code-mixed Twitter data. In *International Conferences ICT, Society, and Human Beings*, pages 181–188.

Donglei Tang, Zhikai Zhang, Yulan He, Chao Lin, and Deyu Zhou. 2019. Hidden topic–emotion transition model for multi-level social emotion detection. *Knowledge-Based Systems*, 164:426–435.

Jun Tao and Xiaohui Fang. 2020. Toward multi-label sentiment analysis: a transfer learning based approach. *J. Big Data*, 7(1):10.

Lewis Tunstall, Nils Reimers, Unso Eun Seo Jo, Luke Bates, Daniel Korat, Moshe Wasserblat, and Oren Pereg. 2022. Efficient Few-Shot Learning Without Prompts. *arXiv preprint arXiv:2209.11055*.

Deepanshu Vijay, Aditya Bohra, Vinay Singh, Syed Sarfaraz Akhtar, and Manish Shrivastava. 2018. Corpus creation and emotion prediction for Hindi-English code-mixed social media text. In *Proceedings of the 2018 conference of the North American chapter of the Association for Computational Linguistics: student research workshop*, pages 128–135.

Rui Wang, Andrew Finch, Masao Utiyama, and Eiichiro Sumita. 2017. Sentence embedding for neural machine translation domain adaptation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 560–566.

Xing Wu, Chaochen Gao, Liangjun Zang, Jizhong Han, Zhongyuan Wang, and Songlin Hu. 2021. ESimCSE: Enhanced Sample Building Method for Contrastive Learning of Unsupervised Sentence Embedding. *CoRR*, abs/2109.04380.

Hua Xu, Weiwei Yang, and Jiushuo Wang. 2015. Hierarchical emotion classification and emotion component analysis on Chinese micro-blog posts. *Expert systems with applications*, 42(22):8745–8752.

Yichi Yang, Yang Liu, and Jian Mao. 2020. Sentiment analysis of financial news and its impact on stock price movements. *Journal of Finance and Data Science*, 6(3):310–320.

Dong Zhang, Xincheng Ju, Junhui Li, Shoushan Li, Qiaoming Zhu, and Guodong Zhou. 2020a. Multimodal Multi-label Emotion Detection with Modality and Label Dependence. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3584–3593.

Junlei Zhang and Zhenzhong Lan. 2021. S-SimCSE: Sampled Sub-networks for Contrastive Learning of Sentence Embedding. *CoRR*, abs/2111.11750.

Lei Zhang, Jian Yang, and Zhiming Tang. 2020b. Designing emotional intelligent tutors: A systematic review of affective computing in education. *Journal of Educational Computing Research*, 57(8):2133–2166.

Shuo Zhang, Yuhong Li, Yan Liu, and Hsinchun Chen. 2016. Emotional advertising: A study of the emotional impact of advertising on consumer behavior. *Journal of Financial Services Marketing*, 21(4):288–299.

Yi Zhu, Xinke Zhou, Jipeng Qiang, Yun Li, Yunhao Yuan, and Xindong Wu. 2022. Prompt-learning for short text classification. *arXiv preprint arXiv:2202.11345*.

Emotion classification on code-mixed text messages via soft prompt tuning

Jinghui Zhang[1,*] and **Dongming Yang**[1,*,†] and **Siyu Bao**[1] and **Lina Cao**[1] and **Shunguo Fan**[1]
[1]China Telecom Cloud Technology Co., Ltd
`{zhangjh33,yangdm1,baosy,caoln, fanshg}@chinatelecom.cn`
† Corresponding author

Abstract

Emotion classification on code-mixed text messages is challenging due to the multilingual languages and non-literal cues (i.e., emoticons). To solve these problems, we propose an innovative soft prompt tuning method, which is lightweight and effective to release potential abilities of the pre-trained language models and improve the classification results. Firstly, we transform emoticons into textual information to utilize their rich emotional information. Then, variety of innovative templates and verbalizers are applied to promote emotion classification. Extensive experiments show that transforming emoticons and employing prompt tuning both benefit the performance. Finally, as a part of WASSA 2023, we obtain the accuracy of 0.972 in track MLEC and 0.892 in track MCEC, yielding the second place in both two tracks.

1 Introduction

Emotion plays an important role in social relationships, decision making, etc. Emotion analysis aims to enable machines to learn the emotions contained in textual information, such as conversations, e-commerce reviews and personal blogs (Balabantaray et al., 2012). With the surge of social media, emotion classification receives increasing attentions and brings huge commercial and social implications. [1] Text message is a common form of communication that exists in Twitter, YouTube, etc. Some text messages may contain typos, e.g., "habe" (have), and emotional words, e.g., "hahhhhh", while others may include visual cues such as icons, e.g., ":-)", and emoji. These features bring rich emotion information, but are typically not included in the vocabulary of pre-trained model like BERT (Bidirectional Encoder Representation from Transformers), T5 (Text-to-Text Transfer Transformer)(Raffel et al., 2020), etc. In addition, code-mixing is a critical challenge in emotion classification, which means that two or more languages (Yulianti et al., 2021), such as English and Roman Urdu, are contained in a single piece of text. Although code-mixed text is widely used, research on emotion classification for code-mixed corpus is still scarce (Ameer et al., 2022) since it is more difficult to recognizing emotions in code-mixing languages than in a monolingual language.

Fine-tuning is a widely-used method to improve performance of pre-trained language models (PLMs), however, it is memory-consuming and time-consuming. Compared to fine-tuning, prompt learning is more lightweight, thus has received increasing attentions (Liu et al., 2021a). T5 introduces a unified framework that converts any language problem into a text-to-text format. By introducing different prompts, the model can then adapt to different tasks, e.g., language translation and emotion classification. GPT-3 (General Pre-trained Transformer-3) (Brown et al., 2020) proposes context learning, where models can be applied directly in zero-shot or few-shot tasks without further fine-tuning. Utilizing language prompts helps to probe knowledge in PLMs and obtains better performance (Gu et al., 2021). Prompt learning has been widely applied in tasks of natural language understanding (Zhu et al., 2022; Yeh et al., 2022) and generation (Zheng and Huang, 2021), and even has been adopted in some vision tasks (Kirillov et al., 2023) and multi-modality tasks (Yao et al., 2021).

In this paper, we present our effort in emotion classification on code-mixed text messages. The contributions are summarized as follows:

- A concise emoticon pre-processing method is proposed to transform emoticons into textual information.
- Variety of innovative templates are designed to improve emotion classification on code-mixed text messages.
- Specific verbalizers are applied, which

---

[1]* These authors contributed equally to this work.

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 596–600
July 14, 2023 ©2023 Association for Computational Linguistics

achieves efficient emotion classification with low resource consumption.

2 Related Work

2.1 Emotion Classification

Because of the vagueness of definition and the similarity of some emotions, emotion classification is often seen as a subjective and challenging task.

Gaind et al. (2019) combines two different approaches to extract emotions. The first approach employs several textual features like emoticons, degree words, negations and other grammatical analysis. The second approach adopts algorithms based on machine learning techniques.

Polignano et al. (2019) proposes a classification method based on deep neural networks, which is effective on different emotion classification datasets. They also compare three pre-trained word embeddings for words encoding.

Hasan et al. (2019) uses a dimensional model to define emotion categories. Besides, they propose a soft classification method to measure the probability of assigning messages to each emotion category. In addition, a framework called EmotexStream is developed for real-time emotion tracking.

Ameer et al. (2023) proposes multiple attention mechanisms that reveal the contribution of each word to each emotion. They also investigate the usage of LSTM, the fine-tuning of transformer networks for multi-label emotion classification. Experimental results show that these new transfer learning models are able to outperform current state-of-the-arts on the SemEval-2018 Task-1C dataset.

Ashraf et al. (2022) creates the first multi-label emotion dataset, consisting of six basic emotions from the Urdu Nastalíq script. In addition, they build a set of baseline classifiers and provide insight into these classifiers. The baseline classifiers include machine learning algorithms (i.e., Random Forest, Decision Tree, Sequential Minimal Optimisation, AdaBoostM1 and Bagging), deep learning algorithms and Baseline Based Transformer (i.e., BERT). They use a combination of text representations, which are stylometric-based features, pre-trained word embedding, word-based n-grams, and character-based n-grams.

2.2 Prompt Learning

Prompt learning typically includes the design of prompt templates, the optimization of verbalizers and the selection of the PLM. Prompt templates and verbalizers are usually produced by expert knowledge or generated automatically using search or optimization methods (Hu et al., 2021; Shin et al., 2020; Gao et al., 2021; Liu et al., 2021a). The selection of PLMs needs to take the model structure(such as auto-regressive model), pre-training dataset into consideration.

Liu et al. (2021a) finds that prompt tuning can be effective among different models and natural language tasks. They propose a universal and simple P-tuning v2 method, which proves that prompt tuning can be comparable to fine-tuning, while only 0.1%-3% of the parameters are fine-tuned.

Hambardzumyan et al. (2021) proposes soft verbalizers for prompt learning. They use a continuous vector for each class and adopt dot multiplication between the output of masked language model and the class vector to generate the probabilities.

Lang et al. (2022) finds that co-training can improve the prompt-based learning using unlabelled data. Specifically, co-training can benefit the original prompt model while learning smaller downstream task-specific models.

Han et al. (2022) proposes a prompt tuning method with rules "PTR", which encodes the prior knowledge of a classification task into a rule. They then design sub-prompts based on the rule to adapt the task. Results show that PTR achieves a good trade-off between effectiveness and efficiency in prompt construction compared to the state-of-arts.

In this paper, we propose innovative prompt tuning techniques and bring valuable attempts for emotion classification on code-mixed text messages.

3 Methods

3.1 Emoticon Pre-processing

Text messages may include icons and Unicode emoji, which provide rich emotional information and are overlooked by many researchers. We propose a concise method to convert these features into textual form, which consists of icon-emotion mapping and Unicode-short name mapping.

We first statistically gather the icons in the training set and collect more icons from the Internet as a supplement. An icon-emotion mapping is then constructed based on the visual features of icons. For example, ":-)" is replaced by [joy]. It is worth mentioning that some icons can represent different emotions in different contexts. For example, ";-/" can indicate emotions such as disgust, anger and sadness. In this case, ";-/" is replaced by [disgust,

ID	Template
Manual 1	A text message of [mask]: [x]
Manual 1F	A text message of joy: Good job! A text message of [mask]: [x]
Manual 2	[x] This talk is about [mask].
Manual 2F	Good job! This talk is about joy. [x] This talk is about [mask].

Table 1: Manual templates based on expert knowledge. F: few-shot prompt, [x]: input text message.

anger, sadness], informing the PLM that this text may contain some of these three feelings. Since only text replacements are performed, the computation cost of this operation is trivial.

For Unicode-short name mapping, we convert the Unicode encoding to a CLDR short name, e.g. replacing U+1F600 with [grinning face].

3.2 Prompt Tuning

According to the experience (Liu et al., 2021b), a single word change in prompts could make a drastic difference. Therefore, we tried a variety of prompts.

Firstly, we design textual prompts following the zero-shot and few-shot approach of GPT-3, as shown in Table 1. Considering the short length of the text messages in CM-MEC-21 corpus, we limit the number of manual prompt samples for the few-shot approach.

Then we design several soft prompts following Zhu et al. (2022). We investigate the effect of soft template with different numbers and distribution of soft tokens. The structure of soft template is shown in Fig 1. Finally, we compare the results of the manual verbalizer and the soft verbalizer (Hambardzumyan et al., 2021).

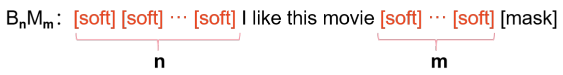

B_nM_m: [soft] [soft] ··· [soft] I like this movie [soft] ··· [soft] [mask]
n
m

Figure 1: Soft template with different numbers and distribution of soft tokens. B: beginning, M: middle. The *n* and *m* denote the numbers of tokens.

For the PLMs, we adopt XLM-RoBERTa with different model scales and pre-training sets. In the process of optimizing the templates and verbalizers, we just freeze the parameters of the pre-trained model, as shown in Fig 2.

Method	MLEC	MCEC
MT+MV	0.024	0.842
MT+MV+EP	0.082	0.856
ST+SV	0.879	0.966
ST+SV+EP	**0.892**	**0.972**

Table 2: Accuracy on test sets for both tracks. M: manual, S: soft, T: template, V: verbalizer, EP: emoticon pre-processing.

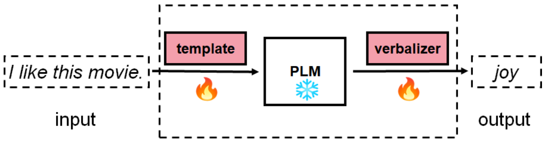


Figure 2: Configuration of prompted model.

Ultimately, the equations for result prediction and objective optimization are noted as

$$\hat{p}_{\nu,\tau}(\mathrm{x}) = \mathrm{softmax}\, V_{\nu}\left(F\left(T_{\tau}(x)\right)\right),$$

$$\{v^*, \tau^*\} = \arg\max_{\nu,\tau} \sum p_{\nu,\tau}(x) \log \hat{p}_{\nu,\tau}(x),$$

where τ, ν denote the parameters of the template and verbalizer respectively. T, F, V represent the template, PLM and verbalizer respectively. The x refers to the input text message, and $\hat{p}$ represents the estimated emotion probability distribution corresponding to the text message while p represents real distribution.

4 Results and Discussion

We adopt CM-MEC-21 corpus (Ameer et al., 2022) in this work, which includes 11650 MCEC messages and 11603 MLEC messages. We combine MCEC and MLEC text messages for both training and evaluating. The accuracy based on the merged dataset for both tracks are summurized in Table 2.

4.1 Emoticon Pre-processing

Firstly, we analyze the effects of emoticon pre-processing. We take XLM-RoBERTa fine-tuned on Roman Urdu corpus[2] as the baseline, where soft template B_5M_6 and manual verbalizer are used but emoticon pre-processing are not employed. As shown in Fig 3, both parts of emoticon pre-processing (i.e., icon-emotion mapping and Unicode-short name mapping) are effective in helping the machine to recognize the emotions.

---

[2] `https://huggingface.co/Aimlab/xlm-roberta-roman-urdu-finetuned`

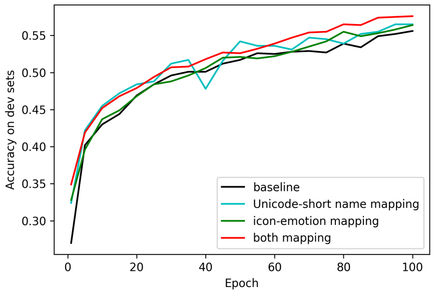


Figure 3: Comparison of different data pre-processing.

4.2 PLMs Comparison

We compare the performance of different PLMs, as shown in Fig 4. We use different XLM-RoBERTa models, where soft template B_5M_6, soft verbalizer and emoticon pre-processing are adopted. We can see that larger models bring better performance. Meanwhile, the language used during pre-training has a significant impact on the final results.

After about 25 epochs of training, XLM-RoBERTa-fine-tuned achieves an accuracy of 0.9, indicating that fine-tuned model can better understand the code-mixed messages. We can draw a conclusion from the experiments that, with the fine-tuned model, just a small amount of prompt tuning can achieve satisfactory results. By contrast, more prompt tuning is required while employing general pre-trained models.

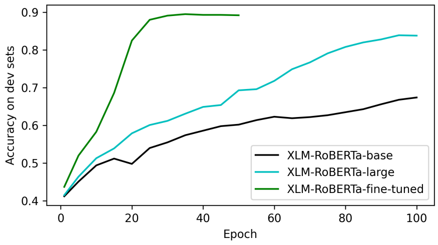


Figure 4: Comparison of different PLMs.

4.3 Prompt Tuning

The combination of template and verbalizer significantly affects classification performance, as shown in Table 3. Although manual prompts are relatively easier to design, their performances are far inferior to that of soft prompts.

We can see from Table 3 that a well-designed prompt template is a prerequisite for efficient classification. Otherwise, the effect of the prompt will be limited, such as Manual 1 and Manual 2F. Also, increasing soft tokens in the template can improve the result, but only to a limited extent.

Template	ID	MV	SV
manual	Manual 1	0.043	0.353
	Manual 1F	0.353	0.353
	Manual 2	0.107	0.887
	Manual 2F	0.353	0.559
soft	B_0M_1	0.434	0.892
	B_0M_3	0.49	0.894
	B_0M_5	0.503	0.896
	B_0M_6	0.551	0.897
	B_1M_6	0.556	0.895
	B_3M_6	0.556	0.892
	$\mathbf{B_5M_6}$	0.576	**0.902**

Table 3: Comparison of different prompt tuning. MV: manual verbalizer, SV: soft verbalizer.

Soft verbalizer can universally benefit the classification performance. For example, while employing template of Manual 2, the soft verbalizer brings an accuracy gain of 0.780 compared with the manual verbalizer. Besides, soft verbalizer enables low resource consumption while maintaining satisfactory performance, e.g., just one soft token in the template can achieve an accuracy of 0.9.

4.4 Limitations

While dealing with the MLEC task, we find that most outputs from our method have only one category. Therefore, our method still has room for improvement in handling MLEC problem.

5 Conclusion

In this paper, we propose the innovative prompt tuning for emotion classification on code-mixed text messages. We first transform emoticons into textual information to utilize their rich emotional information. Then, variety of innovative templates and verbalizers are applied to promote emotion classification. The results show that transforming emoticons benefits the final results. More importantly, even without model fine-tuning, optimizing the prompts yields efficient classification. Finally, we obtain the accuracy of 0.972 in track MLEC and 0.892 in track MCEC, yielding the second place in both two tracks. In future work, we will pay more attention to combining model fine-tuning with prompt learning, and further explore the capacity of PLMs on emotion classification.

References

Iqra Ameer, Necva Bölücü, Muhammad Hammad Fahim Siddiqui, Burcu Can, Grigori Sidorov, and Alexander Gelbukh. 2023. Multi-label emotion classification in texts using transfer learning. *Expert Systems with Applications*, 213:118534.

Iqra Ameer, Grigori Sidorov, Helena Gomez-Adorno, and Rao Muhammad Adeel Nawab. 2022. Multi-label emotion classification on code-mixed text: Data and methods. *IEEE Access*, 10:8779–8789.

Noman Ashraf, Lal Khan, Sabur Butt, Hsien-Tsung Chang, Grigori Sidorov, and Alexander Gelbukh. 2022. Multi-label emotion classification of urdu tweets. *PeerJ Computer Science*, 8:e896.

Rakesh C Balabantaray, Mudasir Mohammad, and Nibha Sharma. 2012. Multi-class twitter emotion classification: A new approach. *International Journal of Applied Information Systems*, 4(1):48–53.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Bharat Gaind, Varun Syal, and Sneha Padgalwar. 2019. Emotion detection and analysis on social media. *arXiv preprint arXiv:1901.08458*.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. 2021. Ppt: Pre-trained prompt tuning for few-shot learning. *arXiv preprint arXiv:2109.04332*.

Karen Hambardzumyan, Hrant Khachatrian, and Jonathan May. 2021. Warp: Word-level adversarial reprogramming. *arXiv preprint arXiv:2101.00121*.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2022. Ptr: Prompt tuning with rules for text classification. *AI Open*, 3:182–192.

Maryam Hasan, Elke Rundensteiner, and Emmanuel Agu. 2019. Automatic emotion detection in text streams by analyzing twitter data. *International Journal of Data Science and Analytics*, 7:35–51.

Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Jingang Wang, Juanzi Li, Wei Wu, and Maosong Sun. 2021. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. *arXiv preprint arXiv:2108.02035*.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. 2023. Segment anything. *arXiv preprint arXiv:2304.02643*.

Hunter Lang, Monica N Agrawal, Yoon Kim, and David Sontag. 2022. Co-training improves prompt-based learning for large language models. In *International Conference on Machine Learning*, pages 11985–12003. PMLR.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2021a. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021b. Gpt understands, too. *arXiv preprint arXiv:2103.10385*.

Marco Polignano, Pierpaolo Basile, Marco de Gemmis, and Giovanni Semeraro. 2019. A comparison of word-embeddings in emotion detection from text using bilstm, cnn and self-attention. In *Adjunct Publication of the 27th Conference on User Modeling, Adaptation and Personalization*, pages 63–68.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*.

Yuan Yao, Ao Zhang, Zhengyan Zhang, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. 2021. Cpt: Colorful prompt tuning for pre-trained vision-language models. *arXiv preprint arXiv:2109.11797*.

Hui-Syuan Yeh, Thomas Lavergne, and Pierre Zweigenbaum. 2022. Decorate the examples: A simple method of prompt design for biomedical relation extraction. *arXiv preprint arXiv:2204.10360*.

Evi Yulianti, Ajmal Kurnia, Mirna Adriani, and Yoppy Setyo Duto. 2021. Normalisation of indonesian-english code-mixed text and its effect on emotion classification. *International Journal of Advanced Computer Science and Applications*, 12(11).

Chujie Zheng and Minlie Huang. 2021. Exploring prompt-based few-shot learning for grounded dialog generation. *arXiv preprint arXiv:2109.06513*.

Yi Zhu, Xinke Zhou, Jipeng Qiang, Yun Li, Yunhao Yuan, and Xindong Wu. 2022. Prompt-learning for short text classification. *arXiv preprint arXiv:2202.11345*.

PrecogIIITH@WASSA2023: Emotion Detection for Urdu-English Code-mixed Text

Bhaskara Hanuma Vedula[†] **Prashant Kodali**[†] **Manish Shrivastava**[†]
Ponnurangam Kumaraguru[†]
[†]International Institute of Information Technology Hyderabad
{vedula.hanuma, prashant.kodali}@research.iiit.ac.in
{m.shrivastava, pk.guru}@iiit.ac.in

Abstract

Code-mixing refers to the phenomenon of using two or more languages interchangeably within a speech or discourse context. This practice is particularly prevalent on social media platforms, and determining the embedded affects in a code-mixed sentence remains as a challenging problem. In this submission we describe our system for WASSA 2023 Shared Task on Emotion Detection in English-Urdu code-mixed text. In our system we implement a multiclass emotion detection model with label space of 11 emotions. Samples are code-mixed English-Urdu text, where Urdu is written in romanised form. Our submission is limited to one of the subtasks - Multi Class classification and we leverage transformer-based Multilingual Large Language Models (MLLMs), XLM-RoBERTa and Indic-BERT. We fine-tune MLLMs on the released data splits, with and without pre-processing steps (translation to english), for classifying texts into the appropriate emotion category. Our methods did not surpass the baseline, and our submission is ranked sixth overall.

1 Introduction

Emotion Detection, which involves understanding the emotion expressed in a given text or conversation, is a widely popular task in the field of natural language processing (Peng et al., 2022). While significant research has been conducted to identify emotions in monolingual languages, the prevalence of code-mixing, particularly on social media, has made this task more challenging. Code-mixing refers to the practice of switching between two or more languages within a single discourse. While classifying emotion as positive or negative is a relatively simple task (binary classification), accurately classifying emotions into 12 categories is comparatively more challenging, due to the higher number of class labels.

In this submission, we describe our methodology and results for our submission to the WASSA 2023 Shared Task on Multi-Class Emotion Classification on Code-Mixed text messages. The data for this task was collected as part of a study (Ameer et al., 2022) that aimed to address the lack of exploration in multi-label emotion classification within code-mixed text. Specifically, the study focused on English and Roman Urdu, a language combination commonly used by the South Asian community in social media posts, comments, tweets, and SMS messages. The study presents a large benchmark corpus of 11,914 code-mixed SMS messages, manually annotated for 12 emotions, including anger, anticipation, disgust, fear, joy, love, optimism, pessimism, sadness, surprise, trust, and neutral (no emotion). Organisers use this particular dataset for the shared task.

The WASSA-2023 shared task has two tracks: Track 1: Multi-Label Emotion Classification (MLEC), where participants had to classify code-mixed SMS messages as either neutral/no emotion, or assign one or more of the eleven emotions that best represented the author's mental state. In contrast, Track 2: Multi-Class Emotion Classification (MCEC), required participants to classify code-mixed SMS messages as either neutral or no emotion, or assign one of the eleven emotions that best represented the author's mental state.

In our submission, we participated in Track 2 of the shared task. We used transformer-based multilingual models, such as XLM Roberta (Conneau et al., 2019) and Indic Bert (Kakwani et al., 2020) to fine-tune on the given dataset. MLLMs are trained on multiple languages covering high and low resource languages. However, MLLMs are known to under perform on low-resource languages. To leverage a model's propensity to perform well for high resource language like English, we also translated the code-mixed sentences into English using the OpenAI API [1]. Fine-tuned XLM-R outperformed all our other approaches. The rest of the

---

[1] https://openai.com/blog/chatgpt

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 601–605
July 14, 2023 ©2023 Association for Computational Linguistics

paper is organized as follows: Section 2 describes the related work; Section 3 describes the implementation in detail as well as the experimental setup; Section 4 covers the results of our experimentation; and we end with Section 5 discussing the implications, and limitations of our current submission along with possible avenues for future work.

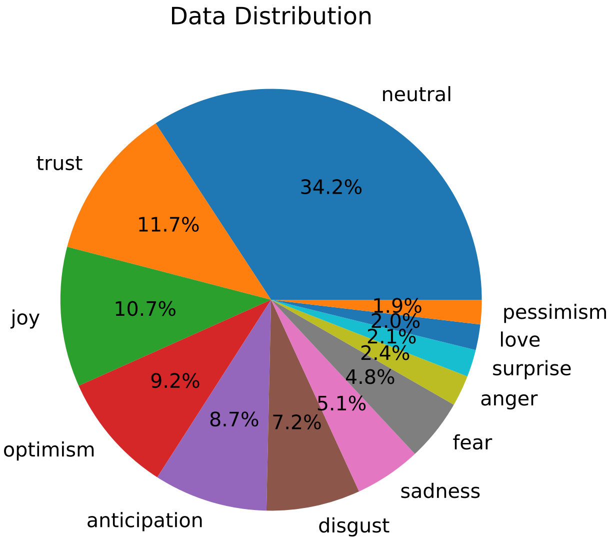


Figure 1: Pie chart showing the distribution of 12 emotions, anger, anticipation, disgust, fear, joy, love, optimism, pessimism, sadness, surprise, trust, and neutral, in the data. Neutral emotions have the highest percentage in the data, followed by a smaller percentage of other emotions.

2 Related Work

The rise of non-native English speakers on social media has led to increased interest in sentiment and emotion analysis of code-mixed data and several works have been done.

A study Ilyas et al. (2023) focused on Roman Urdu (UR) and English (EN) code-mixed text reveals the absence of a dedicated code-mixed emotion analysis corpus. To address this, the authors collect 400,000 sentences from social media, identify 20,000 UR-EN code-mixed sentences, and develop emotion detection guidelines. A large UR-EN-Emotion corpus is created, and experiments done by the authors demonstrate the effectiveness of CNN with GloVe embeddings and the improved use of the developed corpus.

Wadhawan and Aggarwal (2021) introduced a Hinglish dataset labeled for emotion detection and proposed a deep learning approach for detecting emotions in Hindi-English code-mixed tweets. The approach utilizes bilingual word embeddings from FastText and Word2Vec, as well as transformer models such as BERT, RoBERTa, and ALBERT. Experimental results show that the transformer-based BERT model achieves the highest accuracy of 71.43 percent, outperforming other models considered in the study.

Set	Number of Samples
Train	9530
Test	1191
Val	1191

Table 1: Dataset statistics

Ghosh et al. (2023) has done research on Hindi-English code-mixed texts by creating an emotion-annotated Hindi-English dataset through annotations of the SentiMix benchmark dataset. The researchers propose a transformer-based multitask framework for sentiment detection and emotion recognition, utilizing the pre-trained XLMR model. Their multitask solution outperforms both single-task and multitask baselines, obviating the need for ensemble techniques and showcasing its efficiency and applicability in practical natural language processing (NLP) applications.

3 System Description

In this section, we present details about the dataset, along with details about our experiments.

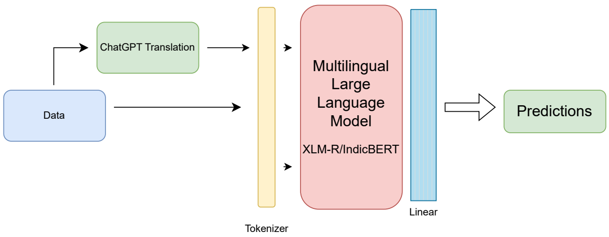


Figure 2: System Architecture for Multi Class Emotion classification using XLM-Roberta and Indic-BERT. Translated code-mixed sentences were also used as input using ChatGPT API.

3.1 Data

Dataset used in this shared task is a collection of texts that contain a mix of Roman Urdu and English language, along with corresponding labels indicating the emotions expressed in the text. This dataset was sampled from a benchmark corpus that was published in (Ameer et al., 2022) study. There were total 12 emotions in dataset which also included no emotion. Data distribution over the 12 labels is visualized in Figure 1. The distribution of the data across training, validation, and test sets is tabulated in Table 1. Dataset is imbalanced with 34% of the samples being labeled as Neutral/no emotion. In the dataset released for the task, English-Urdu sentences were written in roman form. We translated the sentences to English with the intention to leverage MLLM's propensity to perform well in high-resource settings like English.

Using the MLLMs and the datasets we conducted three experiments to evaluate the effectiveness of different language models for code-mixed emotion classification.

3.2 Setup

We use MLLMs models available in Huggingface library (Wolf et al., 2020), and train them using PyTorch (Paszke et al., 2019). We use AdamW optimizer with default hyper-parameters for optimizing our network. We train each model for 50 epochs. For the ideal learning rate, we ran multiple runs with learning rates between [0.05, 0.0005]. Finally, we trained all our models with learning rate of 0.0005. We evaluate the model's performance using standard metrics such as accuracy, recall, precision, and F1 score, which are computed using the scikit-learn (Pedregosa et al., 2011) package.

3.3 Experiments

Fig. 2 shows the system architecture for the current submission. We describe the details pertaining to our experiments and models below. We have used transformer based encoder models - XLM-RoBERTa (Conneau et al., 2019) and Indic BERT (Kakwani et al., 2020).

XLM-Roberta Our initial experiments involved the use of XLM-Roberta. XLM-Roberta is an extension of the RoBERTa model, and it was trained on 2.5 terabytes of filtered Common Crawl data in 100 different languages.

Indic-BERT In our second experiment, we utilized Indic-BERT, another state-of-the-art language model that is specifically designed to handle code-mixed language data in the Indian subcontinent. Indic-BERT is based on the BERT architecture and was pre-trained on a large corpus of code-mixed text in 11 different Indian languages.

We choose these models because of their pre-training on multiple languages, including Indian languages. It is worth noting that romanized Hindi sentences were part of XLM-R's training corpus. Given the linguistic similarities between Hindi and Urdu, we hypothesize that the chosen models will perform well on the downstream task.

Translation: For our third experiment, we explored the use of translation to improve the performance of our emotion classification models. Specifically, we utilized ChatGPT, a language model that is capable of generating human-like text in multiple languages. ChatGPT is based on the GPT-3.5 architecture and was pre-trained on a massive corpus of diverse text. We translated code mixed sentences using OpenAI's official API. For translating the sentences we used the following prompt "Translate roman Urdu English code mixed "sentence" into English". An example of such translation is: "OK mae internet sae dekh or btata hoon" is translated to "OK, let me check on the internet and I will inform you". We, then, concatenated original data and translated English data before passing them into both the XLM-Roberta and Indic-BERT models for code-mixed emotion classification and conducted the experiment in the similar fashion described above.

We added a single MLP on top of MLLMs and trained the models using the PyTorch framework and validated it on the released validation set to determine the accuracy, precision, recall, and Macro F1 score. Predictions for test set made using the final checkpoint of the trained model were submitted for final evaluation. We describe the results for all our experiments in subsequent section.

4 Results

Table 2 presents the results of our experiments, including the F1 scores obtained by each model, allowing for easy comparison of their respective performances. The results indicate that XLM-Roberta achieved the highest F1 score among the models tested (with an F1 score of 0.60). On the other hand, Indic-BERT had the lowest F1 score among the models tested, with an F1 score of 0.54. These results demonstrate the superior performance of XLM-Roberta in the task of code-mixed emotion classification in the English-Urdu dataset. We conjecture that XLM-R's better performance can be attributed to the presence of romanized Hindi in its pre-training corpus.

Initially, we expected Indic-BERT to outperform XLM-Roberta, since IndicBERT is trained on only Indian languages (12 Indian languages), whereas XLM-R is trained on 100 languages across the

	Accuracy	Precision	Recall	Macro-F1 Score
XLM-Roberta	**0.67**	**0.67**	**0.67**	**0.60**
Indic-BERT	0.59	0.59	0.59	0.54
Translation - XLM-Roberta	0.63	0.63	0.63	0.57

Table 2: Performance Metrics Comparison of XLM-Roberta and Indic-BERT Models, and Translation using XLM-Roberta, in terms of Accuracy, Precision, Recall, and Macro-F1 Score. XLM-Roberta outperformed the other two models with highest F1 Score of 0.60.

world. However, the Indic-BERT model produced unsatisfactory results. This could be attributed to multiple reasons - incorrect spellings in the data due to romanization, IndicBERT's lack of familiarity with romanized Hindi/Urdu, making it challenging for the model to accurately capture the nuances of the emotions expressed in the text.

Initially, we hypothesized that translating the code-mixed text into English would result in better performance. However, the pre-trained models employed in our experiments failed to support this hypothesis. Surprisingly, the XLM-Roberta model outperformed the models that included translated texts. We also attempted to augment the dataset by appending translated texts to the original code-mixed data. However, this approach did not significantly improve the performance of the models. FI score obtained for this experiment was 0.57.

5 Conclusion

Based on our experimental results, XLM-Roberta has demonstrated the best performance among the approaches we tested. But our approaches couldn't match the performance of the baseline released as part of the shared task - fine-tuned mBERT.

Future Work Investigating effectiveness of ensemble methods for the task could be fruitful direction for future work. Continued pre-training on code-mixed corpora before fine-tuning on the task-specific dataset could also lead to better results. However, creating such code-mixed corpora for pre-training is non-trivial, and synthetic code-mixed corpora can be leveraged.

Limitations While the multilingual models employed in this study are capable of processing a range of languages, their performance is restricted when it comes to code-mixed sentences that feature a combination of Roman Urdu and English. This limitation suggests that the models may yield comparable results when dealing with similar language pairs. Additionally, the effectiveness of utilizing ChatGPT's API to translate code-mixed sentences into English has not been conclusively established, and thus, it remains uncertain whether this approach represents the optimal solution.

References

Iqra Ameer, Grigori Sidorov, Helena Gómez-Adorno, and Rao Muhammad Adeel Nawab. 2022. Multi-label emotion classification on code-mixed text: Data and methods. *IEEE Access*, 10:8779–8789.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Soumitra Ghosh, Amit Priyankar, Asif Ekbal, and Pushpak Bhattacharyya. 2023. Multitasking of sentiment detection and emotion recognition in code-mixed hinglish data. *Knowledge-Based Systems*, 260:110182.

Abdullah Ilyas, Khurram Shahzad, and Muhammad Kamran Malik. 2023. Emotion detection in code-mixed roman urdu - english text. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 22(2).

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLPSuite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages. In *Findings of EMNLP*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Sancheng Peng, Lihong Cao, Yongmei Zhou, Zhouhao Ouyang, Aimin Yang, Xinguang Li, Weijia Jia, and Shui Yu. 2022. A survey on deep learning for textual emotion analysis in social networks. *Digital Communications and Networks*, 8(5):745–762.

Anshul Wadhawan and Akshita Aggarwal. 2021. Towards emotion recognition in hindi-english code-mixed data: A transformer based approach. *CoRR*, abs/2102.09943.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

BpHigh at WASSA 2023: Using Contrastive Learning to build Sentence Transformer models for Multi-Class Emotion Classification in Code-mixed Urdu

Bhavish Pahwa
Mindtickle / Pune, India
`bhavishpahwa@gmail.com`

Abstract

In this era of digital communication and social media, texting and chatting among individuals occur mainly through code-mixed or Romanized versions of the native language prevalent in the region. The presence of Romanized and code-mixed language develops the need to build NLP systems in these domains to leverage the digital content for various use cases. This paper describes our contribution to the subtask MCEC of the shared task WASSA 2023:Shared Task on Multi-Label and Multi-Class Emotion Classification on Code-Mixed Text Messages. We explore how one can build sentence transformers models for low-resource languages using unsupervised data by leveraging contrastive learning techniques described in the SIMCSE paper and using the sentence transformer developed to build classification models using the SetFit approach. Additionally, we'll publish our code and models on GitHub and HuggingFace, two open-source hosting services.

1 Introduction

The WASSA 2023 Shared Task on Multi-Label and Multi-Class Emotion Classification on Code-Mixed Text Messages (Ameer et al., 2022) aims at building multi-class and multi-label classification systems to detect if a code-mixed text message has neutral emotion or any of the eleven provided emotions which accurately describe the sentiment behind the text message and the author's emotional state. These eleven emotions are trust, joy, optimism, anticipation, disgust, sadness, fear, anger, surprise, love, and pessimism. The core purpose of the shared task is to understand how robust and accurate NLP systems can be built to perform NLU tasks like emotion detection. Many researchers have tried to approach NLU tasks like sentiment classification in code-mixed Urdu earlier and have been attempting to make robust systems to understand how accurately NLP systems can understand code-mixed Urdu (Sharf and Rahman, 2018). To begin, code-mixed Urdu may include words and phrases from many languages, including English, Urdu, and Hindi. This makes it challenging for NLP systems to reliably identify the language of each word and decide the appropriate language model to apply for text processing.

Second, code-mixed Urdu might feature complicated linguistic phenomena such as code-switching, the practice of switching between languages within a sentence or discourse. This can make it challenging for NLP systems to effectively recognize language borders and decide the appropriate language model to apply for each section of the text.

Finally, code-mixed Urdu may contain transliterated words, loanwords, and other linguistic elements not found in conventional Urdu or English. This can make it challenging for NLP algorithms to recognize and understand these phrases effectively. Researchers are building more advanced NLP models based on the transformer architecture designed to handle code-mixed text to meet these problems. These models employ transfer learning approaches, which entail pre-training a model on a vast dataset of code-mixed text before fine-tuning it for a specific purpose. Processing and interpreting code-mixed Urdu and other code-mixed languages are becoming more viable using these more complex models (Devlin et al., 2019).

Many researchers have also started building sentence transformer models by training pre-trained transformer models based on the Sentence-BERT paper (Reimers and Gurevych, 2019) using the sentence transformers framework[1]. These trained sentence transformers can generate sentence embedding vectors, which can be used for many downstream tasks like classification, clustering, and information retrieval. The significant advantage of sentence transformers is that the embedding vectors they generate can capture the respective text's syntactic and semantic meaning.

---

[1] `https://www.sbert.net/index.html`

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 606–610
July 14, 2023 ©2023 Association for Computational Linguistics

Emotion Label	Number of Samples
neutral	3262
trust	1118
joy	1022
optimism	880
anticipation	832
disgust	687
sadness	486
fear	453
anger	226
surprise	199
love	187
pessimism	178

Table 1: Train Dataset Description

Emotion Label	Number of Samples
neutral	388
trust	125
joy	131
optimism	110
anticipation	94
disgust	113
sadness	62
fear	52
anger	35
surprise	35
love	17
pessimism	29

Table 2: Dev Dataset Description

This paper describes our approach based on training a sentence transformers model using the pre-trained MURIL (Khanuja et al., 2021) transformer model based on the BERT architecture. We leverage contrastive learning techniques described in the SIMCSE paper (Gao et al., 2021) to train our sentence transformer model on unsupervised data in Romanized Urdu and Hindi. We call this trained sentence transformer model MURIL-SIMCSE. We further utilize the SetFit framework[2] (Tunstall et al., 2022) to fine-tune our MURIL-SIMCSE model on the training dataset of the Multi-class Emotion Classification (MCEC) subtask of the shared task to perform emotion detection in a multi-class prediction setting.

We will release all our code on GitHub[3] and fine-tuned models on HuggingFace[4] .

2 Dataset Description

The dataset of the MCEC track of the shared task consists of three subsets, namely the train, dev, and test set. The train and dev set consists of examples wherein we have the code-mixed sms message and the respective emotion label assigned to the message. The test set contains the code-mixed sms messages on which the approach will be tested and the gold labels against which the predicted labels will be compared. Table 1 and Table 2 describes the train and dev datasets for the number of examples in each emotion label.

3 Related Work

Reimers and Gurevych (2019) released the Sentence BERT architecture, constructed by altering BERT. The method employs Siamese and triplet network topologies on top of a BERT network to construct sentence embeddings with considerable semantic information. These sentence embeddings can be used for downstream tasks like clustering, classification, and information retrieval. Furthermore, sentence transformer models can be trained by introducing a pooling layer on top of any pre-trained transformer model and by using annotated datasets that can inform the model that a pair of sentences have a degree of semantic similarity or a triplet where two sentences have a certain similarity. The third example is supposed to be dissimilar from the other two.

Khanuja et al. (2021) released a research paper and a new transformer model based on BERT architecture called MuRIL, which was trained in English and 16 other languages spoken in the Indian subcontinent region. The 16 other languages are Assamese (as), Bengali (bn), Gujarati (gu), Hindi (hi), Kannada (kn), Kashmiri (ks), Malayalam (ml), Marathi (mr), Nepali (ne), Oriya (or), Punjabi (pa), Sanskrit (sa), Sindhi (sd), Tamil (ta), Telugu (te) and Urdu (ur). It was trained using Masked language modeling and translation language modeling objectives. The authors show that MuRIL can outperform mBERT on the XTREME benchmark (Hu et al., 2020), Multilingual BERT (mBERT) achieves an average performance of 59.1, whereas MURIL achieves an average performance of 68.6. In the XNLI sentence classification task (Conneau et al., 2018), the MuRIL transformer has an accuracy of 67.7 in Urdu, whereas mBERT has an accuracy of 58.2.

Gao et al. (2021) shows how unsupervised datasets like simple text input sentences can be used to train and build sentence transformer models. In the unsupervised SIMCSE approach, the same input text sentence is passed to the pre-trained

---

[2] `https://github.com/huggingface/setfit`

[3] `https://github.com/bp-high/WASSA_Code-Mixed_Shared_Task`

[4] `https://huggingface.co/bpHigh`

encoder twice; as we use standard dropout, the two sentence embeddings of the same input sentence passed twice will be at slightly different positions. While training, the distance to other embeddings of the other sentences in the same batch (which serve as negative examples) will be maximized. The distance between these two embeddings of the same input sentence will be minimized.

Tunstall et al. (2022) released a research paper and the SetFit framework to build a robust sentence classifier for small datasets using sentence transformers. The SetFit works by fine-tuning the sentence transformer on the respective dataset using contrastive learning. The fine-tuned sentence transformer is then used to generate sentence embeddings to train the classification layer.

4 Methodology

In this section, we describe our approach to training sentence transformer model based on the MuRIL pretrained transformer and further building multiclass classifier systems by finetuning the MURIL-SIMCSE model on the training data of the multiclass emotion classification track of the shared task.

4.1 Training MURIL-SIMCSE model

For training the sentence transformer on top of MuRIL pretrianed transformer using contrastive learning approach we first build a dataset of input text sentences in Urdu by utilizing previously published code-mixed and romanized Urdu datasets. We use the following two datasets to generate the input sentences in Urdu for the train dataset:-

1. **HS-RU-20** (Khan et al., 2021) [5]

2. **Roman Urdu Hate Speech** (Rizwan et al., 2020) [6]

As both these datasets have text and labels and contain hate/toxic examples to contain bias and toxicity, we filter only the normal/ non-hateful/nontoxic examples from these datasets and curate them for the train dataset. We get 13404 input sentences in Urdu from the above-described datasets, which are relatively low for training contrastive learning-based sentence transformer models. We assume that Hindi is similar in spoken forms to Urdu to increase the number of input sentences. Although they are written in different scripts in the Romanized format, they should be similar. So we add romanized hindi sentences from the **HingCorpus dataset**[7] (Nayak and Joshi, 2022) to our pure romanized Urdu sentences dataset and generate a final dataset of two hundred thousand sentences(200,000).

Model	Number of Iterations	Epochs
MURIL-SIMCSE-SETFIT-V1	8	1
MURIL-SIMCSE-SETFIT-V2	15	2
MURIL-SIMCSE-SETFIT-V3	20	2
MURIL-SIMCSE-SETFIT-V4	25	2

Table 3: Hyperparameters

We train the model for one epoch, with a batch size of 32 using AdamW as the optimizer with WarmupLinear scheduler and 20000 warmup steps, the learning rate being 2e-05.

4.2 Training SetFit based classifiers

Using the trained MuRIL-SIMCSE sentence transformer model, we further develop classifiers using the **SetFit framework** (Tunstall et al., 2022) and the training dataset of the MCEC track. Figure 1 shows the training process according to the SetFit framework.

We train four versions of the SetFit-based classifier using different hyperparameters. The various hyperparameters associated with each version can be found in Table 3. In all the different versions we keep the value of batch size same and the value of batch size is 16. All versions are trained using cosine similarity loss, same learning rate of 2e-05, same seed with value 42, same warmup proportion of 0.1.

5 Results

The result for all the different MURIL-SIMCSE-SETFIT model versions on the test dataset are presented in Table 4.

We notice that the SetFit framework based model's performance improves as we increase the value of the hyperparameter 'number of iterations' while training. This hyperparameter refers to the number of iterations for which the sentence pairs would be generated for sentence transformer fine-tuning process in the SetFit training process. Even though we notice this general trend we also have to consider that although accuracy increased

---

[5] `https://www.kaggle.com/datasets/drkhurramshahzad/hate-speech-roman-urdu`

[6] `https://huggingface.co/datasets/roman_urdu_hate_speech`

[7] `https://github.com/l3cube-pune/code-mixed-nlp`

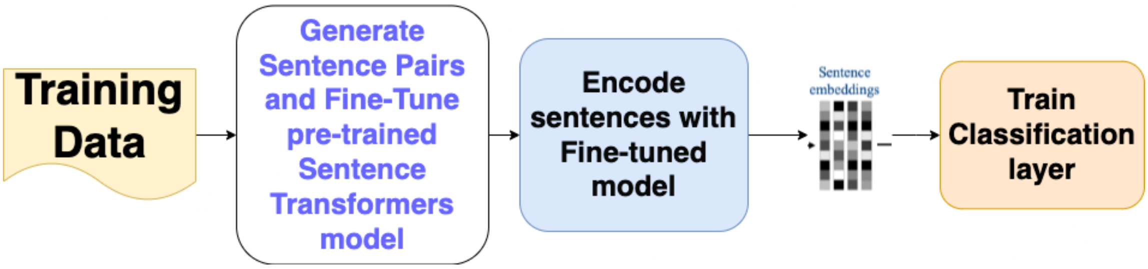

Training Data
Generate Sentence Pairs and Fine-Tune pre-trained Sentence Transformers model
Encode sentences with Fine-tuned model
Sentence embeddings
Train Classification layer

Figure 1: SetFit training process

Model	Macro F1-Score	Recall	Precision	Accuracy
MURIL-SIMCSE-SETFIT-V1	0.3764	0.5642	0.5642	0.5642
MURIL-SIMCSE-SETFIT-V2	0.5657	0.6792	0.6792	0.6792
MURIL-SIMCSE-SETFIT-V3	0.5345	0.6843	0.6843	0.6843
MURIL-SIMCSE-SETFIT-V4	**0.6400**	**0.7044**	**0.7044**	**0.7044**

Table 4: Metric Values of the different SetFit models on the test dataset

from MURIL-SIMCSE-SETFIT-V2 to MURIL-SIMCSE-SETFIT-V3 the Macro-F1 score dipped.

6 Limitations

Train Dataset for MuRIL-SIMCSE: While we try to minimize the hateful samples in this dataset by removing all the toxic/hateful samples of the respective datasets used to form this dataset, there could be samples containing certain biases like gender bias and racial bias. Also the dataset contains the respective languages written in the Roman script, so the results might not be transferable to the respective native scripts of the languages.

MURIL-SIMCSE: The model was trained on a single Tesla P100 GPU for 9 hrs. We could have trained further and on more data, but we could not due to resource and economic constraints.

7 Conclusion

We describe our approach in this paper for the MCEC track of the subtask. We leverage the unsupervised training method using contrastive learning for developing a sentence-transformer model from MuRIL pre-trained model for romanized code-mixed Urdu. We leverage this sentence-transformer model to build multi-class classifiers using the provided training data and the SetFit framework. We show how increasing the value of the hyperparameter number of iterations increases the performance of the classifiers. Further, we will examine how increasing the unsupervised text examples dataset used for training the MURIL-SIMCSE sentence transformer affects the performance of the classifiers built on top of it. We would also look into whether our assumption to mix Romanized Hindi text examples with Urdu examples produces actual benefit or more noise. At the same time, it is not necessary that text examples in Hindi and Urdu would be equivalent even in the Romanized form. For example, Urdu and Hindi speakers romanize the retroflex R differently[8] . Taking the word study as an example, it would be "parho" in Roman Urdu and "padho" in Roman Hindi.

References

Iqra Ameer, Grigori Sidorov, Helena Gómez-Adorno, and Rao Muhammad Adeel Nawab. 2022. Multi-label emotion classification on code-mixed text: Data and methods. *IEEE Access*, 10:8779–8789.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages

---

[8] `https://www.reddit.com/r/Urdu/comments/11vnhbk/question_are_romanized_urdu_and_romanized_hindi/`

4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. XTREME: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4411–4421. PMLR.

Muhammad Moin Khan, Khurram Shahzad, and Muhammad Kamran Malik. 2021. Hate speech detection in roman urdu. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 20(1).

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, Shruti Gupta, Subhash Chandra Bose Gali, Vish Subramanian, and Partha P. Talukdar. 2021. Muril: Multilingual representations for indian languages. *CoRR*, abs/2103.10730.

Ravindra Nayak and Raviraj Joshi. 2022. L3Cube-HingCorpus and HingBERT: A code mixed Hindi-English dataset and BERT language models. In *Proceedings of the WILDRE-6 Workshop within the 13th Language Resources and Evaluation Conference*, pages 7–12, Marseille, France. European Language Resources Association.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Hammad Rizwan, Muhammad Haroon Shakeel, and Asim Karim. 2020. Hate-speech and offensive language detection in Roman Urdu. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2512–2522, Online. Association for Computational Linguistics.

Zareen Sharf and Saif Ur Rahman. 2018. Performing natural language processing on roman urdu datasets. *International Journal of Computer Science and Network Security*, 18(1):141–148.

Lewis Tunstall, Nils Reimers, Unso Eun Seo Jo, Luke Bates, Daniel Korat, Moshe Wasserblat, and Oren Pereg. 2022. Efficient few-shot learning without prompts.

YNU-HPCC at WASSA 2023: Using Text-Mixed Data Augmentation for Emotion Classification on Code-Mixed Text Message

Xuqiao Ran, You Zhang, Jin Wang, Dan Xu and **Xuejie Zhang**
School of Information Science and Engineering
Yunnan University
Kunming, China
Contact: `rxq@mail.ynu.edu.cn`, `yzhang0202@ynu.edu.cn`

Abstract

Emotion classification on code-mixed texts has been widely used in real-world applications. In this paper, we build a system that participates in the WASSA 2023 Shared Task 2 for emotion classification on code-mixed text messages from Roman Urdu and English. The main goal of the proposed method is to adopt a text-mixed data augmentation for robust code-mixed text representation. We mix texts with both multi-label (track 1) and multi-class (track 2) annotations in a unified multilingual pre-trained model, i.e., XLM-RoBERTa, for both subtasks. Our results show that the proposed text-mixed method performs competitively, ranking first in both tracks, achieving an average Macro F_1 score of 0.9782 on the multi-label track and of 0.9329 on the multi-class track.

1 Introduction

Emotion classification is a fundamental task in natural language processing (NLP). The main purpose is to identify the emotions in a written text that potentially represents the author's mental state. Compared with single-label emotion classification, multi-label emotion classification is more difficult to determine all possible emotions instead of only one emotion in a given text. Accordingly, multi-label classification has shown wide applications, such as health care and e-learning (Maxwell et al., 2017).

With the rapid growth of the Internet, linguistic code-mixed culture has become one of the most prominent communication approaches. Code-mixed texts represent texts written by two or more languages, simultaneously. According to Ameer et al. (2022), more than half of Europeans use code-mixed texts in communication. Thus, providing a more accurate judgment about the potential emotional state of such code-mixed texts is essential for various real-world applications, such as author profiling and sentiment analysis (Santosh et al., 2013; Ahmed et al., 2015).

MCEC
Text:Yaro phr Huda and Mara ki *birthdays* ka kya *plan* hai? *I am excited*:D
Label: -Emotions: joy -One-hot: [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
MLEC
Text: *please* jaldi aa jao *we are missing you*
Label: -Emotions: Love, Joy, Trust -One-hot: [0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0]

Figure 1: Examples of code-mixed text messages with multi-class and multi-label emotions. *Italic* and red words present the English language in code-mixed texts. A total of 12 emotional labels are listed in order: neutral, anger, anticipation, disgust, fear, joy, love, optimism, pessimism, sadness, surprise, and trust. Note that *neutral* means no explicit emotions.

Toward this issue, WASSA 2023 proposes a shared task for emotion classification on code-mixed (with Roman Urdu and English) text messages, consisting of two tracks. 1) Track 1: Multi-label Emotion Classification (MLEC). 2) Track 2: Multi-class Emotion Classification (MCEC). Code-mixed texts are given for both tracks. The MLEC requires a system to classify such texts as *neutral* or multi-label (one or more) emotions in given texts while MCEC requires the system for *neutral* or only one emotion that best presents the mental state of the author, as shown in Figure 1.

We participate in both tracks on the shared tasks and found that the main challenges are twofold: 1) Code-mixed texts consist of bilingual languages in a text; 2) Multiple labels are annotated for a code-mixed text. To address these problems, we provide a system that utilizes a text-mixed data augmentation method to handle two tracks at the same time. Initially, we build a shared pair-mixed corpus in a random combination between MLEC

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 611–615
July 14, 2023 ©2023 Association for Computational Linguistics

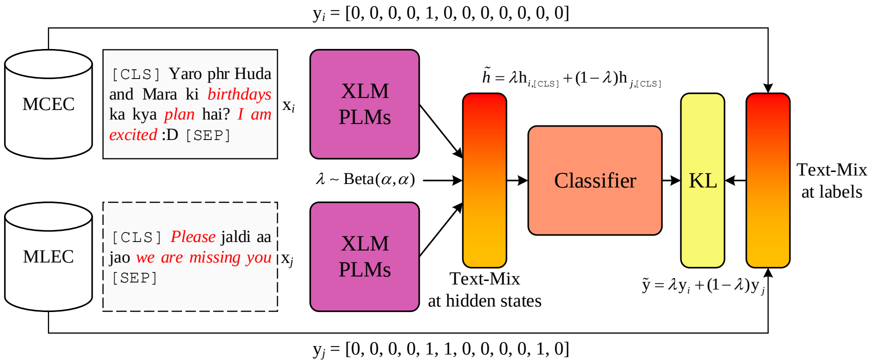

y_i = [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
MCEC
[CLS] Yaro phr Huda and Mara ki *birthdays* ka kya *plan* hai? *I am excited* :D [SEP]
x_i
XLM PLMs
$\tilde{h} = \lambda h_{i,[CLS]} + (1-\lambda) h_{j,[CLS]}$
$\lambda \sim \text{Beta}(\alpha, \alpha)$
Classifier
KL
Text-Mix at labels
MLEC
[CLS] *Please* jaldi aa jao *we are missing you* [SEP]
x_j
XLM PLMs
Text-Mix at hidden states
$\tilde{y} = \lambda y_i + (1-\lambda) y_j$
y_j = [0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0]

Figure 2: The overview of the proposed model.

and MCEC training samples. Next, hidden states and annotated labels in a paired sample are individually mixed in an interpolation method by the same weights. Note that, hidden states in paired texts are generated from multilingual pre-trained language models (PLMs) (Qiu et al., 2020), i.e., XLM-RoBERTa (Conneau et al., 2019), that can align word representations from multilingual tokens and semantics from multilingual sentences. Finally, the predicted probabilities over multiple emotions from the system are converged with mixed annotated labels in a Kullback-Leibler (KL) (Eguchi and Copas, 2006) divergence loss function. Consequently, the proposed system can be shared for both tracks in one training phase. Extensive experiments are conducted to investigate the effect of the proposed method and the best submissions reveal that our system ranks first in both MLEC and MCEC tracks.

The remainder of this paper is constructed as follows. A line of related works is provided in section 2. A detailed description of the proposed system is introduced in section 3. Experimental results are analyzed in section 4. Finally, conclusions are drawn in section 5.

2 Related Work

Emotion classification is a challenging NLP task that aims to automatically classify text documents into one or more predefined emotion categories. It has long been of interest to researchers in areas such as sentiment analysis, opinion mining, and social media analysis.

Recent studies have explored the use of different deep learning architectures, such as Convolutional Neural Networks (CNNs) (Krizhevsky et al., 2017) and Recurrent Neural Networks (RNNs) (Mikolov et al., 2010), to improve the performance of emotion classification. Other studies have focused on the development of pretraining models such as BERT (Devlin et al., 2019), ALBERT (A Lite BERT) (Lan et al., 2019) and RoBERTa (Liu et al., 2019), which have been shown to achieve state-of-the-art results on various NLP tasks, including emotion classification.

A growing body of research has focused on the use of monolingual and cross-lingual models to improve emotion classification. Monolingual methods are based on training models on large amounts of data from a single language, while cross-lingual models make use of data from multiple languages to learn more robust representations. Cross-lingual models such as XLM (Barbieri et al., 2022) have shown promise in many NLP tasks, including emotion classification.

Due to the scarcity of low-resource data, data augmentation is essential. Mixup (Zhang et al., 2017) is a simple and effective data augmentation method, which can significantly improve the effect in multiple fields such as image, text, speech, recommendation. Different variants of mixing methods have also been designed in the different space, the cutMix (Yun et al., 2019) method takes a different approach, instead of interpolating two samples from a numerical perspective, but from the spatial perspective of the image, it cuts a random rectangular area on one picture to another picture to generate a new picture. Manifold mixup (Verma et al., 2018) extends mixup, and extends the mixing of input data (raw input data) to the output mixing of the intermediate hidden layer.

3 Methodology

In this section, we will describe our system that participated in WASSA 2023 shared task on emotion classification on code-mixed text messages. As shown in Figure 2, the proposed models consist of four parts, including pair-mixed corpus, sentence encoder, text-mixed interpolation, and classifier. Before introducing the proposed model, we describe the shared task 2 in advance.

3.1 Emotion Classification on Code-mixed Text

Given a text code-mixed text $\mathbf{x} = [\mathbf{x}_1,\mathbf{x}_2, \ldots, \mathbf{x}_N]$ where N represents the text length. Regarding MCEC and MLEC, it requires a system $f_\theta(\hat{\mathbf{y}}|\mathbf{x})$ to predict ground-truth emotions $\mathbf{y}_c \in \mathbb{R}^C$ and $\mathbf{y}_l \in \mathbb{R}^C$ where θ represents the whole parameters in the system and C is the total number of emotional labels.

3.2 Pair-mixed Corpus

In this paper, we propose a unified system for both MCEC and MLEC tasks, simultaneously. Hence, we initially mix both corpora in a random combination. In detail, for each pair of training sample $((\mathbf{x}_i^c, \mathbf{x}_j^l), (\mathbf{y}_i^c, \mathbf{y}_j^l))$, we random select a code-mixed SMS message from MCEC training set $\mathcal{D}_c$ and another one from MLEC $\mathcal{D}_l$, where $(\mathbf{x}_i^c, \mathbf{y}_i^c) \in \mathcal{D}_c$, $(\mathbf{x}_j^l, \mathbf{y}_j^l) \in \mathcal{D}_l$, $i \in [1 : |\mathcal{D}_c|]$, and $j \in [1 : |D_l|]$ Consequently. a mixed corpus $\mathcal{D}_m = \sum_{i,j}((\mathbf{x}_i, \mathbf{x}_j), (\mathbf{y}_i, \mathbf{y}_j))$ is generated.

3.3 Multilingual Text Encoders

Due to code-mixed texts comprised bilingual languages, aligning word embedding and semantic cross-lingual sentences or phrases is critical for robust text representations. To this end, we adopt a cross-lingual PLM, XLM-RoBERTa, as the sentence encoder to encode paired texts into hidden spaces, formulated as:

$$\mathbf{h}_i, \mathbf{h}_j = \mathbf{XLM}((\mathbf{x}_i, \mathbf{x}_j)), \tag{1}$$

where$(\mathbf{x}_i, \mathbf{x}_j) \sim \mathcal{D}_m$, $\mathbf{h}_i, \mathbf{h}_j \in \mathbb{R}^{N\times d}$; d is the dimensionality of hidden states.

3.4 Text-mixed Interpolation

To further mix up $(\mathbf{x}_i, \mathbf{y}_i)$ and $(\mathbf{x}_j, \mathbf{x}_j)$, a mixed interpolation method is proposed. Based on sentence representation $\mathbf{h}_i$ and $\mathbf{h}_j$, we choose `[CLS]` (a special token in PLMs) to represent the global sentence representation mixed in hidden spaces (denoted as $\tilde{h} \in \mathbb{R}^d$), as well as annotated labels (denoted as $\tilde{\mathbf{y}} \in \mathbb{R}^C$), as shown in Figure 2.

Dataset	Instances
Train	9530
Dev	1191
Test	1191

Table 1: Data distribution.

$$\begin{aligned} &\tilde{h} = \lambda \mathbf{h}_{i,\texttt{[CLS]}} + (1-\lambda)\mathbf{h}_{j,\texttt{[CLS]}} \\ &\tilde{\mathbf{y}} = \lambda \mathbf{y}_i + (1-\lambda)\mathbf{y}_j \\ &\lambda \sim \mathbf{Beta}(\alpha, \alpha) \\ &\lambda = \mathbf{max}(\lambda, 1-\lambda) \end{aligned}, \tag{2}$$

where $\lambda \in [0, 1]$ is generated from Beta distribution with a hyper-parameter α .

3.5 Training Objective and Inference Strategy

In this section, we introduce the proposed system in training and inference phases.

Training objective. To predict emotional probabilities $\hat{\mathbf{y}} \in \mathbb{R}^C$, we apply a Multi-Layer Perceptron (MLP) to encode mixed textual hidden states $\tilde{h}$:

$$\begin{aligned} \hat{\mathbf{y}} &= \mathbf{MLP}(\tilde{h}) \\ &= \mathbf{Linear}(\tanh(\mathbf{Dropout}(\tilde{h}))) \end{aligned}, \tag{3}$$

where $\mathbf{Linear}(\cdot), \mathbf{tanh}(\cdot)$ and $\mathbf{Dropout}(\cdot)$ are fully-connected layer, hyperbolic tangent activation function, and dropout function, respectively. The loss function between predicted probabilities and mixed labels is KL divergence (Eguchi and Copas, 2006) for the system training:

$$\mathcal{L} = \mathbf{KL}(f(\hat{\mathbf{y}}|(\mathbf{x}_i, \mathbf{x}_j); \theta)||\tilde{\mathbf{y}}). \tag{4}$$

Inference strategy. In the inference phase, MCEC and MLEC test datasets are separately fed into the system for individual purposes. Note that there are not mixed procedures in the inference phase. For MCEC task, the argmax function is used to predict one emotion; for MLEC task, a threshold score of 0.5 is used to predict all most possible emotions, formulated as:

$$\begin{aligned} \hat{\mathbf{y}}_c &= \mathbf{argmax}(\mathbf{softmax}(\hat{\mathbf{y}})) \\ \hat{\mathbf{y}}_l &= \mathbf{threshold}(\mathbf{sigmoid}(\hat{\mathbf{y}}), 0.5) \end{aligned}. \tag{5}$$

4 Experimental Results

In this section, extensive experiments were conducted for both MCEC and MLEC tracks.

Model	MCEC				MLEC		
	Mac-P	Mac-R	Mac-F_1	Acc	Mac-F_1	Mic-F_1	Acc
BERT	0.70	0.70	0.69	0.70	0.58	0.64	0.57
mBERT	0.71	0.70	0.70	0.70	0.59	0.66	0.59
mBERT+Text-mixed	0.93	0.92	0.92	0.92	0.79	0.85	0.79
RoBERTa	0.71	0.70	0.70	0.70	0.50	0.63	0.56
XLM-RoBERTa	0.73	0.73	0.73	0.73	0.57	0.66	0.59
XLM-RoBERTa+Text-mixed	**0.94**	**0.93**	**0.93**	**0.93**	**0.82**	**0.86**	**0.80**

Table 2: Performance on Dev dataset (for both tracks). **Boldface** figures mean the best results.

Team	MLEC		MCEC	
	Mac-F_1	Rank	Mac-F_1	Rank
YNU-HPCC†	0.9869	1	0.9329	1
CTcloud	0.9833	2	0.8917	2
wsl&zt	0.9464	3	0.7359	3
Baseline‡	0.8347	-	0.7014	-

Table 3: Official results from the shared task leader board. Team† and Team‡ present our team name and the official baseline, respectively.

4.1 Datasets

The Internet is the most prominent source in promoting global, linguistic code-mixed culture. In South Asian community and particularly in Pakistan, code-mixed (English and Roman Urdu) text became a preferable script for Facebook comments/posts, tweets, and daily communication using SMS messages. The shared task organizers made available the dataset from (Ameer et al., 2022). Table 1 reported the detailed datasets in statistics.

4.2 Evaluation Metrics

To evaluate the performance of participant systems, the official competition provides Micro $\mathbf{F}_1$ (Mic-$\mathbf{F}_1$), Macro $\mathbf{F}_1$ (Mac-$\mathbf{F}_1$), and Accuracy (Acc) for track 1 and Mac-$\mathbf{F}_1$, Mac-Precision (Mac-P), Mac-Recall (Mac-R), and Acc for track 2.

4.3 Implementation Details

Hyper-parameters. All sentence is tokenized by XLM-RoBERTa based tokenizer with a maximum length of 90. Sentence encoder is `twitter-xlm-roberta-base-sentiment` PLM with the dimensionality of 768. α in Beta distribution is set as 0.75. Dropout ratio in MLP module is set as 0.2. For optimization, AdmW is adopted with learning rate of 2e-5 and batch size of 64. The code of this paper is availabled at `https://github.com/linsongisgood/wassa2023`.

Baselines. To investigate the effect of the proposed method, several baseline models are introduced, including BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) and their variants in cross-lingual versions.

4.4 Results and Analysis

Comparative Dev results were reported in Table 2. Due to PLMs that transfer generic language performance learned from a large pretrained corpus into downstream tasks, several PLMs achieved competitive results on both tracks. It can be observed that, with the cross-linguistic pretraining phase, PLMs such as mBERT (Pires et al., 2019) and XLM-RoBERTa (Conneau et al., 2019) outperformed monolingual PLMs such as BERT and XLM-RoBERTa, respectively. This phenomenon demonstrated that aligning cross-linguistic word representation and semantics is crucial to generate robust representation on code-mixed texts.

Furthermore, we found that the introduction of the text-mixed data augmentation method gained more performance on both tracks. A possible reason may be that the combination of multi-class and multi-label corpora improved the generalization capability of the system. Note that the proposed text-mixed method facilitates the shared system simultaneously performing both tracks during one training phase. Table 3 showed our best submissions with official results and ranks, revealing the effectiveness of the proposed system.

5 Conclusions

In this paper, we proposed described our system submission WASSA 2023 shared task 2 in emotion classification. Our system utilizes the text-mixed method and cross-lingual PLMs for robust representation of code-mixed texts. As a result, our system won the first rank in both tracks.

In the future, we will explore the text-mixed method applied to large amounts of unlabeled code-mixed texts for better performance.

Limitations

The limitations of this work can be concluded into three points: 1) The data in the test set is relatively small, so it cannot more accurately reflect the effectiveness of the method proposed in this paper. We believe that tuning the model on a larger dataset can help improve the performance of the model. 2) Due to device performance limitations,we did not experiment with larger models. In our experiment,we only tested the method with models like XLM-RoBERTa, mBERT and BERT. Its performance with larger models is not known. 3) We did not perform an extensive hyperparameter search, which might further improve the model's performance.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant Nos. 61966038 and 62266051, and the Yunnan Postdoctoral Science Foundation under Grant Nos. C615300504048. The authors would like to thank the anonymous reviewers for their constructive comments.

References

Khaled Ahmed, Neamat El Tazi, and Ahmad Hany Hossny. 2015. Sentiment analysis over social networks: an overview. In *2015 IEEE international conference on systems, man, and cybernetics*, pages 2174–2179.

Iqra Ameer, Grigori Sidorov, Helena Gomez-Adorno, and Rao Muhammad Adeel Nawab. 2022. Multi-Label Emotion Classification on Code-Mixed Text: Data and Methods. *IEEE Access*, 10:8779–8789.

Francesco Barbieri, Luis Espinosa Anke, and Jose Camacho-Collados. 2022. XLM-T: Multilingual Language Models in Twitter for Sentiment Analysis and Beyond. *2022 Language Resources and Evaluation Conference, LREC 2022*, pages 258–266.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova Google, and AI Language. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

Shinto Eguchi and John Copas. 2006. Interpreting kullback–leibler divergence with the neyman–pearson lemma. *Journal of Multivariate Analysis*, 97(9):2034–2040.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2017. Imagenet classification with deep convolutional neural networks. *Commun. ACM*, 60(6):84–90.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, Radu Soricut, and Google Research. 2019. ALBERT: A LITE BERT FOR SELF-SUPERVISED LEARNING OF LANGUAGE REPRESENTATIONS.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach.

Andrew Maxwell, Runzhi Li, Bei Yang, Heng Weng, Aihua Ou, Huixiao Hong, Zhaoxian Zhou, Ping Gong, and Chaoyang Zhang. 2017. Deep learning architectures for multi-label classification of intelligent health risk prediction. *BMC bioinformatics*, 18:121–131.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Interspeech*, volume 2, pages 1045–1048.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert? *arXiv preprint arXiv:1906.01502*.

Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897.

K Santosh, Romil Bansal, Mihir Shekhar, and Vasudeva Varma. 2013. Author profiling: Predicting age and gender from blogs. *Notebook for PAN at CLEF*, 2013(2).

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. 2018. Manifold Mixup: Better Representations by Interpolating Hidden States. *36th International Conference on Machine Learning, ICML 2019*, 2019:11196–11205.

Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Seong Joon Oh, Junsuk Choe, and Youngjoon Yoo. 2019. CutMix: Regularization strategy to train strong classifiers with localizable features. *Proceedings of the IEEE International Conference on Computer Vision*, 2019:6022–6031.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Generative Pretrained Transformers for Emotion Detection in a Code-Switching Setting

Andrew Nedilko
Workhuman
agnedil@gmail.com

Abstract

This paper describes the approach that we utilized to participate in the shared task for multi-label and multi-class emotion classification organized as part of WASSA 2023 at ACL 2023. The objective was to build models that can predict 11 classes of emotions, or the lack thereof (neutral class) based on code-mixed Roman Urdu and English SMS text messages. We participated in Track 2 of this task - multi-class emotion classification (MCEC). We used generative pretrained transformers, namely ChatGPT because it has a commercially available full-scale API, for the emotion detection task by leveraging the prompt engineering and zero-shot / few-shot learning methodologies based on multiple experiments on the dev set. Although this was the first time we used a GPT model for the purpose, this approach allowed us to beat our own baseline character-based XGBClassifier, as well as the baseline model trained by the organizers (bert-base-multilingual-cased). We ranked 4th and achieved the macro F1 score of 0.7038 and the accuracy of 0.7313 on the blind test set.

1 Introduction: Emotion Detection

Emotion detection using machine learning (ML) models presents significant challenges due to several factors. Firstly, emotions are subjective - the way they are expressed can vary greatly between different people and cultures.

Secondly, emotions can depend on the context. The topic and the participants of the conversation, the cultural and social context can affect the way emotions are understood. But models that interpret the context are quite a challenging problem in ML.

Thirdly, the language to express emotions can be complex and varied. There are few standards for labeling emotions in text. This variability poses difficulties in developing effective ML models for emotion detection.

Finally, high-quality labeled data for creating ML models in emotion detection is scarce. The labeling process is time-consuming and subjective, making it not easy to receive large volumes of reliable training data. This further complicates the development of effective ML models for emotion detection and is even more true for such low-resource languages as Urdu and for such subjective domains as highly colloquial SMS text messages.

Due to broad usage of artificial intelligence (AI) systems, e.g. automated phone systems, online chatbots as the first step of customer support systems, emotion detection is becoming more important. But it is still a challenging task for a machine because even humans can disagree about emotion interpretation. This makes it harder to transfer the human knowledge to machines.

2 Related Work

People often find it difficult to fathom all the subtleties associated with emotions, and therefore historically there has been only limited research done for the purpose of emotion detection and classification. But recently, emotion detection and empathy prediction have really gained popularity in various shared tasks and challenges. However, we must acknowledge that the majority of emotion research has been done in a monolingual setting. Such methods can hardly be efficient for code switching which is common in social media when two different languages can be used interchangeably in the same message.

As described by Ameer et al. (2022) and Ilyas et al. (2023) - Urdu is a South Asian language spoken by over 300 mln. speakers. Traditionally, it is written in Perso-Arabic script, but Roman Urdu (RU) is more popular for informal settings, for example on social media platforms like Facebook, Twitter, YouTube etc. RU is the language of the Internet.

Ameer et al. (2022) presented a large corpus for the multi-label emotion classification: 11,914 code-mixed SMS messages (Roman Urdu - English) -

Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis, pages 616–620
July 14, 2023 ©2023 Association for Computational Linguistics

which serves as the dataset for the current shared task. Every message was annotated manually for the presence of 12 emotions: anger, anticipation, disgust, fear, joy, love, optimism, pessimism, sadness, surprise, trust, and neutral (no emotion). The best performing ML methods were classical machine learning models trained on word unigrams with a combination of OVR multi-label and SVC single-label ML algorithms.

Ilyas et al. (2023) generated a new code-mixed Urdu - English emotion corpus from multiple sources. They developed and applied their own Annotation Guidelines at two levels: neutral vs. emotional sentences and the choice of correct emotion labels. A CNN architecture with Glove embeddings outperformed all other ML models at the first level, while RandomForest was the best model for the second level.

Akhter et al. (2020) proposed the first offensive Urdu dataset with user-generated content from social media. Their ML experiments showed that character-based trigrams outperformed other word and character n-grams with such ML models as LogitBoost and SimpleLogistic.

As a general approach, Wang et al. (2015) utilized a term-document bipartite graph to capture the bilingual and sentimental code-switching information and proposed a label propagation-based approach to learn and predict in such a graph.

3 Dataset and Task

This work is a result of our participation in **Track MCEC: Multi-Class Emotion Classification**. Given a code-mixed SMS message, the task was to classify it into one of the above 12 categories. The dataset for this task was proposed by Ameer et al. (2022).

The training set consists of approx. 9.5k examples (see Figure 1), while there are approx. 1190 examples in each of the development and test sets (Figure 2 and Figure 3, respectively). To get a more reasonable level of bias variance trade-off in ML models, the dataset had to be deduplicated. The training set contained 2k duplicates, the dev set and test set – 40 duplicates each. Besides, 0.5k data points from the training set leaked into the development set and about the same number [of somewhat different training data points] leaked into the test set (approx. half the size of each subset). The dev and test sets could not be deduplicated or reduced in size in order to be able to report the correct metrics. Therefore, the leaked data points were removed from the training set.

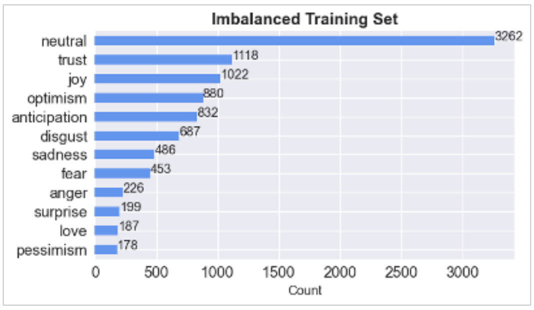


Figure 1: Distribution of labels in the training set.

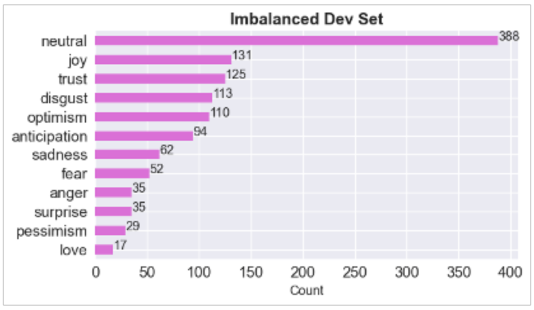


Figure 2: Distribution of labels in the dev set.

After deduplication and removal of the overlap with the dev set and test set, the training set became approx. two thirds of its original size (6.1k examples). All three subsets are imbalanced in a somewhat similar way with slight variations; the majority category is neutral, and the least represented ones are pessimism and love.

4 System Description

4.1 Baseline Models

Our baseline model was XGBClassifier with character ngram counts as features. The ngram range was (1,5). We did limited hyperparameter tuning and cross-validation (the main focus of this study was a different model). The initial macro F1 score

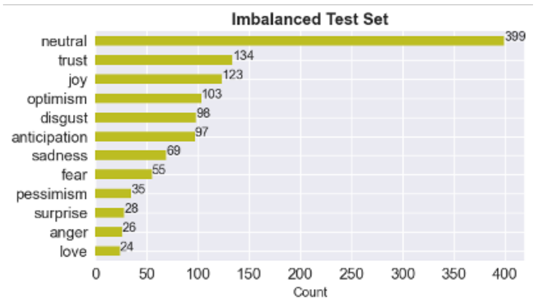


Figure 3: Distribution of labels in the test set.

was below 0.6, while the best one was 0.68 with the accuracy of 0.705.

4.2 GPT and Iterative Prompt Engineering

Large transformer models and their ensembles (Desai et al., 2022) demonstrated good results on sequence classification, but it is still hard to achieve a high macro F1 score when detecting emotions.

Given the impressive recent advances in autoregressive models with the generative pretrained transformer (GPT) architecture, we decided to use GPT models with the zero-shot or few-shot approaches to capture the human emotions in a more precise manner.

We used ChatGPT for this purpose because it has a commercially available full-scale API, and it was trained on multiple languages and understands them inherently under the hood. Several dozens of prompts were engineered to run full-scale experiments on the dev set. The training set was used only for few-shot learning.

First, ChatGPT was asked to translate texts into English and then do the classification. But the final winning prompt did not mention the translation and asked the model to do the classification directly. It was a few-shot approach where, for each data point in the dev set, we selected the top 100 most similar texts from the training set as examples. We used the cosine similarity on OpenAI embeddings as the similarity measure between dev set and training set texts.

This was the first time we used ChatGPT for emotion detection on a code-mixed dataset with such a colloquial vocabulary as SMS text messages. Nevertheless, it outperformed our baseline classifier and the baseline model trained by the organizers of the shared task (bert-base-multilingual-cased). We ranked 4th and achieved the macro F1 score of 0.7038 and the accuracy of 0.7313 on the final test set.

5 Analysis of Results

The initial idea was that LLMs like ChatGPT are perfect for the task at hand because they are known for being able to translate from one language to another and they can do zero-shot text classification, especially with such self-explanatory labels as joy, fear, anger, surprise, etc.

Our initial approach was a zero-shot multiclass emotion classification with a prompt asking the LLM to translate and classify each given text from the dev set into one of the emotion categories. We then calculated the macro F1 score as if it was a usual ML classification task. Extracting emotion categories from the LLM's responses was not always straightforward, as the LLM tended to include extraneous words. We tried to iteratively improve the prompts to achieve better F1 scores.

The quality of the response from ChatGPT greatly depends on the prompt. For example, the first step was to do the translation, and when we asked ChatGPT to translate code-mixed texts into English, it did a good job silently. But when we decided to improve the results and asked ChatGPT: "Act as a smart Roman Urdu to English translator and do your best to translate the text below into English", unexpectedly the LLM started to complain saying that the quality of the Urdu text was not very good, and that it couldn't translate the text well. Since we had to batch process about a thousand of such responses in the dev set, filtering the cases when the English translation was provided vs. the cases when there was a complaint was not a trivial task.

The macro F1 score for the initial zero-shot classification results was below 0.5. This could be due to the fact that there were 12 labels and some of the emotions were really hard to extract from the available text without the knowledge of the full context which was not always provided explicitly.

Next, we tried a zero-shot binary text classification (neutral vs. emotional texts). ChatGPT was biased towards the neutral category, but when we tried to use this prompt: "Act as a careful and accurate zero-shot text classifier and classify the text neutral or emotional . . . ", the LLM started to complain again saying it couldn't produce an accurate classification because of the colloquial nature of the text or for other reasons. It seemed like ChatGPT had a problem of excessive perfectionism when one asked it to be very good at what it does!

The binary classifier's macro F1 score was around 0.5. Therefore, next we decided to do the few-shot multiclass classification. ChatGPT has a limited context window size of 4096 tokens while the total number of tokens in the training set is over 110k. At first, we randomly split the training set into chunks of 100 texts each because this allowed to keep the total number of tokens in the concatenated examples under 4k. We tried to keep the proportional distribution of labels in each chunk in line with the overall distribution. Then, we iteratively used these chunks with all the data points from the dev set: one chunk of examples per one dev set data point.

Afterwards, we tried a smarter approach which allowed us to get our best macro F1 score on the test set of 0.7038 and the accuracy of 0.7313. First, we embedded every SMS text message using the OpenAI embeddings. Then, using the cosine similarity, for each example in the dev set (or in the test set for the final submission) we selected 100 most similar examples from the training set and used those as few-shot examples to teach ChatGPT about existing emotion labels. See Table 1 below for a brief summary of results.

Classifier	**Macro F1**	**Accuracy**
Baseline XGBClassifier	0.68	0.705
Baseline BERT-base	0.7014	0.7298
Best few-shot ChatGPT system (most similar examples)	0.7038	0.7313

Table 1: Summary of Emotion Classification Results

Other approaches included sending the second prompt asking the LLM if it was sure of its previous response. There were quite a few cases when ChatGPT changed its response which helped with the zero-shot classification, but not with the few-shot effort.

We also tried concatenating all examples and their categories into one huge prompt or having each individual example in a single prompt using the chat-like framework offered by the ChatGPT API. This did not have any significant impact on the results.

6 Conclusions

Overall, we should say that this task is quite challenging because:

- the data consists of code-mixed text messages - two languages mixed together;
- Urdu is a low-resource language; during training ChatGPT saw significantly less Urdu than English - it must be considerably less efficient at Urdu;
- the emotion detection task is challenging for machines in general;
- there are 12 emotion categories which is a lot;
- even for a human, the SMS text messages often don't have enough context to understand the type of emotion.

Nevertheless, our approach helped us beat both baseline models. Based on a subjective assessment, the time spent for prompt engineering, as opposed to the wait time when the code was just running, was less than the time we would normally spend on ML model selection, hyperparameter tuning, and cross-validation to avoid overfitting when dealing with classical and deep learning ML models.

Based on our limited experiments in the web UI, we anticipate that using GPT-4 will help achieve even better results once the GPT-4 API becomes available to the general public. It was not available at the time we ran our experiments for this task.

Limitations

In addition to the data-based limitations listed in Section 6, we faced the following technology-based limitations:

- The results are greatly dependent on the prompt design. It is not uncommon to spend a lot of efforts on coming up with the right prompt. Each use case may need a different prompt.
- Overall, ChatGPT provides a stable output especially if one asks for a specific output format. But there is still an element of volatility when one or few responses contain extraneous text, or the categories are outside of the predefined list. This is due to the conversational nature of the model, and such cases had to be processed as exceptions.
- ChatGPT "remembers" the past conversations, but this memory is limited to the context window size which is only 4096 tokens. This makes it challenging to work with large datasets which have to be split into pieces to be processed independently.
- One has to remember that one must pay for the use of the ChatGPT API - very long prompts used multiple times or too many examples for few-shot learning may be discouraged for cost savings purposes.

References

Muhammad Pervez Akhter, Zheng Jiangbin, Irfan Raza Naqvi, Mohammed Abdelmajeed, and Muhammad Tariq Sadiq. 2020. Automatic detection of offensive language for urdu and roman urdu. *IEEE Access*, 8:91213–91226.

Iqra Ameer, Grigori Sidorov, Helena Gomez-Adorno, and Rao Muhammad Adeel Nawab. 2022. Multi-label emotion classification on code-mixed text: Data and methods. *IEEE Access*, 10:8779–8789.

Shaily Desai, Atharva Kshirsagar, Aditi Sidnerlikar, Nikhil Khodake, and Manisha Marathe. 2022. Leveraging emotion-specific features to improve transformer performance for emotion classification. In *Proceedings of the 12th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis*, pages 245–249, Dublin, Ireland. Association for Computational Linguistics.

Abdullah Ilyas, Khurram Shahzad, and Muhammad Kamran Malik. 2023. Emotion detection in code-mixed roman urdu - english text. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 22(2).

Zhongqing Wang, Sophia Lee, Shoushan Li, and Guodong Zhou. 2015. Emotion detection in code-switching texts via bilingual and sentimental information. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 763–768, Beijing, China. Association for Computational Linguistics.

Author Index

Abels, Christoph Maximilian, 479
Adams, Kiki Rose, 467
Agrawal, Ameeta, 58
Al Bataineh, Ali, 587
Al Osman, Hussein, 172
Alacam, Ozge, 278
Alcock, Keith, 416
Alexeeva, Maria, 416
Ameer, Iqra, 587
Ananiadou, Sophia, 1
Anna John, Neha, 19
Anni, Isaac Kobby, 416
Anubhai, Rishita, 19

Ballesteros, Miguel, 19
Bao, Siyu, 596
Barnes, Jeremy, 290
Barriere, Valentin, 511
Barua, Nabarun, 542
Belz, Anya, 341
Benajiba, Yassine, 19
Benedetto, Irene, 354
Bernhardt, Andy, 441
Bethard, Steven, 194
Bhadane, Dnyaneshwar, 497
Bhattacharjee, Amrita, 110
Bhaumik, Ankita, 441
Bizzoni, Yuri, 11
Bohacek, Matyas, 96
Bravansky, Michal, 96
Bölücü, Necva, 587

Cagliero, Luca, 354
Cao, Lina, 596
Carvalho, Joao Paulo, 228
Chandrashekhar, Suraj, 531
Chang, Jung-Ying, 548
Chavan, Tanmay, 564
Chu, Yi, 569
Cohen, Allegra A. Beal, 416

Daelemans, Walter, 237, 251
De Bruyne, Luna, 458
De Lhoneux, Miryam, 365
Deshpande, Kshitij, 564
Dharmadhikari, Shweta, 558
Dias, Isabel, 228
Doyle, Gabriel, 452

Edwards, Matthew, 28

Fan, Shunguo, 596
Fayek, Haytham M., 149
Flek, Lucie, 553
Fyshe, Alona, 401

Gedeon, Tom, 536
Ghanadian, Hamideh, 172
Giorgi, Salvatore, 511
Gokani, Monil, 129
Gokhale, Omkar, 558
Gonzalez-Lopez, Samuel, 194
Grimm, Anna, 138
Gruschka, Fabio, 553
Guntuku, Sharath Chandra, 202
Guo, Lu, 379
Gupta, Akshat, 184

Halder, Kishaloy, 19
Han, Ji-Eun, 162
Hasan, Md Rakibul, 536
Havaldar, Shreya, 202
Hoeken, Sanne, 278
Hossain, Md Zakir, 536
Hoste, Veronique, 315
Hua, Yilun, 110
Huang, Nannan, 149
Hussiny, Mohammad Ali, 271
Hyland, Caroline, 416

Ireland, Molly E., 467
Iserman, Micah, 467

Jentzsch, Sophie, 325

Kadiyala, Ram Mohan Rao, 531
Kang, Pilsung, 215
Kanojia, Diptesh, 497
Kanyamahanga, Hubert, 416
Katsios, Gregorios A, 441
Kersting, Kristian, 325
Kho, Yookyung, 215
Kim, Doyoon, 215
Kim, Jaehee, 215
Kim, Younsun, 215
Kim, Yuheun, 379

Klinger, Roman, 39
Kodali, Prashant, 601
Kolhatkar, Gaurav, 558
Kroll, Henning, 138
Kumar, Vivek, 581
Kumarage, Tharindu Sandaruwan, 110
Kumaraguru, Ponnurangam, 601

La Quatra, Moreno, 354
Lahnala, Allison, 553
Lal, Yash Kumar, 390
Lee, Lung-Hao, 548
Lee, Yukyung, 215
Lefever, Els, 315
Lemmens, Jens, 237
Lent, Heather, 365
Lewandowsky, Stephan, 28, 479
Li, Yingya, 379
Li, Zhuojun, 574
Lin, Tzu-Mi, 548
Liu, Huan, 110
Liu, Langchen, 202
Liu, Xiaomo, 184
Lorandi, Michela, 341
Lu, Xin, 574

Maladry, Aaron, 315
Mamidi, Radhika, 129
McKeown, Kathleen, 110
Mieskes, Margot, 71
Min, Erxue, 1
Mohammad, Saif M., 401
Moniz, Helena Silva, 228
Moravec, Vaclav, 96
Moreira, Pascale Feldkamp, 11
Moscoso Del Prado, Fermin, 305
Mousavi, Seyed Mahed, 85
Muresan, Smaranda, 19
Murthy, Rudra, 497

Nafis, Nazia, 497
Nedilko, Andrew, 569, 616
Nejadgholi, Isar, 172
Nielbo, Kristoffer, 11
Nilsson, August Håkan, 390

O'Boyle, Brianna, 452
Øvrelid, Lilja, 271

Pahwa, Bhavish, 606
Pal, Santanu, 542
Paranjape, Aditya, 558
Patkar, Aditya, 531
Patwardhan, Yash, 558
Pereira, Patrícia, 228
Probol, Nadine, 71

Qin, Bing, 574

Rahman, Shafin, 536
Rai, Sunny, 202
Rajendran, Udhaya Kumar, 431
Ran, Xuqiao, 611
Rawat, Akash, 497
Riccardi, Giuseppe, 85
Roccabruna, Gabriel, 85
Roth, Dan, 19

Sa, Ning, 441
Savinova, Elena, 305
Sazzed, Salim, 52
Schaefer, Robin, 479
Schwartz, H., 390
Sedoc, João, 511
Shah, Sameena, 184
Shrivastava, Manish, 601
Simchon, Almog, 28
Singh, Sushmita, 581
Singhal, Bhumika, 202
Sohn, Kyung-Ah, 162
Sonawane, Sheetal, 564
Soon, Susannah, 536
Spolwind, Calvin, 138
Srinivas, Addepalli Sai, 542
Stede, Manfred, 479
Strzalkowski, Tomek, 441
Subbiah, Melanie, 110
Surdeanu, Mihai, 416
Sutton, Adam, 28

Tafreshi, Shabnam, 511
Tatariya, Kushal, 365
Teodorescu, Daniela, 401
Thomsen, Mads Rosendahl, 11
Tian, Lin, 149
Tiwari, Prayag, 581
Tong, Yanpeng, 574
Trabelsi, Amine, 431
Trevisan, Martino, 354
Trhlík, Filip, 96

Ungar, Lyle, 202

V Ganesan, Adithya, 390
Vacareanu, Robert, 19
Van Hee, Cynthia, 315
Van Nooten, Jens, 251
Varia, Siddharth, 19
Vassio, Luca, 354
Vedula, Bhaskara Hanuma, 601
Velutharambath, Aswathy, 39

Wang, Jin, 526, 611
Wang, Shuai, 19
Wang, Yukun, 526
Welch, Charles, 553

Xie, Justin J, 58
Xu, Dan, 611
Xu, Hua, 587

Yang, Dongming, 596
Yu, Bei, 379

Zarrieß, Sina, 278
Zhang, Jinghui, 596
Zhang, Xiuzhen, 149
Zhang, Xuejie, 526, 611
Zhang, You, 611
Zhao, Yanyan, 574
Zielinski, Andrea, 138