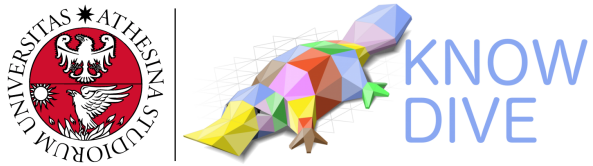
ICNLSP 2021

Proceedings of the 4th International Conference on Natural Language and Speech Processing

12–13 November, 2021 (virtual)




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 978-1-955917-18-6

https://www.icnlsp.org/

Introduction

Welcome to the fourth International Conference on Natural Language and Speech Processing (ICNLSP 2021), held online on November 12th, 13th 2021. ICNLSP is an opportunity and a forum for researchers, students, and industrials to exchange ideas and discuss research and trends in the field of Natural Language Processing. Indeed, many topics were discussed through the interesting works presented during the two days of the conference: speech recognition, machine translation, text summarization, sentiment analysis, natural language understanding, language resources, etc.

The accepted papers are of good quality thanks to the high-quality level of the reviews done by the program committee members who decided to accept 35 papers (long and short ones).

ICNLSP 2021, by including the second NSURL workshop, aims to draw the attention of researchers to provide solutions and resources for under resourced languages, by organizing shared tasks/ competitions for solving NLP problems. This year, the task was on Semantic Relation Extraction in Persian which attracted a number of contributions, 6 of them were accepted and presented in the workshop on November 14th, 2021.

We had the honor of having high-standard speakers with us, who gave valuable talks, starting by Dr. Ahmed Abdelali -QCRI- who presented his talk about *understanding Arabic transformer models*. The second keynote entitled *Figurative Language Analysis* was given by PD Dr. Valia Kordoni -Humboldt University- followed by Dr. Hussein Al-Natsheh -Beyond limits- who gave interesting thoughts on *AI technology commercialization and how to move from research to product innovation*. The last talk was presented by Dr. Kareem Darwish -Aixplain- on one of the challenged topics which is Arabic Diacritic Recovery under the title *Bring All Your Features: Arabic Diacritic Recovery Using a Feature-Rich Recurrent Neural Model*.

We would like to acknowledge the support provided by University of Trento, and KnowDive group (University of Trento), and Datascientia (University of Trento). We would like also to express our gratitude to the organizing and the program committees for the hard and valuable contributions.

Mourad Abbas and Abed Alhakim Freihat

Organizers:

General Chair: Dr. Mourad Abbas
Chair: Dr. Abed Alhakim Freihat

Program Chair: Dr. Abed Alhakim Freihat

Publicity Chair: Dr. Mohamed Lichouri

Program Committee:

Mourad Abbas, HCLA, Algeria
Ahmed Abdelali, QCRI, Qatar
Mohamed Afify, Microsoft, Egypt
Messaoud Bengherabi, CDTA, Algeria
Djamel Bouchaffra, CDTA, Algeria
Fayssal Bouarourou, University of Strasbourg, France
Markus Brückl, TU Berlin, Germany
Hadda Cherroun, Amar Telidji University, Algeria
Gérard Chollet, CNRS, France
Kareem Darwish, QCRI, Qatar
Najim Dehak, Johns Hopkins University, USA
Mohamed Elfeky, Google Inc., USA
Ashraf Elnagar, University of Sharjah, UAE
Abed Alhakim Freihat, University of Trento, Italy
Nada Ghneim, Syrian Virtual University, Syria
Neil Glackin, Intelligent Voice, UK
Ahmed Guessoum, USTHB, Algeria
Mahmoud Gzawi, university of Lyon 2, France
Valia Kordoni, Humboldt University, Germany
Tomi Kinnunen, University of Eastern Finland, Finland
Eric Laporte, UPEM, France
Shang-Wen Li, Facebook AI., USA
Georges Linarès, University of Avignon, France
Shervin Malmasi, Harvard University, USA
Lluis Marquez, Amazon, Spain
Mhamed Mataoui, EMP, Algeria
Mohammed Mediani, University of Adrar, Algeria
Fatiha Merazka, USTHB, Algeria
Hamdy Mubarak, QCRI, Qatar
Preslav Nakov, QCRI, Qatar
Alexis Neme, UPEM, France
Axel Roebel, IRCAM, France
Younes Samih, Universität Düsseldorf, Germany

Hassan Satori, Sidi Mohammed Ben Abbdallah University, Morocco
Tim Schlippe, Silicon Surfer, Germany
Khaled Shaalan, The British University in Dubai, UAE
Otakar Smrz, Džám-e Džam Language Institute, Czech Republic
Rudolph Sock, University of Strasbourg, France
Irina Temnikova, QCRI, Qatar
Jan Trmal, Johns Hopkins University, USA
Stephan Vogel, QCRI, Qatar
Fayçal Ykhlef, CDTA, Algeria
Hasna Zaouali, University of Strasbourg, France

Additional Reviewers:

Hadi Khalilia, University of Trento, Italy
Mohamed Lichouri, USTHB, Algeria
Khaled Lounnas, USTHB, Algeria
Attia Nehar, University of Ziane Achour, Algeria
Slimane Bellaouar, University of Ghardaia, Algeria.

Organizing Committee:

Hadi Khalilia, University of Trento
Khaled Lounnas, USTHB, Algeria
Nandu C Nair, University of Trento

Invited Speakers:

Dr. Ahmed Abdelali, QCRI, Qatar
PD Dr. Valia Kordoni, Humboldt-Universität zu Berlin, Germany
Dr. Hussein Al-Natsheh, Beyond Limits.
Dr. Kareem Darwish, AiXplain.

Invited Talks

Understanding Arabic Transformer Models

Dr. Ahmed Abdelali

The success of pre-trained transformer models trained on Arabic and its dialects have gained more attention in the last few years . They were able to set and achieve new state of the art performance and accuracy in numerous downstream NLP tasks. Despite such popularity, no evaluation to compare the internal representations has been conducted. In this work we present deep comparison for these pre-trained Arabic models beyond the data used for the training or detailed architecture. We present an in-depth analysis for the layers and neurons for these models. The evaluation is done using three intrinsic tasks: two morphological tagging tasks based on MSA (modern standard Arabic) and dialectal Arabic and a dialectal identification task.

Figurative Language Analysis

PD Dr. Valia Kordoni

This talk focuses on figurative language analysis in multi-genre data. While metaphor has been tackled in Natural Language Processing before, the focus has never simultaneously been on the analysis of multi-genre and heterogeneous texts.

AI Technology Commercialization: From Research to Product Innovation

Dr. Hussein Al-Natsheh

The number of researchers in the NLP research community, and the AI research at large, is increasing as well as the funding from both the public and private sectors. However, not enough of these invented technologies are applied in solving real-life problems. In this keynote, we will shed the light on this challenge and how we can turn it into an opportunity that can motivate investing in more research both applied and scientific. This topic touches many areas that we will present and link to in the session including open innovation, open-source, open data, technology licensing, product innovation, marketing and pricing models, investment, team building, and MLOps. We will also provide some examples where we have successfully turned research-level technology into successful and scalable products.

Bring All Your Features: Arabic Diacritic Recovery Using a Feature-Rich Recurrent Neural Model

Dr. Kareem Darwish

Diacritics (short vowels) are typically omitted when writing Arabic text, and readers have to reintroduce them to correctly pronounce words. There are two types of Arabic diacritics: the first are core-word diacritics (CW), which specify the lexical selection, and the second are case endings (CE), which typically appear at the end of word stems and generally specify their syntactic roles. Recovering CEs is significantly harder than recovering core-word diacritics due to inter-word dependencies, which are often distant. The presentation shows the use of a feature-rich recurrent neural network model that uses a variety of linguistic and surface-level features to recover both core word diacritics and case endings. The model surpasses all previous state-of-the-art systems with a CW error rate of 2.86% and a CE error rate (CEER) of 3.7%, which is 61% lower than any state-of-the-art system. When combining diacritized word cores with case endings, the resultant word error rate is 6.0%. This highlights the effectiveness of feature engineering for such deep neural models.

Table of Contents

End-to-End Annotator Bias Approximation on Crowdsourced Single-Label Sentiment Analysis . . 1
Gerhard Hagerer, David Szabo, Andreas Koch, Maria Luisa Ripoll Dominguez, Christian Widmer, Maximilian Wich, Hannah Danner and Georg Groh

Speech Technology for Everyone: Automatic Speech Recognition for Non-Native English 11
Toshiko Shibano, Xinyi Zhang, Mia Taige Li, Haejin Cho, Peter Sullivan and Muhammad Abdul-Mageed

Orthographic Transliteration for Kabyle Speech Recognition 21
Christopher Haberland and Ni Lao

Machine Translation for Zero and Low-resourced Dialects using a New Extended Version of the Dialectal Parallel Corpus (Padic v2.0) 33
Mohamed Lichouri and Mourad Abbas

Improving BERT Performance for Aspect-Based Sentiment Analysis 39
Akbar Karimi, Leonardo Rossi and Andrea Prati

MAPLE –MAsking words to generate blackout Poetry using sequence-to-sequence LEarning . . . 47
Aditeya Baral, Himanshu Jain, Deeksha D and Dr. Mamatha H R

Beyond Voice Activity Detection: Hybrid Audio Segmentation for Direct Speech Translation . . . 55
Marco Gaido, Matteo Negri, Mauro Cettolo and Marco Turchi

A Sample-Based Training Method for Distantly Supervised Relation Extraction with Pre-Trained Transformers 63
Mehrdad Nasser, Mohamad Bagher Sajadi and Behrouz Minaei-Bidgoli

Static Fuzzy Bag-of-Words: a Lightweight and Fast Sentence Embedding Algorithm 73
Matteo Muffo, Roberto Tedesco, Licia Sbattella and Vincenzo Scotti

ITAcotron 2: Transfering English Speech Synthesis Architectures and Speech Features to Italian . 83
Anna Favaro, Licia Sbattella, Roberto Tedesco and Vincenzo Scotti

Supporting Undotted Arabic with Pre-trained Language Models 89
Aviad Rom and Kfir Bar

Identifying and Understanding Game-Framing in Online News: BERT and Fine-Grained Linguistic Features 95
Hayastan Avetisyan and David Broneske

Formulating Automated Responses to Cognitive Distortions for CBT Interactions 108
Ignacio de Toledo Rodriguez, Giancarlo Salton and Robert Ross

The Quality of Lexical Semantic Resources: A Survey 117
Hadi Khalilia, Abed Alhakim Freihat and Fausto Giunchiglia

An interpretable person-job fitting approach based on classification and ranking 130
Mohamed Amine Menacer, Fatma Ben Hamda, Ghada Mighri, Sabeur Ben Hamidene and Maxime Cariou

Beam Search with Bidirectional Strategies for Neural Response Generation 139
Pierre Colombo, Chloé Clavel, Chouchang Yack and Giovanna Varni

A3C: Arabic Anaphora Annotated Corpus 147

Mohamed Amine Cheragui, Abdelhalim Hafedh Dahou and Mohamed Abdelmoazz

User Generated Content and Engagement Analysis in Social Media case of Algerian Brands . . . 156
Aicha Chorana and Hadda Cherroun

Automatic Assessment of Speaking Skills Using Aural and Textual Information . . . 166
Sofia Eleftheriou, Panagiotis Koromilas and Theodoros Giannakopoulos

A New Approach for Arabic Text Summarization . . . 176
Samira Lagrini and Mohammed Redjimi

Compressive Performers in Language Modelling . . . 186
Anjali Ragupathi, Siddharth Shanmuganathan and Manu Madhavan

Learning ULMFiT and Self-Distillation with Calibration for Medical Dialogue System . . . 196
Shuang Ao and Xeno Acharya

Automated Recognition of Hindi Word Audio Clips for Indian Children using Clustering-based Filters and Binary Classifier . . . 204
Anuj Gopal

BloomNet: A Robust Transformer based model for Bloom's Learning Outcome Classification . . . 209
Abdul Waheed, Muskan Goyal, Nimisha Mittal, Deepak Gupta, Ashish Khanna and Moolchand Sharma

Indic Languages Automatic Speech Recognition using Meta-Learning Approach . . . 219
Anugunj Naman and Kumari Deepshikha

TPT: An Empirical Term Selection for Arabic Text Categorization . . . 226
Mourad Abbas and Mohamed Lichouri

From local hesitations to global impressions of a speaker's feeling of knowing . . . 232
Tanvi Dinkar, Beatrice Biancardi and Chloé Clavel

The Task2Dial Dataset: A Novel Dataset for Commonsense-enhanced Task-based Dialogue Grounded in Documents . . . 242
Carl Strathearn and Dimitra Gkatzia

Domain and Task-Informed Sample Selection for Cross-Domain Target-based Sentiment Analysis 252
Kasturi Bhattacharjee, Rashmi Gangadharaiah and Smaranda Muresan

Audio-Visual Recipe Guidance for Smart Kitchen Devices . . . 257
Caroline Kendrick, Mariano Frohnmaier and Munir Georges

Arabic Named Entity Recognition Using Transformer-based-CRF Model . . . 262
Muhammad Saleh Al-Qurishi and Riad Souissi

The Articulatory and acoustics Effects of Pharyngeal Consonants on Adjacent Vowels in Arabic Language . . . 272
Fazia Karaoui, Amar Djeradi and Yves Laprie

A Comparative Study on Language Models for the Kannada Language . . . 280
Danish Mohammed Ebadulla, Rahul Raman, Hridhay Kiran Shetty and Mamatha H.R.

Using Bloom's Taxonomy to Classify Question Complexity . . . 285
Sabine Ullrich and Michaela Geierhos

Towards Phone Number Recognition For Code Switched Algerian Dialect . . . 290
Khaled Lounnas, Mourad Abbas and Mohamed Lichouri

End-to-End Annotator Bias Approximation on Crowdsourced Single-Label Sentiment Analysis

Gerhard Hagerer, **David Szabo**, **Andreas Koch**, **Maria Luisa Ripoll Dominguez**, **Christian Widmer**, **Maximilian Wich**, **Hannah Danner**, **Georg Groh**
Technical University of Munich, Germany
{ghagerer, grohg}@mytum.de

Abstract

Sentiment analysis is often a crowdsourcing task prone to subjective labels given by many annotators. It is not yet fully understood how the annotation bias of each annotator can be modeled correctly with state-of-the-art methods. However, resolving annotator bias precisely and reliably is the key to understand annotators' labeling behavior and to successfully resolve corresponding individual misconceptions and wrongdoings regarding the annotation task. Our contribution is an explanation and improvement for precise neural end-to-end bias modeling and ground truth estimation, which reduces an undesired mismatch in that regard of the existing state-of-the-art. Classification experiments show that it has potential to improve accuracy in cases where each sample is annotated only by one single annotator. We provide the whole source code publicly[1] and release an own domain-specific sentiment dataset containing 10,000 sentences discussing organic food products[2]. These are crawled from social media and are singly labeled by 10 non-expert annotators.

1 Introduction

Modeling annotator bias in conditions where each data point is annotated by multiple annotators, below referred to as multi-labeled crowdsourcing, has been investigated thoroughly. However, bias modeling when every data point is annotated by only one person, hereafter called singly labeled crowdsourcing, poses a rather specific and difficult challenge. It is in particular relevant for sentiment analysis, where singly labeled crowdsourced datasets are prevalent. This is due to data from the social web which is annotated by the data creators themselves, e.g., rating reviewers or categorizing image uploaders. This might further include multi-media contents such as audio, video, images, and other forms of texts. While the outlook for such forms of data is promising, end-to-end approaches have not yet been fully explored on these types of crowdsourcing applications.

With these benefits in mind, we propose a neural network model tailored for such data with singly labeled crowdsourced annotations. It computes a latent truth for each sample and the correct bias of every annotator while also considering input feature distribution during training. We modify the loss function such that *the annotator bias converges towards the actual confusion matrix of the regarding annotator and thus models the annotator biases correctly*. This is novel, as previous methods either require a multi-labeled crowdsourcing setting (Dawid and Skene, 1979; Hovy et al., 2013) or do not produce a correct annotator bias during training which would equal the confusion matrix, see Zeng et al. (2018, figure 5) and Rodrigues and Pereira (2018, figure 3). A correct annotator- or annotator-group bias, however, is necessary to derive correct conclusions about the respective annotator behavior. This is especially important for highly unreliable annotators who label a high number of samples randomly – a setting, in which our proposed approach maintains its correctness, too.

Our contributions are as follows. We describe the corresponding state-of-the-art for crowdsourcing algorithms and tasks in section 2. Our neural network model method for end-to-end crowdsourcing modeling is explained in section 3, which includes a mathematical explanation that our linear bias modeling approach yields the actual confusion matrices. The experiments in section 4 underline our proof, show that the model handles annotator bias correctly as opposed to previous models, and demonstrate how the approach impacts classification.

[1] https://github.com/theonlyandreas/end-to-end-crowdsourcing
[2] https://github.com/ghagerer/organic-dataset

2 Related Work

2.1 Crowdsourcing Algorithms

Problem definition. The need for data in the growing research areas of machine learning has given rise to the generalized use of crowdsourcing. This method of data collection increases the amount of data, saves time and money but comes at the potential cost of data quality. One of the key metrics of data quality is annotator reliability, which can be affected by various factors. For instance, the lack of rater accountability can entail spamming. *Spammers* are annotators that assign labels randomly and significantly reduce the quality of the data. Raykar and Yu (2012) and Hovy et al. (2013) addressed this issue by detecting spammers based on rater trustworthiness and the SpEM algorithm. However, spammers are not the only source of label inconsistencies. The varied personal backgrounds of crowd workers often lead to *annotator biases* that affect the overall accuracy of the models. Several works have previously ranked crowd workers (Hovy et al., 2013; Whitehill et al., 2009; Yan et al., 2010), clustered annotators (Peldszus and Stede, 2013), captured sources of bias (Wauthier and Jordan, 2011) or modeled the varying difficulty of the annotation tasks (Carpenter, 2008; Whitehill et al., 2009; Welinder et al., 2010) allowing for the elimination of unreliable labels and the improvement of the model predictions.

Ground truth estimation. One common challenge in crowdsourced datasets is the ground truth estimation. When an instance has been annotated multiple times, a simple yet effective technique is to implement majority voting or an extension thereof (TIAN and Zhu, 2015; Yan et al., 2010). More sophisticated methods focus on modeling label uncertainty (Spiegelhalter and Stovin, 1983) or implementing bias correction (Snow et al., 2008; Camilleri and Williams, 2020). These techniques are commonly used for NLP applications or computer vision tasks (Smyth et al., 1995; Camilleri and Williams, 2020). Most of these methods for inferring the ground truth labels use variations of the EM algorithm by Dawid and Skene (1979), which estimates annotator biases and latent labels in turns. We use its recent extension called the *Fast Dawid-Skene* algorithm (Sinha et al., 2018).

End-to-end approaches. The Dawid-Skene algorithm models the raters' *abilities* as respective bias matrices. Similar examples include GLAD (Whitehill et al., 2009) or MACE (Hovy et al., 2013), which infer true labels as well as labeler expertise and sample difficulty. These approaches infer the ground truth only from the labels and do not consider the input features. *End-to-end approaches* learn a latent truth, annotator information, and feature distribution jointly during actual model training (Zeng et al., 2018; Khetan et al., 2017; Rodrigues and Pereira, 2018). Some works use the EM algorithm (Raykar et al., 2009), e.g., to learn sample difficulties, annotator representations and ground truth estimates (Platanios et al., 2020). However, the EM algorithm has drawbacks, namely that it can be unstable and more expensive to train (Chu et al., 2020). LTNet models imperfect annotations derived from various image datasets using a single latent truth neural network and dataset-specific bias matrices (Zeng et al., 2018). A similar approach is used for crowdsourcing, representing annotator bias by confusion matrix estimates (Rodrigues and Pereira, 2018). Both approaches show a mismatch between the bias and how it is modeled, see Zeng et al. (2018, figure 5) and Rodrigues and Pereira (2018, figure 3). We adapt the LTNet architecture (see section 3), as it can be used to model crowd annotators on singly labeled sentiment analysis, which, to our knowledge, is not done yet in the context of annotator bias modeling. Recent works about noisy labeling in sentiment analysis do not consider annotator bias (Wang et al., 2019).

2.2 Crowdsourced Sentiment Datasets

Sentiment and Emotion. Many works use the terms *sentiment* and *emotion* interchangeably (Demszky et al., 2020; Kossaifi et al., 2021), whereas sentiment is directed towards an entity (Munezero et al., 2014) but emotion not necessarily. Both can be mapped to valence, which is the affective quality of goodness (high) or badness (low). Since emotion recognition often lacks annotated data, crowdsourced sentiment annotations can be beneficial (Snow et al., 2008).

Multi-Labeled Crowdsourced Datasets. Crowdsourced datasets, such as, Google GoEmotion (Demszky et al., 2020) and the SEWA database (Kossaifi et al., 2021), usually contain multiple labels per sample and require their aggregation using ground truth estimation. Multi-labeled datasets are preferable to singly labeled ones on limited data. Snow et al. (2008) proved that many non-expert annotators give a better performance than a few expert annotators and are cheaper in comparison.

Singly Labeled Crowdsourced Datasets. Singly labeled datasets are an option given a fixed budget and unlimited data. Khetan et al. (2017) showed that it is possible to model worker quality with single labels even when the annotations are made by non-experts. Thus, multiple annotations can not only be redundant but come at the expense of fewer labeled samples. For singly labeled data, it can be distinguished between reviewer annotators and external annotators. Reviewer annotators rate samples they created themselves. It is common in forums for product and opinion reviews where a review is accompanied by a rating. As an example of this, we utilized the TripAdvisor dataset (Thelwall, 2018). Further candidates are the Amazon review dataset (Ni et al., 2019), the Large Movie Review Dataset (Maas et al., 2011), and many more comprising sentiment. External annotators annotate samples they have not created. Experts are needed for complex annotation tasks requiring domain knowledge. These are not crowdsourced, since the number of annotators is small and fixed. More common are external non-experts. Snow et al. (2008) showed that multi-labeled datasets annotated by non-expert improve performance. Khetan et al. (2017) showed that it also performs well in the singly labeled case. Thus, datasets made of singly labeled non-expert annotations can be cheaper, faster, and obtain performances comparable to those comprised of different types of annotations. Our organic dataset is annotated accordingly, see section 4.3.

3 Methodology

3.1 Basic Modeling Architecture

The model choice is determined by the fact that some of our datasets are small. Thus, the model should have only few trainable parameters to avoid overfitting. We utilize a simple attention mechanism, as it is comon for NLP applications. The input words w_j are mapped to their word embeddings $e_{w_j} \in \mathbb{R}^D$ with $j = 1, ..., S$, and S being the input sequence length and D the dimensionality of the input word vectors. These are GloVe embeddings of 50 dimensions pre-trained on 6B English tokens of the "Wikipedia 2014 + Gigaword 5" dataset (Pennington et al., 2014). Then, it computes the attention a_i of each word using the trainable attention vector $e \in \mathbb{R}^D$ via $a_j = e \cdot e_{w_j}$. It takes the accordingly weighted average $z_n = \sum_{i=1}^{S} a_i \cdot e_{w_i}$ of the word vectors with n denoting the n-th sample or input text.

Finally, the classification head is the sigmoid of a simple linear layer $p_n = \text{softmax}(W \cdot z_n + b)$, with $W \in \mathbb{R}^{L \times D}$ and $b \in \mathbb{R}$ as the weights of the model. We refer to this last layer and to p_n as *latent truth layer* or *latent truth*.

3.2 End-to-End Crowdsourcing Model

On top of the basic modeling architecture, the biases of the annotators are modeled as seen in figure 1. The theory is explained by Zeng et al. (2018) as follows:

"*The labeling preference bias of different annotators cause inconsistent annotations. Each annotator has a coder-specific bias in assigning the samples to some categories. Mathematically speaking, let* $\mathcal{X} = \{x_1, \ldots, x_N\}$ *denote the data,* $y^c = [y_1^c, \ldots, y_N^c]$ *the regarding annotations by coder* c*. Inconsistent annotations assume that* $P(y_n^c|x_n) \neq P(y_n^{\hat{c}}|x_n), \forall x_n \in \mathcal{X}, c \neq \hat{c}$*, where* $P(y_n^i|x_n)$ *denotes the probability distribution that coder* c *annotates sample* x_n*.*

LTNet assumes that each sample x_n *has a latent truth* y_n*. Without the loss of generality, let us suppose that LTNet classifies* x_n *into the category* i *with probability* $P(y_n = i|x_n; \Theta)$*, where* Θ *denotes the network parameters. If* x_n *has a ground truth of* i*, coder* c *has an opportunity of* $\tau_{ij}^c = P(y_n^c = j|y_n = i)$ *to annotate* x_n *as* j*, where* y_n^c *is the annotation of sample* x_n *by coder* c*. Then, the sample* x_n *is annotated as label* j *by coder* c *with a probability of* $P(y_n^c = j|x_n; \Theta) = \sum_{i=1}^{L} P(y_n^c = j|y_n = i)P(y_n = i|x_n; \Theta)$*, where* L *is the number of categories and* $\sum_{j=1}^{L} P(y_n^c = j|y_n = i) = \sum_{j=1}^{L} \tau_{ij}^c = 1$*.*
$T^c = [\tau_{ij}^c]_{L \times L}$ *denotes the transition matrix (also referred to as annotator bias) with rows summed to* 1 *while* $[p_n]_i = P(y_n = i|x_n; \Theta)$ *is modeled by the base network (Zeng et al., 2018). We define* $[p_n^c]_j = P(y_n^c = j|x_n; \Theta)$*. Given the annotations from* C *different coders on the data, LTNet aims to maximize the log-likelihood of the observed annotations. Therefore, parameters in LTNet are learned by minimizing the cross entropy loss of the predicted and observed annotations for each coder* c*."*

We represent the annotations and predictions as vectors of dimensionality L such that y_n^c is one-hot encoded and p_n^c contains the probabilities for all class predictions of sample n. The

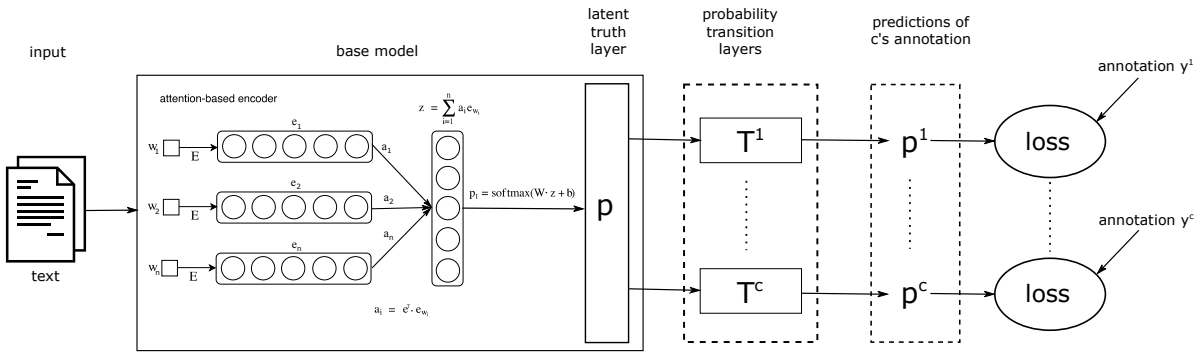


Figure 1: Architecture of the end-to-end trainable LTNet (Zeng et al., 2018). The base model is a simple attention model with a single trainable attention vector e and linear layer with parameters W and b. The transition matrices T^c are the bias matrices from the annotators c. "Each row of the transition matrix T is constrained to be summed to 1" (Zeng et al., 2018). The base model is inspired by ABAE (He et al., 2017).

cross entropy loss function is then defined as $-\sum_{n=1}^{C}\sum_{n=1}^{N}\log(p_n^{c\intercal}\cdot y_n^c)$.

3.3 The Effect of Logarithm Removal on Cross Entropy

The logarithm in the cross entropy formula leads to an exponential increase in the loss for false negative predictions, i.e., when the predicted probability $[p_n^c]_i$ for a ground truth class i is close to 0 and $[y_n^c]_i$ is 1. This increase can be helpful in conditions with numerical underflow, but at the same time this introduces a disproportionate high loss of the other class due to constantly misclassified items. This happens in crowdsourcing, for example, when one annotator is a spammer assigning a high degree of random annotations, which in turn leads to a disproportionally higher loss caused by that annotator's many indistinguishable false negative annotations. Consequentially, the bias matrix of that annotator would be biased towards the false classes. Moreover, this annotator would cause overall more loss than other annotators, which can harm the model training for layers which are shared among all annotators, e.g., the latent truth layer when it is actually trained.

By omitting the log function, these effects are removed and all annotators and datapoints contribute with the same weight to the overall gradient and to the trainable annotator bias matrices, independent of the annotator and his respective annotation behavior. As a consequence, the annotator matrices are capable of modeling the real annotator bias, which is the mismatch between an annotation y_n^c of coder c and the latent truth prediction p_n. If p_n is one-hot encoded, this results to the according classification ratios of samples and is equal to the confusion matrix, without an algorithmically encoded bias towards a certain group of items. This is shown mathematically in the following, where it is assumed that the base network is fixed, i.e., backpropagation is performed through the bias matrices and stops at the latent truth layer.

We define $N=\sum_{k=1}^{L}N_k$ as the number of all samples and N_k of class $k=1,...,L$. L is the number of classes, $T^c=[\tau_{ij}^c]_{L\times L}$ the bias matrix of coder c, p_n the latent truth vector of sample $n=1,...,N$, and p_n^c the annotator prediction. p_{km} is the latent truth of the m-th sample of class k with $m=1,...,N_k$, same for x_{km} and y_{km}^c. The loss without logarithm is

$$
\begin{aligned}
\boldsymbol{O} &= -\sum_{n=1}^{N} p_n^{c\intercal}\cdot y_n^c \\
&= -\sum_{k=1}^{L}\sum_{m=1}^{N_k} p_{km}^{\intercal}\cdot T^c\cdot y_{km}^c \\
&= -\sum_{k=1}^{L}\sum_{m=1}^{N_k} p_{km}^{\intercal}\cdot \begin{pmatrix}\tau_{1k}^c\\ \vdots \\ \tau_{Lk}^c\end{pmatrix} \\
&= \sum_{k=1}^{L}\sum_{m=1}^{N_k}\sum_{h=1}^{L} -[p_{km}]_h\cdot\tau_{hk}^c
\end{aligned}
$$

Apparently, the derivation step between the second and third line would not work if there would be the logarithm from the standard cross entropy. Now, let the learning rate be α, the number of epochs E and the starting values of the initialized bias matrix $(\tau_{lh}^c)_0$. The bias parameters τ_{lh}^c of the bias matrix T^c are updated according to

Figure 2: Male and female bias (top) and confusion (bottom) matrices which are trained using cross entropy loss with and without logarithm in two different settings. The left side has only the original annotations, whereas the right side has 80% random male labels.

$$
\begin{aligned}
(\tau^c_{hk})_E &= (\tau^c_{hk})_0 + \sum_{i=1}^{E} \alpha \left(\frac{\partial \boldsymbol{O}}{\partial \tau^c_{hk}} \right)_i \\
&= (\tau^c_{hk})_0 + \sum_{i=1}^{E} \alpha \left[\sum_{m=1}^{N_k} - [p_{km}]_h \right]_i \\
&= (\tau^c_{hk})_0 - \alpha E \underbrace{\sum_{m=1}^{N_k} [p_{km}]_h}_{=:Z_{hk}}
\end{aligned}
$$

For sufficiently large E the starting values $(\tau^c_{hk})_0$ become infinitesimally small in comparison to the second additive term and thus negligible. As we are normalizing the rows of $(T^c)_E$ after training so that the bias fulfills our probability constraint defined in section 3.2, the linear factor $-\alpha E$ is canceled out, too. Thus, the bias matrix T^c results in the row normalized version of $[Z_{hk}]_{L\times L}$. Z_{hk} is the sum of the latent truth probabilities for class h on all samples of a ground truth class k. If we assume that the latent truth is one hot encoded, $[Z_{hk}]_{L\times L}$ equals to the confusion matrix, of which the k-th column sums up to the number of samples in class k: $\sum_{h=1}^{L} Z_{hk} = \sum_{h=1}^{L} \sum_{m=1}^{N_k} [p_{km}]_h = \sum_{m=1}^{N_k} 1 = N_k$.

4 Experiments

4.1 Bias Convergence

The following experiment compares how training with and without the logarithm in the cross entropy loss affects the LTNet bias matrices empirically. The mathematical explanations in section 3.3 suggest that the logarithm removal from cross entropy leads to an annotator bias matrix identical to the confusion matrix, which would not be the case for the normal cross entropy.

Experiment Description. For the data, we use the TripAdvisor dataset from Thelwall et al. consisting of 11,900 English consumer reviews about hotels from male and female reviewers plus their self-assigned sentiment ratings (Thelwall, 2018). We use the gender information to split the data into two annotator groups, male and female, from which we model each one with a corresponding bias matrix. We exclude neutral ratings and binarize the rest to be either positive or negative. As the dataset is by default completely balanced regarding gender and sentiment at each rating level, it is a natural candidate for correct bias approximation. Throughout our experiments, we use 70% of the obtained data as training, 20% as validation and the 10% remaining as test sets.

Similar to the explanation in 3.3, the base model with its latent truth predictions is pre-trained on all samples and then frozen when the bias matrices are trained. The stochastic gradient descent method is used to optimize the parameters, as other widespread optimizers, such as Adam and AdaGrad (the latter introduced that feature first), introduce an – in our case undesired – bias towards certain directions in the gradient space, namely by using the previous learning steps to increase or decrease the weights along dimensions with larger or smaller gradients (Kingma and Ba, 2014). For all four sub-experiments, we train the base models with varying hyperparameters and pick the best based on accuracy. We train the transition matrices 50 times with different learning rates from the interval $[1e{-}6, 1e{-}3]$. The batch size is 64. In addition to a normal training setting, we add random annotations to 80% of the instances annotated by male

subjects, such that 40% from them are wrongly annotated. This results in four models: with and without logarithm in the cross entropy, with and without random male annotations, each time respectively with two annotator group matrices, male and female – see figure 2.

Results. The bias matrices of the models with the best accuracy are picked and presented in figure 2 in the top row. The corresponding confusion matrices depict the mismatch between latent truth predictions and annotator-group labels in the bottom row. The bias matrices trained without logarithm in the cross entropy are almost identical to the confusion matrices in all cases, which never holds for the normal cross entropy. This confirms our mathematically justified hypothesis given in section 3.3 that the logarithm removal from cross entropy leads to a correctly end-to-end-trained bias. In this context, it is relevant that the related work shows the same mismatch between bias and confusion matrix when applying cross entropy loss without explaining nor tackling this difference, see Zeng et al. (2018, figure 5) and Rodrigues and Pereira (2018, figure 3).

It is worth mentioning for the 80% random male annotations that these are correctly modeled without cross entropy, too, as opposed to normal cross entropy. If the goal is to model the annotator bias correctly in an end-to-end manner, this might be considered as particularly useful to analyze annotator behavior, e.g., spammer detection, later on.

Finally, we report how much variation the bias matrices show during training for cross entropy with and without logarithm. As mentioned in the experiment description, we trained each model 50 times. The elements of the resulting bias matrices with standard cross entropy have on average 7.7% standard deviation compared to 2.8% without logarithm. It can be concluded that the bias produced by standard cross entropy is less stable during training, which raises questions about the overall reliability of its outcome.

In summary, the observations confirm our assumptions that cross entropy without logarithm captures annotator bias correctly in contrast to standard cross entropy. This carries the potential to detect spammer annotators and leads to an overall more stable training.

4.2 Ground Truth Estimation

In the following paragraphs, we demonstrate how to estimate the ground truth based on the latent truth from LTNet. This is then compared to two other kinds of ground truth estimates. All of them can be applied in a single label crowdsourcing setting.

The Dawid-Skene algorithm (Sinha et al., 2018) is a common approach to calculate a ground truth in crowdsourcing settings where there are multiple annotations given on each sample. This method is, for instance, comparable to majority voting, which tends to give similar results for ground truth estimation. However, in single label crowdsourcing settings, these approaches are not feasible. Under single label conditions, the Dawid-Skene ground truth estimates equal to the single label annotations.

This is given by Sinha et al. (2018, formula 1) in the expectation step, where the probability for a class $k \in 1, 2, ..., L$ given the annotations is defined as

$$P(Y_n = k|k_{n_1}, k_{n_2}, ..., k_{n_L}) = \frac{\left(\prod_{c=1}^{C} P(k_{n_c}|Y_n = k)\right) \cdot P(Y_n = k)}{\sum_{k=1}^{L}\left(\prod_{c=1}^{C} P(k_{n_c}|Y_n = k)\right) \cdot P(Y_n = k)}.$$

Here, n is the sample to be estimated, C the number of annotators for that sample, $n_1, n_2, ..., n_C$ the set of annotators who labeled this sample, $k_{n_1}, k_{n_2}, ..., k_{n_C}$ the set of annotation choices chosen by these C participants for sample n, and Y_n the correct (or aggregated) label to be estimated for the sample n (Sinha et al., 2018).

In the single label case C equals to 1, which reduces the formula to $P(Y_n = k|k_{n_1}, k_{n_2}, ..., k_{n_C}) = P(Y_n = k|k_{n_1})$. This in turn equals to 1 if k is the assigned class label to sample n by annotator n_1, or 0 otherwise. In other words, if there is only one annotation per sample, this annotation defines the ground truth. Since different annotators do not assign labels on the same samples, there is also no way to model mutual dependencies of each other.

LTNet, however, provides estimates for all variables from this formula. $P(Y_n = k)$ is the prior and is approximated by the latent truth probability for class k of sample n. $P(k_{n_c}|Y_n = k)$ is the probability that, assuming k would be the given class, sample n is labeled as k_{n_c} by annotator n_c. This equals to $\tau^c_{k_{n_c},k}$, i.e., the entries of the LTNet bias matrix T^c of annotator c.

Eventually, the LTNet ground truth can be derived by choosing k such that the probability $P(Y_n = k|k_{n_1}, ...)$ is maximized:

$$k_{\text{ground truth}} = \arg\max_k P(Y_n = k | k_{n_1}, ...).$$

We will leverage this formula to derive and evaluate the ground truth generated by LTNet.

Experiment We calculate the LTNet ground truth according to the previous formula on the organic dataset, a singly labeled crowdsourcing dataset, which is described in Section 4.3. To demonstrate the feasibility and the soundness of the approach, we compare it with two other ways of deriving a ground truth. Firstly, we apply the fast Dawid-Skene algorithm on the annotator-wise class predictions from the LTNet model. Secondly, we train a base network on all annotations while ignoring which annotator annotated which samples. Eventually, we compare the ground truth estimates of all three methods by calculating Cohen's kappa coefficient (Cohen, 1960), which is a commonly used standard to analyze correspondence of annotations between two annotators or pseudo annotators. The training procedures and the dataset are identical to the ones from the classification experiments in Section 4.3.

Results As can be seen on Table 1, the three ground truth estimators are all highly correlated to each other, since the minimal Cohen's kappa score is 0.98. Apparently, there are only minor differences in the ground truth estimates, if any at all. Thus, it appears that the ground truths generated by the utilized methods are mostly identical. Especially, the LTNet and Dawid-Skene ground truths are highly correlated with a kappa of 99%. The base model, which is completely unaware of which annotator labeled which sample, is slightly more distant with kappas between 98% – 99%. So with respect to the ground truth itself, we do not see a specific benefit of any method, since they are almost identical.

However, it must be noted that LTNet additionally produces correct bias matrices of every annotator during model training, which is not the case for the base model. Correct biases have the potential to help improving model performance by analyzing which annotators tend to be more problematic and weighting them accordingly.

4.3 Classification

We conduct classification comparing LTNet in different configurations on three datasets with crowdsourced sentiment annotations to discuss the potential related benefits and drawbacks of our proposed loss modification.

Ground truths	Dawid Skene	LTNet	Basic Model
Dawid Skene	1.0000	0.9905	0.9832
LTNet	0.9905	1.0000	0.9918
Base Model	0.9832	0.9918	1.0000

Table 1: Cohen's kappa scores between three different ground truth estimation methods applied on the singly labeled crowdsourced organic dataset.

Emotion Dataset. The emotion dataset consists of 100 headlines and their ratings for valence by multiple paid Amazon Mechanical Turk annotators (Snow et al., 2008). Each headline is annotated by 10 annotators, and each annotated several but not all headlines. We split the interval-based valence annotations to positive, neutral, or negative. Throughout our experiments, we used 70% of the obtained data as training, 20% as validation and 10% as test sets.

Organic Food Dataset. With this paper, we publish our dataset containing social media texts discussing organic food related topics.

Source. The dataset was crawled in late 2017 from Quora, a social question-and-answer website. To retrieve relevant articles from the platform, the search terms "organic", "organic food", "organic agriculture", and "organic farming" are used. The texts are deemed relevant by a domain expert if articles and comments deal with organic food or agriculture and discuss the characteristics, advantages, and disadvantages of organic food production and consumption. From the filtered data, 1,373 comments are chosen and 10,439 sentences annotated.

Annotation Scheme. Each sentence has sentiment (positive, negative, neutral) and entity, the sentiment target, annotated. We isolate sentiments expressed about organic against non-organic entities, whereas for classification only singly labeled samples annotated as organic entity are considered. Consumers discuss organic or non-organic products, farming practices, and companies.

Annotation Procedure. The data is annotated by each of the 10 coders separately; it is divided into 10 batches of 1,000 sentences for each annotator and none of these batches shared any sentences between each other. 4616 sentences contain organic entities with 39% neutral, 32% positive, and 29% negative sentiments. After annotation, the

Dataset	Model	F1 %	Acc %
TripAdvisor	Base Model	88.92	88.91
	LTNet w/o log	**89.71**	**89.71**
	LTNet	89.39	89.39
Organic	Base Model	32.08	45.75
	LTNet w/o log	**44.71**	**50.54**
	LTNet	40.51	47.77
Emotion	Base Model	51.74	56.00
	LTNet w/o log	58.15	63.00
	LTNet	**61.23**	**66.00**
	Base Model DS	44.17	54.00

Table 2: Macro F1 scores and accuracy measured in the classification experiment.

data splits are 80% training, 10% validation, and 10% test set. The data distribution over sentiments, entities, and attributes remains similar on all splits.

Experiment Description. The experiment is conducted on the TripAdvisor, organic, and emotion datasets introduced in section 4.3. We compare the classification of the base network with three different LTNet configurations. Two of them are trained using cross entropy with and without logarithm. For the emotion dataset, we compute the bias matrices and the ground truth for the base model using the fast Dawid-Skene algorithm (Sinha et al., 2018). This is possible for the emotion dataset, since each sample is annotated by several annotators.

We apply pre-training for each dataset by training several base models with different hyperparameters and pick the best based on accuracy. Eventually, we train the LTNet model on the crowdsourcing annotation targets by fine-tuning the best base model together with the bias matrices for the respective annotators. The bias matrices are initialized as row normalized identity matrices plus uniform noise around 0.1. The models are trained 50 times with varying learning rates sampled from between $[1e{-}6, 1e{-}3]$. A batch size of 64 is used.

Results. The classification results of the models are presented in table 2 with their macro F1 score and accuracy as derived via predictions on the test sets. LTNet generally shows a significant classification advantage over the base model. On all three databases, LTNet approaches performed better on the test datasets. The LTNet improvement has a big delta of $11\% + / - 1\%$ when there is a low annotation reliability (organic and emotion datasets) and a small delta $< 1\%$ with high reliability (TripAdvisor) [3]. Apparently, model each annotator separately gives significant advantages.

Regarding the comparison between cross entropy (CE) loss with and without logarithm on LTNet, the removed logarithm shows better classification results on organic ($+3\%$) and TripAdvisor data ($+0.3\%$) and worse on the emotion dataset (-3%). This means that on both of the singly labeled crowdsourcing datasets, the removal of the logarithm from the loss function leads to better predictions than the standard CE loss. On the multi-labeled emotion dataset, however, this does not appear to be beneficial. As this data has only a very small test set of 100 samples, it is not clear if this result is an artifact or not. Concluding, the log removal appears to be beneficial on large datasets, where the bias is correctly represented in the training and test data splits, such that it can be modeled correctly by the denoted approach. It shall be noted, that it is not clear if that observation would hold generally. We advice to run the same experiments multiple times on many more datasets to substantiate this finding.

5 Conclusion

We showed the efficacy of LTNet for modeling crowdsourced data and the inherent bias accurately and robustly. The bias matrices produced by our modified LTNet improve such that they are more similar to the actual bias between the latent truth and ground truth. Moreover, the produced bias shows high robustness under very noisy conditions making the approach potentially usable outside of lab conditions. The latent truth, which is a hidden layer below all annotator biases, can be used for ground truth estimation in our single label crowdsourcing scenario, providing almost identical ground truth estimates as pseudo labeling. Classification on three crowdsourced datasets show that LTNet approaches outperfom naive approaches not considering each annotator separately. The proposed log removal from the loss function showed better results on singly labeled crowdsourced datasets, but this observation needs further experiments to be substantiated. Furthermore, there might be many use cases to explore the approach on other tasks than sentiment analysis.

[3] Unreliable means that the provided annotations have a low Cohen's kappa inter-rater reliability on the organic 51.09% and emotion (27.47%) dataset. On the organic dataset we prepared a separate data partition of 300 sentences annotated by all annotators for that purpose. For the TripAdvisor dataset, it is apparent that the correspondence of annotations between the two annotator groups (male and female) is high as can be seen in figure 2 for cross entropy without logarithm.

References

Michael P. J. Camilleri and Christopher K. I. Williams. 2020. The extended dawid-skene model. In *Machine Learning and Knowledge Discovery in Databases*, pages 121–136, Cham. Springer International Publishing.

Bob Carpenter. 2008. Multilevel bayesian models of categorical data annotation.

Zhendong Chu, Jing Ma, and Hongning Wang. 2020. Learning from crowds by modeling common confusions.

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

A. P. Dawid and A. M. Skene. 1979. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied Statistics*, 28(1):20–28.

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

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2017. An unsupervised neural attention model for aspect extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 388–397, Vancouver, Canada. Association for Computational Linguistics.

Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. 2013. Learning whom to trust with MACE. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Atlanta, Georgia. Association for Computational Linguistics.

Ashish Khetan, Zachary C. Lipton, and Anima Anandkumar. 2017. Learning from noisy singly-labeled data.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

J. Kossaifi, R. Walecki, Y. Panagakis, J. Shen, M. Schmitt, F. Ringeval, J. Han, V. Pandit, A. Toisoul, B. Schuller, K. Star, E. Hajiyev, and M. Pantic. 2021. Sewa db: A rich database for audio-visual emotion and sentiment research in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(3):1022–1040.

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*, Portland, Oregon, USA. Association for Computational Linguistics.

M. Munezero, C. S. Montero, E. Sutinen, and J. Pajunen. 2014. Are they different? affect, feeling, emotion, sentiment, and opinion detection in text. *IEEE Transactions on Affective Computing*, 5(2):101–111.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. 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 188–197, Hong Kong, China. Association for Computational Linguistics.

Andreas Peldszus and Manfred Stede. 2013. Ranking the annotators: An agreement study on argumentation structure. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 196–204, Sofia, Bulgaria. 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.

Emmanouil Antonios Platanios, Maruan Al-Shedivat, Eric Xing, and Tom Mitchell. 2020. Learning from imperfect annotations.

Vikas C. Raykar and Shipeng Yu. 2012. Eliminating spammers and ranking annotators for crowdsourced labeling tasks. *J. Mach. Learn. Res.*, 13:491–518.

Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Anna K. Jerebko, Charles Florin, Gerardo Hermosillo Valadez, Luca Bogoni, and Linda Moy. 2009. Supervised learning from multiple experts: whom to trust when everyone lies a bit. In *ICML*, volume 382 of *ACM International Conference Proceeding Series*, pages 889–896. ACM.

Filipe Rodrigues and Francisco C. Pereira. 2018. Deep learning from crowds. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1):1611–1618.

Vaibhav B Sinha, Sukrut Rao, and Vineeth N Balasubramanian. 2018. Fast dawid-skene: A fast vote aggregation scheme for sentiment classification.

Padhraic Smyth, Usama Fayyad, Michael Burl, Pietro Perona, and Pierre Baldi. 1995. Inferring ground truth from subjective labelling of venus images. In *Advances in Neural Information Processing Systems*, volume 7, pages 1085–1092, San Diego, CA. MIT Press.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Y. Ng. 2008. Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *EMNLP '08: Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Morristown, NJ, USA. Association for Computational Linguistics.

DJ Spiegelhalter and PGI Stovin. 1983. An analysis of repeated biopsies following cardiac transplantation. *Statistics in medicine*, 2(1):33–40.

Mike Thelwall. 2018. Gender bias in sentiment analysis. *Online Information Review*, 42(3):343–354.

TIAN TIAN and Jun Zhu. 2015. Max-margin majority voting for learning from crowds. In *Advances in Neural Information Processing Systems*, volume 28, pages 1621–1629, San Diego, CA. Curran Associates, Inc.

Hao Wang, Bing Liu, Chaozhuo Li, Yan Yang, and Tianrui Li. 2019. Learning with noisy labels for sentence-level sentiment classification. *CoRR*, abs/1909.00124.

Fabian L. Wauthier and Michael I. Jordan. 2011. Bayesian bias mitigation for crowdsourcing. In *NIPS*, volume 24, pages 1800–1808, San Diego, CA. Curran Associates, Inc.

Peter Welinder, Steve Branson, Serge Belongie, and Pietro Perona. 2010. The multidimensional wisdom of crowds. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 2*, Red Hook, NY, USA. Curran Associates Inc.

Jacob Whitehill, Paul Ruvolo, Tingfan Wu, Jacob Bergsma, and Javier R. Movellan. 2009. Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *NIPS*, pages 2035–2043, San Diego, CA. Curran Associates, Inc.

Yan Yan, Rmer Rosales, Glenn Fung, Mark W. Schmidt, Gerardo Hermosillo Valadez, Luca Bogoni, Linda Moy, and Jennifer G. Dy. 2010. Modeling annotator expertise: Learning when everybody knows a bit of something. In *AISTATS*, volume 9 of *JMLR Proceedings*, pages 932–939, Chia Laguna Resort, Sardinia, Italy. JMLR.org.

Jiabei Zeng, Shiguang Shan, and Xilin Chen. 2018. Facial expression recognition with inconsistently annotated datasets. In *ECCV (13)*, volume 11217 of *Lecture Notes in Computer Science*, pages 227–243, Red Hook, NY, USA. Springer.

Speech Technology for Everyone: Automatic Speech Recognition for Non-Native English with Transfer Learning

Toshiko Shibano* **Xinyi Zhang*** **Mia Taige Li*** **Haejin Cho***
Peter Sullivan* **Muhammad Abdul-Mageed***
The University of British Columbia
tshibano@student.ubc.ca, {jeremyzxy0803, mia.taige.li, haejin2909}@gmail.com,
prsull@student.ubc.ca, muhammad.mageed@ubc.ca

Abstract

To address the performance gap of English ASR models on L2 English speakers, we evaluate fine-tuning of pretrained wav2vec 2.0 models (Baevski et al., 2020; Xu et al., 2021) on L2-ARCTIC, a non-native English speech corpus (Zhao et al., 2018) under different training settings. We compare **(a)** models trained with a combination of diverse accents to ones trained with only specific accents and **(b)** results from different single-accent models. Our experiments demonstrate the promise of developing ASR models for non-native English speakers, even with small amounts of L2 training data and even without a language model. Our models also excel in the zero-shot setting where we train on multiple L2 datasets and test on a blind L2 test set.

Index Terms: Automatic Speech Recognition (ASR), ASR for L2 English speakers

1 Introduction

Although non-native (L2) English speakers outnumber native (L1) English speakers (Crystal, 2003), major challenges contribute to a gap between performance of ASR systems on L2 speech, mainly due to the influence of L1 pronunciation on the learned language, and the lack of annotated L2 speech data (Radzikowski et al., 2021; Viglino et al., 2019). To meet these challenges, previous studies have exhibited two distinct approaches. The first is to make L2 speech representations more closely match those of L1 speech (Radzikowski et al., 2021). The second approach leverages L2 speech data to improve model robustness. Due to L2 data scarcity, and hence the challenge of training L2 models from scratch, this second approach necessitates employment of transfer learning or domain adaptation (Shi et al., 2021; Sun et al., 2018).

Figure 1: The various data splits we use in our experiments. Shade represents a different run of our training, with the gradient blocks in Split 4 being present in all runs. For cross validation splits, we show a single fold as an example, where number indicates the participants included.

State-of-the-art ASR models based on unsupervised/self-supervised pre-training such as wav2vec (Schneider et al., 2019) and wav2vec 2.0 (Baevski et al., 2020)[1] offer a tantalizing starting point for applying the second approach we list above, especially due to their strong performance on ASR even without a language model. However, challenges remain in identifying how best to apply models such as wav2vec 2.0 in L2 fine-tuning scenarios. For this reason, our objective in the current work is to investigate a rich set of conditions under which we can fine-tune ASR models for optimal L2 performance. More concretely, we attempt to achieve the following:

1. Evaluate fine-tuning strategies for adapting

*All authors contributed equally.

[1]Although sometimes referred to as 'unsupervised', these models employ a self-supervised objective.

pre-trained L1 English ASR models to L2 English;

2. Explore impact of non-native (L2) accents on performance of these fine-tuned ASR models, comparing multi-accent training to single-accent training; and

3. Quantify the impact of L2 fine-tuning on model performance for L1 English speech recognition.

Although external language models are often used in improving ASR performance (Nakatani, 2019; Xu et al., 2020), models trained with great quantities of data can potentially internalize this linguistic information (Graves and Jaitly, 2014). In particular, some of the wav2vec 2.0 models perform nearly as well with and without a language model on difficult speech such as LibriSpeech Test-Other (Xu et al., 2021). We thus use this robust pre-trained model as our starting point, and carry out our work without use of an external language model to see if this performance is retained through the fine-tuning process.

The rest of the paper is organized as follows: Section 2 is an overview of related works. We describe our data in Section 3. Section 4 is about our experiments and results. We conclude in Section 5.

2 Related Work

Because of the difficulty in linguistically annotating corpora for Hidden Markov Model (HMM)-based ASR (Graves and Jaitly, 2014), researchers have broadly embraced End-to-End (E2E) deep learning architectures either based on Connectionist Temporal Classification (CTC) (Graves et al., 2006; Graves and Jaitly, 2014), Attention (Chorowski et al., 2015; Chan et al., 2016; Gulati et al., 2020), or hybrids of the two (Watanabe et al., 2017; Wang et al., 2020). Recent efforts inspired by work such as BERT (Devlin et al., 2019) have improved on these purely supervised learning baselines through self-supervised pre-training (Schneider et al., 2019; Baevski et al., 2019, 2020) and self-training (Xu et al., 2021). These self-supervised wav2vec models represent one line of research in speech representation. Other works include models similar to wav2vec that also use a contrastive loss (Oord et al., 2018), models using an autoregressive loss function (Ling et al., 2020; Chung et al., 2019), as well as models using a masked language model closer to the original BERT (Liu et al., 2020a).

With these efforts, ASR technologies for native languages have evolved significantly. However, we still observe problems in many applications. In particular, several researchers have emphasized how performance of ASR models drops when the input speech is from non-native speakers whose native languages are different from the models' target languages (Radzikowski et al., 2021; Livescu and Glass, 2000; Wang et al., 2003; Ping, 2008). For systems developed for English ASR, this can be a real issue. The reason, as observed earlier, is that large populations of English language speakers are non-native (Crystal, 2003). In line with this argument, Ping (2008), for example, pointed out the necessity to improve speech recognition technology for L2 speakers given that many people speak more than one language for economic and social reasons, especially considering human migration is becoming more common these days. It is hoped that continued efforts aiming at improving ASR for non-native speakers will eventually lead to improved results for many as voice recognition technology becomes increasingly pervasive in our daily lives (Ping, 2008).

As we explained in Section 1, there are two distinct approaches to improve current ASR performance on L2 speech: 1) accent conversion as an extension to the active area of research of voice conversion; and 2) incorporation of L2 speech data, which is often limited in quantity and quality, during the model training process.

The first approach takes inspiration from voice conversion, but instead of focusing on modifying the pitch, it modifies the pronunciation to reduce accents. Additionally, voice conversion models aim to generate results that are speaker-dependent, while accent conversion models deal with generalizing accents from a group of speakers, hence being speaker-independent. With this approach, the resulting model can be used as a pre-processing step to remove accents in the data prior to feeding these data into an ASR model. Bearman et al. (2017) adopt this approach but focus on L1 English accents, while Radzikowski et al. (2021) work on L2 English accents with speakers' L1 being Japanese. Liu et al. (2020b) took a step further and turned Hindi-accented English to native American English without utilizing native utterances.

The second approach often employs techniques such as domain adversarial training and transfer learning in order to utilize as much available ac-

cented speech data as possible. Domain adversarial training (DAT) is a popular approach as it encourages models to learn accent-invariant features (Sun et al., 2018; Hou et al., 2019; Hu et al., 2021). Transfer learning is another popular approach in L2 speech recognition, as it possibly allows a model to gain knowledge from both the base task and the new task, even when the new task has limited data (Matassoni et al., 2018; Das et al., 2021; Shi et al., 2021). In the Accented English Speech Recognition Challenge 2020 (AESRC2020), many teams utilize transfer learning to tackle the L2 accent recognition task (Shi et al., 2021). In a recent work, Das et al. (2021) combine both DAT and transfer learning to achieve robust accented speech recognition performance. We now introduce our data.

3 Data

3.1 Corpus Information

We choose **L2-ARCTIC**, a non-native English speech corpus (Zhao et al., 2018), for L2 fine-tuning. The recordings are from 24 non-native speakers of English with a total of six different L1s, and each of the L1s consists of two female speakers and two male speakers. The L1s we use for our experiments are Arabic (AR), Hindi (HI), Korean (KO), Mandarin (ZH), Spanish (ES), and Vietnamese (VI). Because L2-ARCTIC is based on the original L1 English corpus, CMU ARCTIC (Kominek et al., 2003) (henceforth **L1-ARCTIC**, for simplicity), we can easily evaluate performance from fine-tuning on same-domain L1 data.

Each speaker in L2-ARCTIC contributed approximately one hour of phonetically-balanced read speech based on the L1-ARCTIC prompts, which consist of carefully selected sentences ($1,132$ sentence prompts) from Project Gutenberg (Kominek et al., 2003). We note this, as the pretrained wav2vec 2.0 model we use was first pre-trained on LibriSpeech[2] (Panayotov et al., 2015) and then self-trained on Libri-Light[3] (Kahn et al., 2020). Both corpora rely on audiobooks from the LibriVox project,[4] much of which comes from Project Gutenberg.[5] This minimizes discrepancies between domains of the text.

[2]http://www.openslr.org/12/
[3]https://github.com/facebookresearch/libri-light
[4]https://librivox.org
[5]http://www.gutenberg.org

We also evaluate our fine-tuned models on **1) LibriSpeech** to compare the fine-tuning with the original performance of self-trained wav2vec 2.0 Large (LV-60) model (Xu et al., 2021), which we will refer to as *Wav2Vec 2.0-ST*. In addition, we evaluate on **2) L1-ARCTIC**, identical to our L2-ARCTIC corpus but spoken by four native US English speakers, allowing us to identify any degradation in performance on L1 speech. Each of L1-ARCTIC speakers' datasets contain approximately the same number of utterances ($n =\sim 1,132 * 4$) as each of L2-ARCTIC speakers' datasets.

For the purpose of our experiments, we define *native (L1) accents* as those represented in the LibriSpeech and L1-ARCTIC, and *non-native (L2) accents* as those represented in L2-ARCTIC.

3.2 Data Splits

For both L2-ARCTIC and L1-ARCTIC, we split the data into three distinct Train, Dev, and Test sets with an 80:10:10 ratio. Importantly, we ensure there is *no overlap between utterances*. For L2-ARCTIC, we split the data across the following settings (see Fig. 1).

- **Split-1** *(speaker-dependent, multi-accent split)*: All speakers from all accents in the Train set are also included in the Dev and Test sets; however, no utterances are shared between Train, Dev, and Test.

- **Split-2** *(speaker-independent cross-validation splits with multiple accents)*: A speaker from each accent[6] is removed from the Train and Dev sets, but other speakers with the same accent remain in the Train and Dev sets.

- **Split-3** *(speaker-independent zero-shot splits with multiple accents)*: All speakers from one of the accents are entirely removed from the Train and Dev sets. The removed speakers are included in Test.

- **Split-4** *(all-speaker, single-accent split)*: Speakers are broken down by accents (six accents in total) and all speakers in a given accent are split into the Train, Dev, and Test sets (3 data splits x 6 accents).

- **Split-5** *(speaker-independent cross-validation splits with single accent)*: One speaker in each

[6]We use the term 'accent' here to loosely refer to variation in speakers with L1 other than English.

		Accent dependency		Speaker dependency	
		Dependent	**Independent**	**Dependent**	**Independent**
Multi-accent	Model-1 (Split 1)	x		x	
	Model-2 (Split 2)	x			x
	Model-3 (Split 3)		x		x
Single-accent	Model-4 (Split 4)	x	x	x	x
	Model-5 (Split 5)	x			x

Table 1: Summary of data splits, fine-tuning, and evaluation setups.

accent is removed from the Train and Dev sets, but the other speakers with the same accent remain in the Train and Dev sets. As there are four speakers per accent, four splits are created for each accent, which are further split into the Train, Dev, and Test sets (3 data splits x 6 accents x 4 speakers).

4 Experiments

For all our wav2vec 2.0 models, we use Fairseq [7] fine-tuning default settings as a reference and convert the hyper-parameters to align with Huggingface's implementation. We train each model with three random seeds and take average over three WERs, one each from the three seeds.

4.1 Model Architecture, Fine-tuning, Baselines, and Evaluation

For our model development, we use the wav2vec 2.0 architecture (Baevski et al., 2020) which is composed of a multi-layer convolutional neural network feature extractor and a Transformer context network. It takes in raw audio and converts it into representations of the input sequence. The encoder consists of multiple blocks of temporal convolution followed by a layer normalization and a GELU activation function. The relative positional embedding in the Transformer is accomplished by a convolutional layer.

Fine-tuning of pre-trained wav2vec 2.0 is performed with CTC and the transcriptions of the audio segments. For each model, we identify the optimal hyper-parameters on the respective Dev set. We choose hyper-parameters as follows: For `mask_feature_prob`, we pick from {*0.25, 0.5*}, for `mask_feature_length`, we choose from {*15, 30*}, for `mask_time_prob` we use {*0.5, 0.75*}, and a batch size of 16. To mimic the tri-state learning rate schedule (Baevski et al., 2020), we set different learning rates for different stages: warm-up (1e-5, 3e-5), constant stage (1e-5, 3e-5), and decay (1e-5, 3e-5, 5e-6). The decay stage is followed by another constant stage (1e-5, 2e-6, 5e-6) to simulate the Fairseq's fine-tuning configuration. We evaluate all our models in terms of word error rate (WER). *All* our results are the average of three runs, and we use the following baselines:

- **Baseline-I:** Wav2Vec 2.0-ST (Xu et al., 2021),[8] a self-trained version of wav2vec 2.0 (Baevski et al., 2020) exploiting a Transformer large architecture and pre-training on 960 hours of speech data from LibriSpeech (Panayotov et al., 2015). The self-training is performed on 60K hours of Libri-Light (Kahn et al., 2020). We believe this as an already strong baseline. We use the model released via HuggingFace. [9]
- **Baseline-II:** This is Wav2Vec 2.0-ST, the same as Baseline-I, fine-tuned on L1-ARCTIC described earlier. The purpose of Baseline-II is to allow for measuring the trade-off of L1 English ASR performance by fine-tuning the English pre-trained model on L2 accents.

4.2 Multi-Accent Models

With our multi-accent models, we examine performance using multiple accents during training. We introduce each of our models here, and present the results acquired with each. We provide a summary of our different data splits and models across accent and speaker dependency categories in Table 1.

Model-1 (speaker- and accent-dependent): The model is fine-tuned with Split-1 data to identify any speaker-dependent training impact, as well as an upper limit on performance. In addition to

[7] https://github.com/pytorch/fairseq

[8] https://github.com/pytorch/fairseq/tree/master/examples/wav2vec#wav2vec2.0

[9] https://huggingface.co/facebook/wav2vec2-large-960h-lv60-self

Model	L2-ARCTIC		L1-ARCTIC		LS_{dev}		LS_{test}	
	Dev	Test	Dev	Test	Clean	Other	Clean	Other
Baseline-I	13.47	12.47	2.30	2.23	**1.69**	**3.55**	**1.86**	**3.89**
Baseline-II	17.29	15.95	**1.26**	**1.30**	2.19	5.13	2.32	5.00
Model-1	**9.78**	**9.27**	1.94	1.86	2.75	5.55	2.82	6.36

Table 2: Model-1 performance in word error rate (WER) (lower is better) on non-native accents (L2-ARCTIC) and native accents (L1-ARCTIC, LS_{dev} and LS_{test}). Baseline-I and Baseline-II are reported on the same Dev and Test sets of each corpus for comparison.

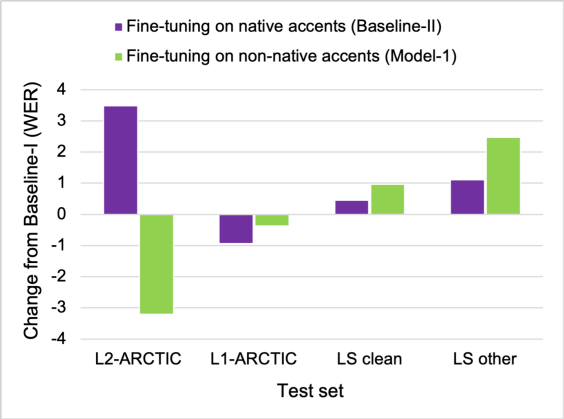


Figure 2: Trade-offs of fine-tuning on native accents (Baseline-II) vs. non-native accents (Model-1). As we evaluate model accuracy by error rate, the bars extending into the negative values mean that the model gains accuracy by fine-tuning.

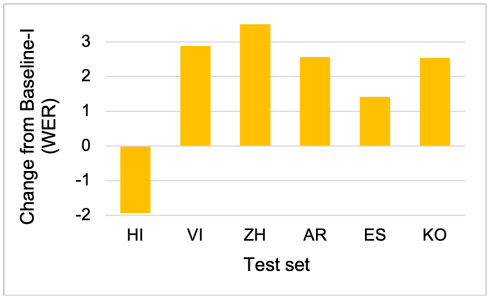


Figure 3: HI-specific Model-4 evaluated on individual accents. As we evaluate model accuracy by error rate, the bars extending downwards represent the performance gain by fine-tuning. HI-specific fine-tuning benefits HI but hinders performance on all the other accents.

Model	Dev_{L2}		$Test_{L2}$	
	Mean	SD	Mean	SD
Baseline-I	13.47	0.23	12.47	0.84
Baseline-II	17.29	0.41	15.96	1.58
Model-2	**9.57**	0.19	**9.96**	0.64

Table 3: Model-2 cross validated performance on L2-ARCTIC Dev and Test sets, alongside Baseline-I and Baseline-II performance on the same cross validation splits. Mean refers to the average WER over the four runs and SD refers to the standard deviation.

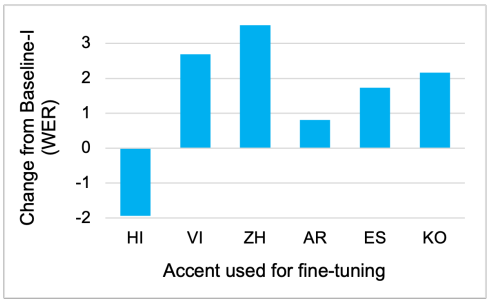


Figure 4: Individual Model-4s evaluated on the HI accent. All the bars except HI extend upwards, meaning that all the other single-accent models hinder performance on the HI accent.

evaluating on L2-ARCTIC Test, we evaluate on L1-ARCTIC Test and LibriSpeech in order to observe any changes in model performance on L1 English.

As Table 2 shows, our Model-1 achieves best performance on both Dev and Test of **L2-ARCTIC** as compared to our two baselines. On Test, our Model-1 acquires 25.66% improvement over our Baseline-I wav2vec 2.0 system on L2-ARCTIC (9.27 WER for our model vs. 12.47 WER for Baseline-I). This gain is not surprising and simply means that a model with access to L2 data for fine-tuning will improve over models fine-tuned with L1 data (Baseline-II, which is fine-tuned on L1-ARCTIC) or not-fine-tuned at all (Baseline-I). Nor is performance on **L1-ARCTIC** surprising: a model fine-tuned with native data (Baseline-II) outperforms one fine-tuned with accented data (our Model-1), both of which outperform a model without fine-tuning (Baseline-I). These results, however, show that in absence of L1 data, L2 data can be valuable for improving ASR model performance even on L1. For **LibriSpeech**, Baseline-I, which is trained on LibriSpeech data, outperforms the two fine-tuned models (our Model-1 and Baseline-II). The reason is that these two latter

	Baseline-I	Baseline-II	Model-3	
$\mathbf{L1_{removed}}$	$\mathbf{Test_{zeroshot}}$	$\mathbf{Test_{zeroshot}}$	$\mathbf{Test_{zeroshot}}$	$\mathbf{Test_{all}}$
VI	23.30	28.81	18.81	9.43
ZH	14.85	19.32	12.13	9.08
AR	10.95	14.82	10.10	9.13
ES	10.48	13.48	8.89	**8.98**
KO	8.18	10.22	6.95	9.01
HI	**6.93**	**8.93**	**6.67**	9.11

Table 4: Model-3 setting, where a different accent is removed each run. $Test_{all}$ refers to Test of *all* 24 speakers, and $Test_{zeroshot}$ refers to Test of those four speakers who have $L1_{removed}$ accent. Baseline-I acquires 12.47 on $Test_{all}$, while Baseline-II acquires 15.95 on the same test set (i.e., $Test_{all}$).

	Baseline-I	Baseline-II	Model-1	Model-4			
L1	$\mathbf{Test_{L2}}$	$\mathbf{Test_{L2}}$	$\mathbf{Test_{L2}}$	$\mathbf{Test_{L2}}$	$\mathbf{Test_{L1}}$	$\mathbf{LS_{Clean}}$	$\mathbf{LS_{Other}}$
VI	23.30	28.81	15.14	**12.12**	2.02	3.08	6.96
ZH	14.85	19.32	11.49	**8.95**	1.82	2.84	6.22
AR	10.95	14.82	8.90	**6.92**	1.55	2.66	6.24
ES	10.48	13.48	8.92	**6.68**	1.56	2.53	6.11
KO	8.18	10.22	6.60	**4.99**	1.71	2.51	5.63
HI	6.93	8.93	5.51	**4.99**	1.52	2.36	6.05
Mean	12.45	15.93	9.43	7.44	1.70	2.66	6.20
SD	5.97	7.30	3.49	2.72	0.20	0.26	0.43

Table 5: Model-4 performance on L2 accent ($Test_{L2}$) and native accent ($Test_{L1}$, LS_{Clean}, LS_{Other}), compared with Baseline-I, Baseline-II, and Model-1. SD refers to the standard deviation.

models are fine-tuned on a domain that is different from LibriSpeech. That is, fine-tuning models on out-of-domain data will, and as we see here does, result in deterioration of performance on in-domain data. We also note that our Model-1's performance on LibriSpeech is worse than that of Baseline-II on both the 'Clean' (LS_{Clean}, native speech under quite recording environments), and 'Other' (LS_{Other}, both noisy environment and accented recordings), Dev and Test splits. This may be because LibriSpeech is mostly comprised of L1 data and the greater variability on our L2-ARCTIC Train set (24 non-native speakers in our Model-1 vs. 4 native speakers in Baseline-II).

Model-2 (speaker-independent, accent-dependent): While Model-1 mimics a situation where we have some training data from speakers that we serve (i.e., test on), this is rarely a realistic scenario. We instead switch to a speaker-independent (but still *accent-dependent*) setting, Split-2. We carry out four-fold cross-validation with the 24 speakers in the data, every time using 18 speakers (three speakers per accent) in Train[10] and six speakers in Test (one per accent). We report the average of the four folds/runs, along with standard deviation.

As Table 3 shows, Model-2 performance is consistent with Model-1. Our Model-2 outperforms the two baselines on both Dev and Test, reaching 9.96 WER on Test compared to 12.47 for Baseline-I and 15.96 for Baseline-II. These results demonstrate that fine-tuning with multiple accents improves the accented ASR system without access to test speaker data.

Model-3 (speaker- and accent-independent): To evaluate performance on *unseen* accents, we adopt a zero-shot strategy by removing one accent at a time from both Train and Dev sets and evaluating on the Test set of the removed accent, Split-3. To evaluate model performance on each accent, we conduct six runs in total with one accent removed at a time.

As Table 4 shows, fine-tuning on accented speech benefits unseen accents and speakers (Model-3 setting). All the multi-accent, zero-shot models outperform Baseline-I and Baseline-II, which means each of the six accents benefit from other accents through this process of transfer

[10] We use 10% of the utterances from these 18 speakers for development (Dev).

	VI	ZH	AR	ES	KO	HI
Baseline-I	23.30	14.85	10.95	10.48	8.18	6.93
VI-specific	12.12	13.62	13.01	9.95	8.55	9.62
ΔWER	-11.18	-1.23	2.06	-0.53	0.37	2.69
Δ%	-48.00	**-8.31**	18.84	-5.03	4.52	38.77
ZH-specific	20.37	8.95	11.42	9.79	6.82	10.91
ΔWER	-2.93	-5.90	0.47	-0.69	-1.36	3.98
Δ%	-12.58	-39.75	4.26	-6.62	**-16.67**	57.43
AR-specific	23.88	14.86	6.92	9.86	9.16	7.74
ΔWER	0.58	0.01	-4.03	-0.62	0.98	0.81
Δ%	2.47	0.07	-36.83	**-5.92**	11.94	11.69
ES-specific	20.71	13.99	11.00	6.68	7.92	8.66
ΔWER	-2.59	-0.86	0.05	-3.80	-0.26	1.73
Δ%	**-11.13**	-5.81	0.43	-36.23	-3.22	25.01
KO-specific	20.07	12.12	11.66	10.04	4.99	9.09
ΔWER	-3.23	-2.73	0.71	-0.44	-3.19	2.16
Δ%	-13.88	**-18.38**	6.45	-4.23	-39.04	31.17
HI-specific	26.18	18.39	13.51	11.90	10.72	4.99
ΔWER	2.88	3.54	2.56	1.42	2.54	-1.94
Δ%	**12.37**	23.82	23.35	13.55	31.01	-27.99

Table 6: Model-4 performance in the zero-shot setting. Bold fonts represent the accent whose WER drops the most in the zero-shot setting. For example, compared with Baseline-I, the VI-specific fine-tuning not only improves performance on VI (i.e., a drop in WER), but also improves on ZH despite ZH being the unseen accent. One notable pattern is that HI-specific fine-tuning only benefits HI-accented speech recognition while all the other fine-tuning hinder performance on the HI accent.

learning. Our results also show that, in absence of in-accent data, some unseen accents are easier for the model than others. For example, on $\text{Test}_{\text{zeroshot}}$, Vietnamese (VI) is the most challenging (with 18.81 WER) and Hindi (HI) is the least challenging (with only 6.67 WER).

	Test_{all}		$\text{Test}_{\text{zeroshot-speaker}}$	
L1	**Mean**	**SD**	**Mean**	**SD**
VI	12.67	0.38	14.28	4.87
ZH	9.65	0.31	11.26	3.03
AR	7.28	0.29	8.56	2.28
ES	6.95	0.26	7.76	3.99
KO	5.22	0.18	5.69	2.20
HI	5.27	0.11	5.79	1.12

Table 7: Model-5 performance on L2 accent. Test_{all} contains utterances by all speakers within each L1 whereas $\text{Test}_{\text{zeroshot-speaker}}$ contains utterances by a single speaker that is absent in the training phase. Mean refers to the average WER over four folds for each L1, and SD refers to the standard deviation.

4.3 Accent-Specific Models

We evaluate the accent-dependent performance by fine-tuning our models on a single type of L1-specific accent at a time.

Model-4 (speaker-dependent, accent-dependent): The model is fine-tuned with Split-4 data to identify any accent-dependent training impact on downstream performance, as well as an upper bound on performance when the model is optimized for a single accent. In addition to evaluating on L2-ARCTIC Test, we test the model on L1-ARCTIC Test and LibriSpeech as a means to identify any degradation on L1 English data.

As Table 5 shows, while the multi-accent model (Model-1) outperforms Baseline-I for all six accents, all of the accent-specific models (Model-4 setting) outperform Model-1 on the Test_{L2} setting despite the small amount of data (roughly five hours) used for fine-tuning each of the versions of Model-4. On average, Model-4 setting is two points WER better than Model-1. In addition, Model-4 type models (each of which is fine-tuned on one non-native accent) perform reasonably well

Model	Model output
Ref	at lake linderman i had one canoe very good peterborough canoe
VI	at LAY LINDEMAN i had one canoe very good PETERBORROUG CANOES A lake LNDER MAN i had one canoe very good BIET OF ROCK canoe
ZH	at lake LINGERMAN i had ONCE canoe very good PETERBROUGH canoe at lake LINERMAN i had one canoe very good PETERE BROUGHTA canoe
AR	at lake LUNDERBOGH i had one canoe very good BITTERBOROUGH canoe at lake LUNDERMAN i had one canoe very good BETTER BORT canoe
ES	at lake linderman i had one canoe a very good PETERBOURN canoe at lake linderman i had ONCE canoe very good PIERREBOROUGH canoe
KO	at lake linderman i had one canoe very good peterborough canoe at lake LINDEMAN i had ONCE canoe very good PITTEBRAUG canoe
HI	at lake LINDEMAN i had one canoe very good PETERBURGH canoe at lake linderman i had one canoe A very good PEACHERBROROU canoe

Table 8: Examples of transcription output of selected utterances from the Test set of Model-4 among all six L1s without a language model. Capitalized words indicate errors. We show samples from two speakers per accent.

on L1 data (Test_{L1}, LS_{Clean}, and LS_{Other}). Further, large accent-specific variability is observed across different model types on Test_{L2} (SD = $[2.72 - 7.30]$), compared with native counterparts such as Test_{L1} (SD = $[0.20 - 0.43]$). An interesting result is the apparent difficulty difference between different accents (HI and KO easiest, VI hardest), regardless of model types. We provide sample outputs from Model-4 in Table 8.

As shown in Table 6, we also perform accent-wise zero-shot evaluation. Results of this set of experiments reveal an interesting pattern: while fine-tuning on a single accent generally benefits *at least one other accent*, fine-tuning on the Hindi accent only benefits Hindi (the same accent) and hinders performance on *all the other accents*. Figure 3 and Figure 4 illustrate this observation.

Model-5 (speaker-independent and accent-dependent): This setup simulates a more realistic scenario where we target a single accent, without access to all speakers during development time. Thus, we use Split-5 data which mimics a speaker-independent setting. We cross-validate each L1 subset with one of the four speakers per fold. The hyper-parameters we use are those identified for Model-4. To evaluate the performance on each speaker, we conduct 24 folds in total with one speaker removed at a time, and report the average and standard deviation of the four folds per each accent.

As Table 7 shows, speaker-dependent variability is small for Test_{all} (SD = $[0.11 - 0.38]$) but large for $\text{Test}_{\text{zeroshot-speaker}}$ (SD = $[1.12 - 4.87]$). These results suggest that individual speaker's differences may play an important role in how much performance gain can be obtained by fine-tuning.[11]

[11] For those speakers whose TOEFL scores are known (Zhao et al., 2018), a strong negative correlation was observed between speaker-specific WERs of Baseline-I and speaker's TOEFL scores, $r(8) \approx -.77$, $p < .01$.

5 Conclusion

We demonstrated potential of developing accent-independent and accent-dependent models that improve non-native speech recognition simply by fine-tuning the pre-trained wav2vec 2.0 model on a small amount of labeled data. Both the multi- and single-accent models improve performance on L2 English speakers. However, each accent benefits differently: results of the multi-accent, zero-shot experiments suggest that transfer learning on accent is possible and single-accent models improve the most for the target L2 accents.

As to future work, while we chose a language model-free setting to focus specifically on wav2vec 2.0's acoustic capacity, comparison with language model decoding would be a useful direction to explore as a way to gauge any further potential improvements a language model can bring. In addition, finding the optimal combination of accented speech datasets when there is no available dataset for a target accent (Model-3) may constitute another interesting direction. Finally, although we have offered a number of sample transcriptions from one of our models, a thorough error analysis on each experiment would help advance the research into improving ASR models for non-native English speakers. Since L2 English speakers have specific accent characteristics influenced by their native languages, an error analysis focused on each language as well as on groups or families of languages will likely aid effective model development. Future directions could also investigate different strategies for developing ASR systems for challenging languages such as Vietnamese.

References

Alexei Baevski, Steffen Schneider, and Michael Auli. 2019. vq-wav2vec: Self-supervised learning of discrete speech representations. *arXiv preprint arXiv:1910.05453*.

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *arXiv preprint arXiv:2006.11477*.

Amy Bearman, Kelsey Josund, and Gawan Fiore. 2017. Accent conversion using artificial neural networks. Technical report, Stanford University, Tech. Rep, Tech. Rep.

William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. 2016. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4960–4964. IEEE.

Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. 2015. Attention-based models for speech recognition. In *Advances in neural information processing systems*, pages 577–585.

Yu-An Chung, Wei-Ning Hsu, Hao Tang, and James Glass. 2019. An unsupervised autoregressive model for speech representation learning. *arXiv preprint arXiv:1904.03240*.

David Crystal. 2003. *English as a global language*. Ernst Klett Sprachen.

Nilaksh Das, Sravan Bodapati, Monica Sunkara, Sundararajan Srinivasan, and Duen Horng Chau. 2021. Best of both worlds: Robust accented speech recognition with adversarial transfer learning. *arXiv preprint arXiv:2103.05834*.

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, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

Alex Graves and Navdeep Jaitly. 2014. Towards end-to-end speech recognition with recurrent neural networks. In *International conference on machine learning*, pages 1764–1772.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. 2020. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*.

Jingyong Hou, Pengcheng Guo, Sining Sun, Frank K Soong, Wenping Hu, and Lei Xie. 2019. Domain adversarial training for improving keyword spotting performance of esl speech. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8122–8126. IEEE.

Hu Hu, Xuesong Yang, Zeynab Raeesy, Jinxi Guo, Gokce Keskin, Harish Arsikere, Ariya Rastrow, Andreas Stolcke, and Roland Maas. 2021. Redat: Accent-invariant representation for end-to-end asr by domain adversarial training with relabeling. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6408–6412. IEEE.

Jacob Kahn, Morgane Rivière, Weiyi Zheng, Evgeny Kharitonov, Qiantong Xu, Pierre-Emmanuel Mazaré, Julien Karadayi, Vitaliy Liptchinsky, Ronan Collobert, Christian Fuegen, et al. 2020. Libri-light: A benchmark for asr with limited or no supervision. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7669–7673. IEEE.

John Kominek, Alan W Black, and Ver Ver. 2003. Cmu arctic databases for speech synthesis.

Shaoshi Ling, Yuzong Liu, Julian Salazar, and Katrin Kirchhoff. 2020. Deep contextualized acoustic representations for semi-supervised speech recognition. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6429–6433. IEEE.

Andy T Liu, Shu-wen Yang, Po-Han Chi, Po-chun Hsu, and Hung-yi Lee. 2020a. Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6419–6423. IEEE.

Songxiang Liu, Disong Wang, Yuewen Cao, Lifa Sun, Xixin Wu, Shiyin Kang, Zhiyong Wu, Xunying Liu, Dan Su, Dong Yu, et al. 2020b. End-to-end accent conversion without using native utterances. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6289–6293. IEEE.

Karen Livescu and James Glass. 2000. Lexical modeling of non-native speech for automatic speech recognition. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100)*, volume 3, pages 1683–1686. IEEE.

Marco Matassoni, Roberto Gretter, Daniele Falavigna, and Diego Giuliani. 2018. Non-native children speech recognition through transfer learning. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6229–6233. IEEE.

Tomohiro Nakatani. 2019. Improving transformer-based end-to-end speech recognition with connectionist temporal classification and language model integration. In *Proc. Interspeech 2019*.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE.

Tan Tien Ping. 2008. *Automatic speech recognition for non-native speakers*. Ph.D. thesis, Université Joseph-Fourier-Grenoble I.

Kacper Radzikowski, Le Wang, Osamu Yoshie, and Robert Nowak. 2021. Accent modification for speech recognition of non-native speakers using neural style transfer. *EURASIP Journal on Audio, Speech, and Music Processing*, 2021(1):1–10.

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. 2019. wav2vec: Unsupervised pre-training for speech recognition. *arXiv preprint arXiv:1904.05862*.

Xian Shi, Fan Yu, Yizhou Lu, Yuhao Liang, Qiangze Feng, Daliang Wang, Yanmin Qian, and Lei Xie. 2021. The accented english speech recognition challenge 2020: open datasets, tracks, baselines, results and methods. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6918–6922. IEEE.

Sining Sun, Ching-Feng Yeh, Mei-Yuh Hwang, Mari Ostendorf, and Lei Xie. 2018. Domain adversarial training for accented speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4854–4858. IEEE.

Thibault Viglino, Petr Motlicek, and Milos Cernak. 2019. End-to-end accented speech recognition. In *Interspeech*, pages 2140–2144.

Yongqiang Wang, Abdelrahman Mohamed, Due Le, Chunxi Liu, Alex Xiao, Jay Mahadeokar, Hongzhao Huang, Andros Tjandra, Xiaohui Zhang, Frank Zhang, et al. 2020. Transformer-based acoustic modeling for hybrid speech recognition. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6874–6878. IEEE.

Zhirong Wang, Tanja Schultz, and Alex Waibel. 2003. Comparison of acoustic model adaptation techniques on non-native speech. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, volume 1, pages I–I. IEEE.

Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R Hershey, and Tomoki Hayashi. 2017. Hybrid ctc/attention architecture for end-to-end speech recognition. *IEEE Journal of Selected Topics in Signal Processing*, 11(8):1240–1253.

Haihua Xu, Yerbolat Khassanov, Zhiping Zeng, Eng Siong Chng, Chongjia Ni, Bin Ma, Haizhou Li, et al. 2020. Independent language modeling architecture for end-to-end asr. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7059–7063. IEEE.

Qiantong Xu, Alexei Baevski, Tatiana Likhomanenko, Paden Tomasello, Alexis Conneau, Ronan Collobert, Gabriel Synnaeve, and Michael Auli. 2021. Self-training and pre-training are complementary for speech recognition. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3030–3034. IEEE.

Guanlong Zhao, Sinem Sonsaat, Alif O Silpachai, Ivana Lucic, Evgeny Chukharev-Hudilainen, John Levis, and Ricardo Gutierrez-Osuna. 2018. L2-arctic: A non-native english speech corpus. *Perception Sensing Instrumentation Lab*.

Orthographic Transliteration for Kabyle Speech Recognition

Chris Haberland
Mosaix AI
Palo Alto, California
crh2ke@virginia.edu

Ni Lao
Mosaix AI
Palo Alto, California
nlao@cs.cmu.edu

Abstract

Training on graphemes alone without phonemes simplifies the speech-to-text pipeline. However, models respond differently to training on graphemes of different writing systems. We investigate the impact of differences between Latin and Tifinagh orthographies on automatic speech recognition quality on a Kabyle Berber speech corpus. We train on a corpus represented in a Latin orthography marked for vowels and gemination and subsequently transliterate model output to a consonantal Tifinagh orthography not marked for these features, which results in 10% absolute improvement in word error rate over a model trained on the unmarked orthography. We find that this performance gain is primarily due to a reduced error rate for graphemes marked for vocalic and voiced consonantal phonemes. Our results suggest that speech-to-text corpora for languages with alternative defective orthographies may lead to better model quality by being fully marked for vowels and gemination.[1]

1 Introduction

Graphemic modeling units and their correspondence with the spoken word can vary between different language communities (Turki et al., 2016), and even a single language community may have multiple orthographic conventions for application in different contexts (Zitouni, 2014) (diglossia). Minority languages in particular have often undergone less standardization (Jaffe, 2000), contributing to a greater tendency to be written in multiple orthographies. Improving speech technologies to support minority and 'low-resource' languages and orthographies is crucial to ensuring their vitality and their users' access to information in the digital era (Cooper, 2019). Poor quality of low-resource language systems can compel users to interact with ASR systems in languages of which they are non-native, diminishing use of their native language. Furthermore, high error rates for a low-resource language ASR systems disadvantage monolingual speakers of the low-resource language that have a limited ability to switch to systems in more prevalent languages with better recognition quality.

Modern speech-to-text (S2T) models are trained on audio data paired with sequences of modeling units (Davel et al., 2015), which may be graphemes, phonemes, or other representations (Belinkov et al., 2019) that represent the linguistic constituents. Training models on phonemes constitutes a general paradigm in the creation of S2T systems (Yu et al., 2020a) especially in the context of low-resource languages (Besacier et al., 2014). Training on phonemes can be advantageous for decoding out-of-vocabulary words or words from an external language (Hu et al., 2019), but manual annotation of speech data can be prohibitively expensive for low-resource languages (Cooper, 2019).

ASR pipelines often include a component to automatically generate phoneme-based training data through grapheme to phoneme (G2P) conversions (Kubo and Bacchiani, 2020; Chen et al., 2019) by training supervised models (Rao et al., 2015; Jyothi and Hasegawa-Johnson, 2017; Arora et al., 2020) or constructing rule-based systems (Abbas and Asif, 2020a). Recently, there is a trend towards G2P conversion with minimal intervention and preparation to streamline the end-to-end learning process. Several systems have sought to streamline the G2P process using self-training (Hasegawa-Johnson et al., 2019), en-

[1] A repository of our work can be found at https://github.com/berbertranslit/berbertranslit

sembles of varying degrees of supervision (Yu et al., 2020b), and leveraging open dictionaries of higher-resource languages (Deri and Knight, 2016). For low-resource languages, training S2T systems with graphemes alone obviates the G2P step in the S2T pipeline and the need for language-specific expert annotations (Le et al., 2019). However, S2T models respond differently to training on graphemes of different writing systems.

In this paper, we the study the impact of graphemic vowel inclusion, gemination marking, and elision on S2T performance for Kabyle, a Berber language of northern Algeria. We chose to experiment with this language to augment the discussion surrounding orthographic choice on S2T quality that has been conducted primarily on Semitic languages that are comparatively more resourced, such as Arabic. While several previous studies (Alhanai, 2014; Alshayeji et al., 2019; Al-Anzi and AbuZeina, 2017) have demonstrated the effect of training and decoding using defective and non-defective orthographies separately, our study is the first to compare a neural speech model's performance between a) training and decoding in a defective orthography, and b) training on a non-defective orthography and decoding into its defective representation. Our study is also the first to analyze the nature of phonemic errors made by neural ASR models trained on a corpus in a defective and non-defective orthography to understand any systematic difference of types of errors made by models trained on these orthographies. The results demonstrate the importance of including vocalic graphemic inputs for improved S2T recognition of vowels and voiced consonants. To our knowledge, this result represents the first S2T system trained on a Tifinagh-encoded corpus of a Berber language.

2 Related Work

The investigation of orthographic choices on S2T system performance parallels the research on human language comprehension of written text. A significant body of research has sought to uncover how different G2P mappings across writing systems may predict reading level achievement and interactions with dyslexia (Daniels and Share, 2018; Rafat et al., 2019). For example, Law et al. (2018) assess the reading abilities of children diagnosed with dyslexia when taught a novel orthography consisting of new G2P mappings. Maroun et al. (2020) study the effect of diacritization and non-diacritization of dyslexic and non-dyslexic readers' processing of the Arabic script and found spelling knowledge of study participants to be the most significant predictor of processing speed.

S2T learning solely with graphemes has a long history (Eyben et al., 2009). More recently, Wang et al. (2018) report that the phonemic-graphemic performance gap closes when model architecture and hyperparameters are attuned to the specific data input. Rao and Sak (2017) found improved performance of graphemically trained models in multi-accented corpora and in trials of increased input data scale. Other work has tested derivatives of graphemes, such as bytes (Li et al., 2019), wordpieces (Rao and Sak, 2017), and context-dependent graphemes (i.e. chenones) (Le et al., 2019; Wang et al., 2020). Wang et al. (2020) achieved state-of-the-art results on English data with graphemically-derived modeling units for English.

Imputation of diacritics to augment defective model inputs has been, and continues to be, an active area of research (Schone, 2006; Ananthakrishnan et al., 2005; Alqahtani and Diab, 2019; Alqahtani et al., 2019; Darwish et al., 2020). Diacritic imputation systems are designed to help computational models resolve heterophonic homographs, or congruent graphemic sequences that have multiple phonemic interpretations, in orthographies that do not mark certain features. Sequences of this type are prevalent in consonantal writing systems, such as that used for Arabic, in which roughly one-third of tokens may be pronounced differently when not diacritized (Maroun and Hanley, 2017).

There has been work investigating diacritization's effect on speech modeling in languages that are written in defective orthographies, or those not marked for certain phonemes. Afify et al. (2005) used HMMs to demonstrate that training on voweled graphemes could increase performance over training on unvow-

elled graphemes on Arabic broadcast transcripts, even when decoding into unvowelled text. However, to the authors' knowledge, this has not been demonstrated in modern neural speech models. However, more recently, Alhanai (2014) showed that training neural acoustic models *and decoding* into voweled graphemes generally improved WER over unvowelled graphemes. Alsharhan and Ramsay (2019) pre-annotate training transcripts with phonetic information deduced from graphemic context with rules to improve system performance. Alshayeji et al. (2019) and Al-Anzi and AbuZeina (2017) compare diacritized and non-diacritized input with various S2T model architectures and hyperparameters and observe higher WER for diacritized trials, though they do not train on diacritized data and decode on non-diacritized data.

Augmenting inputs via transliteration has been shown to improve S2T systems or machine translation performance. Emond et al. (2018) transliterate model output as a post-process to improve the recognition of code-switched speech. Le and Sadat (2018) and Cho et al. (2020) model the G2P task as a neural sequence-to-sequence model and record improvements in named entity recognition and code-switched speech for Vietnamese and mixed Korean-Chinese scripts, respectively. While these studies use neural G2P models, rule-based systems are commonly developed for under-resourced languages (Ahmadi, 2019; Abbas and Asif, 2020b).

To date, there are limited efforts that apply neural speech models to Berber languages. OCR techniques have been applied to Tifinagh recently (Sadouk et al., 2017; Benaddy et al., 2019), and Lyes et al. (2019) produced a pronunciation dictionary for speech modeling of phonemes. However, to the best of our knowledge, the ASR research community has not documented the training of Berber S2T models aside from those produced from the CommonVoice initiative (Ardila et al., 2019) trained with a Latin-script corpus, although Zealouk et al. (2020) do describe a speech recognition system for Amazigh of Morocco.

3 The Kabyle Language and Berber Writing Systems

Kabyle is a Berber language spoken in northern Algeria that has historically been written in Latin, Arabic, and Tifinagh scripts. Contemporary Kabyle is most widely written in a Latin orthography popularized by the linguist Mouloud Mammeri in a 1976 grammar of the language, though the Arabic and Tifinagh scripts are still promoted among certain groups within Algeria society (Souag, 2019). Souag (2019) contends that the Latin script predominates over the others in modern usage.

The alphabetic Neo-Tifinagh orthograpies came into use after language planning initiatives for the Berber languages in the mid-twentieth century spearheaded by organizations such as Morocco's IRCAM (Amazigh), the Nigerien APT (Tuareg) (Blanco, 2014), and the Académie berbère (Kabyle) (Souag, 2019). The traditional, consonantal Tifinagh orthographies are not commonly used to write Kabyle. However, we transliterate Kabyle into a consonantal orthography to expand the incomplete literature on decoding into defective orthographies, which has primarily focused on Semitic languages. To our knowledge, no prior study has trained or decoded a speech model for a Berber language using a Tifinagh orthography.

We outline the fundamental differences between the Latin Kabyle orthography and the consonantal Tifinagh orthography: the first is that the Latin marks for gemination via digraphs, unlike the traditional Tifinagh. Some consonantal digraphs are spirantized with respect to their singleton counterparts (e.g. 'tt' from 't'). In the Latin orthography, these digraphs are phonemically "tense" and correlate with increased pronunciation length and register a fortis-lenis contrast, including devoicing. They are phonemically distinct from their singleton counterparts and can form minimal pairs (Elias, 2020).

The second fundamental difference is of vowel denotation. Although vowels are written in all contexts in Neo-Tifinagh orthographies, they are not marked save for word-final positions in the traditional Tifinagh orthographies (Elghamis, 2011; Savage, 2008). From the set of Tifinagh characters that may repre-

sent vowels, only '◌ⴰ' exclusively represents non-glide vowels (for 'a', 'ə'[2]), while '◌ⵓ' ('u') and '◌ⵉ' ('i') also represent semi-vowels ('w' and 'j', respectively). These latter two graphemes are analogous to the *matres lectionis* of Semitic language scripts (Posegay, 2020).

A final difference is that certain Tifinagh orthographies make use of ligatures that elide certain combinations of adjacent graphemes. The number of attested ligatures across the many varieties of traditional Tifinagh is vast (Savage, 2008) and most are not supported by Unicode[3]. We test the effect of ligatures by encoding those used in the Ahaggar orthography Elghamis (2011) as distinct characters in trial (1c) described in Section 5.

4 Approach

4.1 Mozilla CommonVoice

We use the original CommonVoice Kabyle corpus for all experiments[4]. The audio-transcript pairs from Mozilla's CommonVoice crowdsourced initiative (Ardila et al., 2019), which has collected data for over 54 languages at the time of writing. All corpora are released with train/dev/test subsets, and a unique speaker may appear in only a single set among each split. Most utterances are derived from Wikipedia, but some have been added by annotators through the language community's Pontoon page[5]. We removed special symbols and normalized Unicode characters of similar graphical appearance to ensure that characters intended to represent a single grapheme were treated as such[6].

4.2 Mozilla DeepSpeech

For S2T model training, we use Mozilla's DeepSpeech pipeline, which is based on the DeepSpeech framework (Hannun et al., 2014) and is maintained by a large community. After parameter tuning we found that the default hyperparameters worked well. For all experiments, we used models of 1024 hidden units and trained for 50 epochs, with a learning rate of .0001 and dropout of 0.3. We used batch sizes of 32, 16, and 16 for train, dev, and test sets, respectively. We used the default trigram settings for training the LM with KenLM (Heafield et al., 2013) in our experiments.

4.3 Transliterator

To convert the Latin-script CommonVoice corpus to the Tifinagh orthographies in our experiments, we use the Graph Transliterator Python package(Pue, 2019). This constructs a directed tree of ranked transition rules (e.g, **mm** -> ⴸ (not ⴸⴸ) because **mm** -> ⴸ ranks before **m** -> ⴸ) to convert between between Latin and Berber orthographies. We write rules for two distinct defective orthographies modelled after Elghamis (2011)'s description of the Ahaggar variant of Tiginagh - one with ligatures, and one without. In cases where multiple Unicode graphemes represent the same phonemes across Berber languages and orthographies (e.g. ⴽ, ⵯ), we opted to use the symbol closest to that described in Elghamis (2011). Heterophonic homographs in the Latin corpus remain as such in the transliterated Tifinagh (e.g. 'd' represents both 'd' and 'ð', and is transliterated as 'ⴷ' and not the IRCAM 'ⵔ'. All Kabyle phonemes that do not have distinct graphemes in the orthography described in Elghamis (2011) are represented with a corresponding Neo-Tifinagh symbol (e.g. č -> ⵛ, ṛ -> ⵕ).

Table 1: Kabyle CommonVoice Data Statistics

Split	Downloaded	Processed	Length
Train	37,056	35,715	35 hrs, 24 min
Dev	11,482	11,100	10 hrs, 52 min
Test	11,483	11,125	11 hrs, 42 min

4.4 Sequence Alignment

We sought to investigate which, and to what degree, phonemic classes are affected by different training orthographies. To facilitate this analysis, we required a tool to align the graphemic output sequences from the ASR systems, such that the aligned character pairs represented the audio data at the same time periods in the input data. Therefore, we conduct a phonemic confusion analysis from the

[2] We do not find attestations of '◌ⵉ' in the traditional Tifinagh orthographies described in Elghamis (2011), so we transliterate word-final 'e' (primarily in loanwords) as '◌ⴰ'

[3] https://www.unicode.org/charts/PDF/U2D30.pdf

[4] Accessed April 2020, 4th ed.

[5] https://pontoon.mozilla.org/projects/common-voice/

[6] E.g., ɛ, and ∈ were converted to ɛ (U+025B)

Table 2: Normalization and transliteration examples

Original	Normalized	Tifinagh Transliteration
D tasnareft taserdasit i yettreṣṣin deg Lezzayer.	**d tasnareft taserdasit i yettreṣṣin deg lezzayer**	ⴷ ⵜⴰⵙⵏⴰⵔⴻⴼⵜ ⵜⴰⵙⴻⵔⴷⴰⵙⵉⵜ ⵉ ⵢⵜⵜⵔⴻⵚⵚⵉⵏ ⴷⴻⴳ ⵍⴻⵣⵣⴰⵢⴻⵔ
Teĉĉiḍ iles-ik waqila?	**teĉĉiḍ iles ik waqila**	ⵜⴻⵛⵛⵉⴹ ⵉⵍⴻⵙ ⵉⴽ ⵡⴰⵇⵉⵍⴰ
Σerḍeɣ-t-id ad yekkes lxiq, yeẓẓel iḍarren.	**ɛerḍeɣ t id ad yekkes lxiq yeẓẓel iḍarren**	ⵄⴻⵔⴹⴻⵖ ⵜ ⵉⴷ ⴰⴷ ⵢⴻⴽⴽⴻⵙ ⵍⵅⵉⵇ ⵢⴻⵥⵥⴻⵍ ⵉⴹⴰⵔⵔⴻⵏ
Tawaɣit d lmeḥna d-yeɣḍel ṭrad ɣef tmurt.	**taɣaɣit d lmeḥna d yeɣḍel ṭrad ɣef tmurt**	ⵜⴰⵖⴰⵖⵉⵜ ⴷ ⵍⵎⴻⵃⵏⴰ ⴷ ⵢⴻⵖⴹⴻⵍ ⵟⵔⴰⴷ ⵖⴻⴼ ⵜⵎⵓⵔⵜ

Table 3: Modelling unit experiment (1c) input example. Note: ⵠ and ⵣ are stand-in single-character substitutions for ligatures that are not represented in Unicode and are not graphically representative of the traditional graphemes for these ligatures

Non-ligatured	ⵑⵠⵒⵔ	ⵚⵣⵠⵑ	ⵣⵒⵔ	ⵔⴰⵜ	ⵄⵑⵠ	ⵒⵜ	ⵜⵑⵑ	ⴰⵚ	ⵜⵎⵑⵜⵡⵜ
Ligatured	ⵒⵒⵔ	ⵚⵣⵠⵑ	ⵣⵒⵔ	ⵔⴰⵜ	ⵄⵒ	ⵒⵜ	ⵜⵑⵑ	ⴰⵚ	ⵜⵎⵠⵣ

graphemes with Sound-Class-Based Phonetic Alignment (SCA) List (2014). This was possible due to the high transparency, or unambiguous correspondence between graphemes to phonemes (Marjou, 2021) of the Kabyle Latin script. We use the *prog_align* function contained in the Lingpy package (List et al., 2019), which constructs a similarity matrix and applies a Neighbor-Joining algorithm (see Saitou and Nei (1987)) to construct a guide tree to successively align phonemic sequences. A dynamic programming routine finds a least-cost path through the matrix to align the two sequences according to similar sound classes. We find that this approach gives reliable alignment for phonemic sequences. We found no errors after manually inspecting a thousand aligned phoneme pairs[7].

5 Experimentation and Results

Now we present our result comparing S2T performance when training on orthographies of varying degrees of phonemic informativeness, and analyzing phonemic confusing using sequence alignment techniques.

5.1 Experiments

First, we test the hypothesis that training and testing upon an orthography unmarked for vowels, as opposed to marked, yields lower ASR word error rates. Experiment 1 compares the effect of training and testing upon the Latin-based orthography and transliterated Tifinagh orthography in a set of trials listed in Table 4 (1a-c). In 1a, the Latin corpus is used for training and testing. The outputs were evaluated against Latin gold utterances in the test split. In 1b, we train in the same manner, but test by applying a transliterator to convert the Latin test set into the consonantal Tifinagh orthography without ligatures. The corpus used to train the language model (LM) is composed of the transliterated utterances of the original corpus. In the third setup (1c), we repeat experiment 1b using a transliterator that models the ligatures described in Section 3. Examples of the ligatured Tifinagh are shown in Table 3.

Secondly, we test the hypothesis that learning from an orthography marked for vowels and decoding on an orthography unmarked for vowels results in lower word error rates compared to training and testing on either of the marked or unmarked orthographies alone. In experiment 2, we test the hypothesis that training on the plene (fully marked) Latin orthography and subsequently decoding into and testing against the defective Tifinagh orthography yields lower error rates compared to both training and testing on the Tifinagh orthography. We train all components on the Latin script and obtain Latin-script output for test utterances as in 1a. However, we then transliterate the output and test against gold utterances transliterated into Tifinagh, as in 1b. Because our main goal is to study the acoustic

[7] https://github.com/berbertranslit/berbertranslit

Table 4: The impact of orthography and language modeling. Group 1: trained and tested on the same orthography types. Group 2: Latin to Tifinagh transliteration at test time given a Latin model. Group 3: the same as Group 1 but without language modeling.

Exp.	Train Orthography	Transliteration	LM	Test Orthography	CER (%)	WER (%)
1a	Latin	no	yes	Latin	29.9	49.9
1b	Tifinagh	no	yes	Tifinagh	35.8	57.9
1c	Tifinagh (ligatured)	no	yes	Tifinagh (ligatured)	33.7	57.4
2	Latin	yes	yes	Tifinagh	29.7	47.4
3a	Latin	no	no	Latin	34.9	78.3
3b	Tifinagh	no	no	Tifinagh	38.8	77.9
3c	Latin	yes	no	Tifinagh	35.6	72.1

Table 5: Alignment of the same sentence produced by different models in Table 4. * indicates a missing space in the alignment. + indicates transliterated gold sequence in Tifinagh.

Group		Raw	Alignment (in IPA representation)
3a - Latin - Latin	Gold	yuweḍ ɣer lebɣi s	j u w ə ḍ ʁ ə r _ l ə b * ʁ * i _ s
	Pred	yuweḍ ɣaleb ɣ is	j u w ə ḍ ʁ a - * l ə b _ ʁ _ i * s
3b - Tifinagh - Tifinagh	Gold+	ⵢⵓⵡⴻⴹ ⵖⵔ ⵍⴻⴱⵉⵖ ⵙ	j w ḍ ʁ r _ l b ʁ j s
	Pred	ⵢⵓⵡⴻⴹ ⵍⴻⴱⵉ ⵙ	j w ḍ ʁ - * l b ʁ - s
3c - Latin - Tifinagh	Gold+	ⵢⵓⵡⴻⴹ ⵖⵔ ⵍⴻⴱⵉⵖ ⵙ	j w ḍ ʁ r _ l b * ʁ j s
	Pred	ⵢⵓⵡⴻⴹ ⵍⴻⴱ ⵉ ⵙ	j w ḍ ʁ - * l b ʁ - s

model and we do not want a small LM training corpus to negatively affect the experimental result, we build the LM in DeepSpeech on all train, dev, and test utterances of the normalized CommonVoice Kabyle Latin-script data for experiments 1 and 2.

Finally, we train the S2T model without a LM as a post-process to specifically understand the sensitivity of the neural speech component. Trials 3a-c replicate 1a-c, but do not apply LM post-processing to help understand the effect of our interventions on the neural ASR component.

5.2 Results

We report the results of all three sets of trials in Table 4. 1a and 1b show that the original Kabyle input encoded in the plene Latin orthography yields lower error rates than when training and testing on the transliterated Tifinagh alone (CER: -5.9%, WER: -8%). However, this reduction is less pronounced when the ligatured Tifinagh orthography is used (1c) (CER: -3.8%, WER: -7.5%).

Trial 2 exhibits improved recognition when training on the Latin orthography and subsequently transliterating to and testing against Tifinagh. This arrangement reduces CER by 0.2% and WER by 2.5% with respect to trial 1a in which the plene orthography was used for both training and testing. Compared to training and testing in the defective orthography (1b), 2 shows a 10.5% absolute decrease in WER and 6.1% absolute decrease in CER.

Trial 3 shows that, without the language model, the WER for training upon and testing against Latin orthography (3a) is greater than when using the Tifinagh orthography (3b) by 0.4%. However, the CER for the former procedure with respect to the latter is less by 3.9%, likely due to the increased difficulty of predicting more characters. Applying a Tifinagh tansliterator to the Latin trained model (3c) resulted in a WER reduction of 6.2% and 5.8% with respect to 3a and 3b. 3c exhibits an improved CER compared to the Tifinagh-only trial (3b) (-3.2%), although it is 0.7% higher when compared to the Latin-only trial (3a).

5.3 Phonemic Confusion Analysis

To understand the orthographies' effects on the speech model we conduct an analysis by alignment between the gold utterances and the predictions from experiments 3b and 3c. This analysis is inspired by recent studies by Kong et al. (2017), Alishahi et al. (2017) and Belinkov et al. (2019), to explore the nature of neural learning of phonemic information.

More specifically we use Lingpy (List et al., 2019) package to determine phone error rates as described in Section 4.4. We translate all graphemes of the gold utterances and their predicted counterparts into sequences of G2P IPA representations and tabulate phoneme class confusions using PHOIBLE's sound classes (Moran and McCloy, 2019). Table 5 shows example aligned sentences produced by this procedure. By analyzing the aligned utterances, we tabulate estimated confusions between the gold and predicted alignments.

We count phonemic disagreements between the models as a proportion of gold target contexts of the aligned matching phoneme. To understand which model achieves better performance for word-final vowel recognition that is denoted in the Tifinagh orthography, we analyze the counts of all gold contexts in which vowels or semi-vowels appear (always word-finally) against the counts of aligned model inferences at these contexts. Table 6 shows that the model trained on the Latin orthography and subsequently transliterated (3c) achieves higher recognition of the pure vowel grapheme compared to the model trained on the unvowelled traditional Tifinagh (3b).

Table 7 compares the errors across several different phonemic classes. We do not consider the 'continuant' and 'delayedRelease' features, as the distinction between allophonic and phonemic fricativity is difficult to determine for Kabyle from graphemes alone. Although the PHOIBLE database includes these features as 'syllabic', we tally counts for the 'approximate', 'sonorant', and 'dorsal', and 'periodic glottal source' features without 'syllabic' phonemes so as to better analyze the contribution of non-syllabic features. McNemar's asymptotic test with continuity correction Edwards (1948) affirms the significance of the difference between 3b and 3c ($P < 0.025$ for all features except the 'geminate' feature).

6 Discussion

Performance when training on plene inputs (3c) to decode word-final vowels improves when compared 3b in which intra-word vowels are hidden from the model. The results suggest that sonorous phonemes benefit more from model training on the voweled text. When only one model between 3b and 3c is correct, we see that 'approximate', 'sonorant', and 'period glottal' phonemes exhibit comparatively high disagreement, surpassed only by the phonemes with positive 'lateral' and 'syllabic' features. The model may share information across these features, and in particular, voicing. All of these features record higher recognition rates in the case of 3c. While the difference in error rates for sonorous and voiced consonants between 3b and 3c does not exactly trend according to the sonority hierarchy (Ladefoged and Johnson, 2014), the number of disagreements between the models does follow this trend. These findings suggest that the model in 3c is leveraging correlates of sonority for phoneme recognition (Figure 1).

A surprising contrast was discovered in the models' differential abilities in detecting coronal and dorsal consonants. We hypothesize that this difference is a function of the differing contexts that these sounds occur in relation to vowels and geminate consonants. The improvement in the 'spread glottis' feature between 3b and 3c is notable, though it is difficult to generalize given the low prevalence of graphemes representing phonemes possessing this feature.

Our study experiments with the DeepSpeech architecture using a single set of hyperparameters for a single data set and language. Future work can investigate the interactions of model architectures, hyperparameters, data scales, G2P mappings, and statistics of orthographic informativeness on S2T performance.

7 Conclusion

Our study is the first to document S2T performance on Tifinagh inputs and shows that the choice of orthography may be consequential for S2T systems trained on graphemes. We amplify findings of prior studies focused on Semitic languages by showing that a Berber S2T model intended to output unvowelled graphemes benefits from training on fully-featured inputs. Our research suggests that ensuring data inputs are fully-featured would improve ASR model quality for languages that conventionally use consonantal orthographies, like Syriac, Hebrew, Persian, and Arabic vernaculars.

Table 6: Comparison of model performance for different word-final vowels. The columns represents phoneme pairs (Tifinagh grapheme : Latin IPA). Trial 3c shows considerably higher recognition of vowels.

	ⴰ : a/ə	ⵉ : i (j)	ⵓ : u/ (w)	All Vowels
The number of word-final vowels in gold	7,430	6,557	1,341	15,328
C_w: The portion (%) of all word-final phonemes	11.7%	10.3%	2.1%	13.0%
C_2: The portion (%) of C_w either 3b (x)or 3c is correct	23.7%	28.1%	30.4%	26.2%
C_3: Both 3b and 3c are incorrect	38.2%	46.8%	34.9%	41.6%
C_{3b}: The portion (%) of C_2 for which 3b is correct	18.5%	13.2%	13.7%	15.6%
C_{3c}: The portion (%) of C_2 for which 3c is correct	**81.5%**	**86.8%**	**86.3%**	**84.5%**

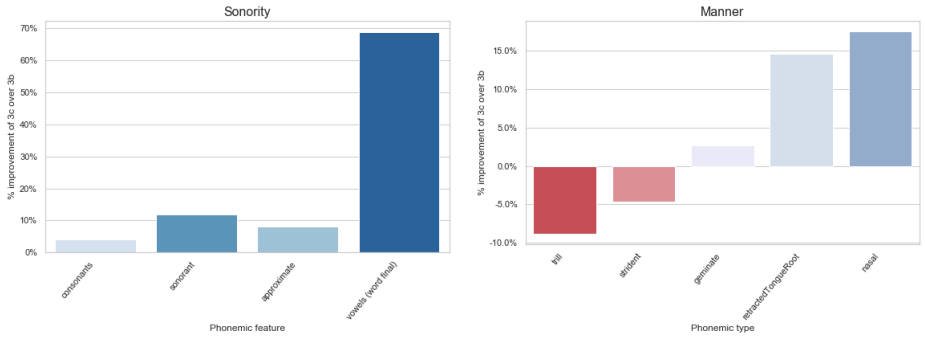


Figure 1: Comparison of the relative error difference between 3b and 3c.

Table 7: Comparison of model performance for different phonemic features. C_p represents the portion (%) of G2P mappings the feature comprises of the total number of G2P mappings in the corpus. See the definition of C_2, C_3, C_{3b} and C_{3c} in Table 6. 3c is correct for more disagreements for all features except for the coronal, strident, and trill features. We use McNemar's asymptotic test with continuity correction Edwards (1948) to test the null hypothesis that there is no difference between the performance of C_{3b} and C_{3c} with respect to different sound classes. χ^2_1 values are particularly high for voiced and syllabic phonemes. We bold the higher between C_{3b} and C_{3c} when $\chi^2_1 > 18.5$ (corresponding to P=0.001).

	C_p	C_2	C_3	C_{3b}	C_{3c}	χ^2_1
consonants	53.1%	16.6%	29.7%	47.9%	**52.1%**	38.9
sonorant (- syllabic)	24.0%	18.1%	26.2%	44.1%	**55.9%**	151.4
approximate (- syllabic)	12.6%	18.6%	28.2%	45.9%	**54.0%**	38.2
nasal	11.5%	17.6%	24.1%	42.0%	**58.0%**	130.1
retracted tongue root	2.1%	16.7%	60.0%	45.8%	54.2%	6.0
labial	11.7%	16.1%	49.8%	35.0%	**65.0%**	26.3
labiodental	1.5%	17.8%	30.3%	40.7%	**59.3%**	23.7
coronal	37.1%	16.4%	29.1%	**51.9%**	48.1%	22.3
strident	8.0%	10.5%	33.5%	**53.8%**	46.2%	12.3
lateral	4.3%	18.6%	30.0%	47.4%	52.6%	5.3
geminate	8.6%	9.0%	56.8%*	49.6%	50.4%	0.13
trill	5.0%	16.3%	30.7%	**55.7%**	44.3%	26.0
dorsal (- syllabic)	11.8%	17.7%	28.7%	38.2%	**61.8%**	294.6
periodic glottal (voiced) (- syllabic)	36.4%	18.5%	29.2%	42.2%	**57.8%**	407.9
spread glottis	0.4%	20.3%	47.1%	34.6%	**65.4%**	18.8
syllabic (vowels) (word-final)	6.1%	26.2%	41.6%	15.6%	**84.4%**	1902.8

References

Muhammad Raihan Abbas and Dr Khadim Hussain Asif. 2020a. Punjabi to iso 15919 and roman transliteration with phonetic rectification. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 19(2):1–20.

Muhammad Raihan Abbas and Dr. Khadim Hussain Asif. 2020b. Punjabi to iso 15919 and roman transliteration with phonetic rectification. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 19(2).

Mohamed Afify, Long Nguyen, Bing Xiang, Sherif Abdou, and John Makhoul. 2005. Recent progress in arabic broadcast news transcription at bbn. In *Ninth European Conference on Speech Communication and Technology*.

Sina Ahmadi. 2019. A rule-based kurdish text transliteration system. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 18(2):1–8.

Fawaz S Al-Anzi and Dia AbuZeina. 2017. The effect of diacritization on arabic speech recogntion. In *2017 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, pages 1–5. IEEE.

Tuka Tuka Waddah Talib Ali Al Hanai Alhanai. 2014. *Lexical and Language Modeling of Diacritics and Morphemes in Arabic Automatic Speech Recognition*. Ph.D. thesis, Massachusetts Institute of Technology.

Afra Alishahi, Marie Barking, and Grzegorz Chrupała. 2017. Encoding of phonology in a recurrent neural model of grounded speech. *arXiv preprint arXiv:1706.03815*.

Sawsan Alqahtani and Mona Diab. 2019. Investigating input and output units in diacritic restoration. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pages 811–817. IEEE.

Sawsan Alqahtani, Ajay Mishra, and Mona Diab. 2019. Efficient convolutional neural networks for diacritic restoration. *arXiv preprint arXiv:1912.06900*.

Eiman Alsharhan and Allan Ramsay. 2019. Improved arabic speech recognition system through the automatic generation of fine-grained phonetic transcriptions. *Information Processing & Management*, 56(2):343–353.

Mohammad Alshayeji, Sari Sultan, et al. 2019. Diacritics effect on arabic speech recognition. *Arabian Journal for Science and Engineering*, 44(11):9043–9056.

Sankaranarayanan Ananthakrishnan, Shrikanth Narayanan, and Srinivas Bangalore. 2005. Automatic diacritization of arabic transcripts for automatic speech recognition. In *Proceedings of the 4th International Conference on Natural Language Processing*, pages 47–54.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M Tyers, and Gregor Weber. 2019. Common voice: A massively-multilingual speech corpus. *arXiv preprint arXiv:1912.06670*.

Aryaman Arora, Luke Gessler, and Nathan Schneider. 2020. Supervised grapheme-to-phoneme conversion of orthographic schwas in hindi and punjabi. *arXiv preprint arXiv:2004.10353*.

Yonatan Belinkov, Ahmed Ali, and James Glass. 2019. Analyzing phonetic and graphemic representations in end-to-end automatic speech recognition. *arXiv preprint arXiv:1907.04224*.

Mohamed Benaddy, Othmane El Meslouhi, Youssef Es-saady, and Mustapha Kardouchi. 2019. Handwritten tifinagh characters recognition using deep convolutional neural networks. *Sensing and Imaging*, 20(1):9.

Laurent Besacier, Etienne Barnard, Alexey Karpov, and Tanja Schultz. 2014. Automatic speech recognition for under-resourced languages: A survey. *Speech Communication*, 56:85–100.

Juan Luis Blanco. 2014. Tifinagh & the ircam: Explorations in cursiveness and bicameralism in the tifinagh script. *Unpublished Dissertation, University of Reading*.

Zhehuai Chen, Mahaveer Jain, Yongqiang Wang, Michael L Seltzer, and Christian Fuegen. 2019. Joint grapheme and phoneme embeddings for contextual end-to-end asr. In *INTERSPEECH*, pages 3490–3494.

Won Ik Cho, Seok Min Kim, and Nam Soo Kim. 2020. Towards an efficient code-mixed grapheme-to-phoneme conversion in an agglutinative language: A case study on to-korean transliteration. In *Proceedings of the The 4th Workshop on Computational Approaches to Code Switching*, pages 65–70.

Erica Lindsay Cooper. 2019. *Text-to-speech synthesis using found data for low-resource languages*. Ph.D. thesis, Columbia University.

Peter T Daniels and David L Share. 2018. Writing system variation and its consequences for reading and dyslexia. *Scientific Studies of Reading*, 22(1):101–116.

Kareem Darwish, Ahmed Abdelali, Hamdy Mubarak, and Mohamed Eldesouki. 2020. Arabic diacritic recovery using a feature-rich bilstm model. *arXiv preprint arXiv:2002.01207*.

Marelie Davel, Etienne Barnard, Charl van Heerden, William Hartmann, Damianos Karakos, Richard Schwartz, and Stavros Tsakalidis. 2015. Exploring minimal pronunciation modeling for low resource languages. In *Sixteenth Annual Conference of the International Speech Communication Association*.

Aliya Deri and Kevin Knight. 2016. Grapheme-to-phoneme models for (almost) any language. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 399–408.

Allen L Edwards. 1948. Note on the "correction for continuity" in testing the significance of the difference between correlated proportions. *Psychometrika*, 13(3):185–187.

Ramada Elghamis. 2011. Le tifinagh au niger contemporain: Étude sur l'écriture indigène des touaregs. *Unpublished PhD Thesis, Leiden: Universiteit Leiden*.

Alexander Elias. 2020. Kabyle" double" consonants: Long or strong?

Jesse Emond, Bhuvana Ramabhadran, Brian Roark, Pedro Moreno, and Min Ma. 2018. Transliteration based approaches to improve code-switched speech recognition performance. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 448–455. IEEE.

Florian Eyben, Martin Wöllmer, Björn Schuller, and Alex Graves. 2009. From speech to letters-using a novel neural network architecture for grapheme based asr. In *2009 IEEE Workshop on Automatic Speech Recognition & Understanding*, pages 376–380. IEEE.

Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. 2014. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*.

Mark Hasegawa-Johnson, Camille Goudeseune, and Gina-Anne Levow. 2019. Fast transcription of speech in low-resource languages. *arXiv preprint arXiv:1909.07285*.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 690–696, Sofia, Bulgaria. Association for Computational Linguistics.

Ke Hu, Antoine Bruguier, Tara N Sainath, Rohit Prabhavalkar, and Golan Pundak. 2019. Phoneme-based contextualization for cross-lingual speech recognition in end-to-end models. *arXiv preprint arXiv:1906.09292*.

Alexandra Jaffe. 2000. Introduction: Non-standard orthography and non-standard speech. *Journal of sociolinguistics*, 4(4):497–513.

Preethi Jyothi and Mark Hasegawa-Johnson. 2017. Low-resource grapheme-to-phoneme conversion using recurrent neural networks. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5030–5034. IEEE.

Xiang Kong, Jeung-Yoon Choi, and Stefanie Shattuck-Hufnagel. 2017. Evaluating automatic speech recognition systems in comparison with human perception results using distinctive feature measures. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*.

Yotaro Kubo and Michiel Bacchiani. 2020. Joint phoneme-grapheme model for end-to-end speech recognition. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6119–6123. IEEE.

Peter Ladefoged and Keith Johnson. 2014. *A course in phonetics*. Nelson Education.

Jeremy M Law, Astrid De Vos, Jolijn Vanderauwera, Jan Wouters, Pol Ghesquière, and Maaike Vandermosten. 2018. Grapheme-phoneme learning in an unknown orthography: A study in typical reading and dyslexic children. *Frontiers in psychology*, 9:1393.

Duc Le, Xiaohui Zhang, Weiyi Zheng, Christian Fügen, Geoffrey Zweig, and Michael L Seltzer. 2019. From senones to chenones: Tied context-dependent graphemes for hybrid speech recognition. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 457–464. IEEE.

Ngoc Tan Le and Fatiha Sadat. 2018. Low-resource machine transliteration using recurrent neural networks of asian languages. In *Proceedings of the Seventh Named Entities Workshop*, pages 95–100.

Bo Li, Yu Zhang, Tara Sainath, Yonghui Wu, and William Chan. 2019. Bytes are all you need: End-to-end multilingual speech recognition and synthesis with bytes. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5621–5625. IEEE.

Johann-Mattis List. 2014. *Sequence comparison in historical linguistics*. Ph.D. thesis, Düsseldorf University Press.

Johann-Mattis List, Simon Greenhill, Tiago Tresoldi, and Robert Forkel. 2019. Lingpy. a python library for quantitative tasks in historical linguistics.

Demri Lyes, Falek Leila, and Teffahi Hocine. 2019. Building a pronunciation dictionary for the kabyle language. In *International Conference on Speech and Computer*, pages 309–316. Springer.

Xavier Marjou. 2021. OTEANN: Estimating the transparency of orthographies with an artificial neural network. In *Proceedings of the Third Workshop on Computational Typology and Multilingual NLP*, pages 1–9, Online. Association for Computational Linguistics.

Lateefeh Maroun, Raphiq Ibrahim, and Zohar Eviatar. 2020. Visual and orthographic processing in arabic word recognition among dyslexic and typical readers. *Writing Systems Research*, pages 1–17.

Maryse Maroun and J Richard Hanley. 2017. Diacritics improve comprehension of the arabic script by providing access to the meanings of heterophonic homographs. *Reading and Writing*, 30(2):319–335.

Steven Moran and Daniel McCloy, editors. 2019. *PHOIBLE 2.0*. Max Planck Institute for the Science of Human History, Jena.

Nick Posegay. 2020. Connecting the dots: The shared phonological tradition in syriac, arabic, and hebrew vocalisation. *Studies in Semitic Vocalisation and Reading Traditions*, page 191.

A. Sean Pue. 2019. Graph transliterator: A graph-based transliteration tool. *Journal of Open Source Software*, 4(44):1717.

Yasaman Rafat, Veronica Whitford, Marc Joanisse, Mercedeh Mohaghegh, Natasha Swiderski, Sarah Cornwell, Celina Valdivia, Nasim Fakoornia, Riham Hafez, Parastoo Nasrollahzadeh, et al. 2019. First language orthography influences second language speech during reading: Evidence from highly proficient korean-english bilinguals. In *Proceedings of the International Symposium on Monolingual and Bilingual Speech*, pages 100–107.

Kanishka Rao, Fuchun Peng, Haşim Sak, and Françoise Beaufays. 2015. Grapheme-to-phoneme conversion using long short-term memory recurrent neural networks. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4225–4229. IEEE.

Kanishka Rao and Haşim Sak. 2017. Multi-accent speech recognition with hierarchical grapheme based models. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 4815–4819. IEEE.

Lamyaa Sadouk, Taoufiq Gadi, and El Hassan Essoufi. 2017. Handwritten tifinagh character recognition using deep learning architectures. In *Proceedings of the 1st International Conference on Internet of Things and Machine Learning*, pages 1–11.

Naruya Saitou and Masatoshi Nei. 1987. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular biology and evolution*, 4(4):406–425.

Andrew Savage. 2008. Writing tuareg—the three script options. *International journal of the sociology of language*, 2008(192):5–13.

Patrick Schone. 2006. Low-resource autodiacritization of abjads for speech keyword search. In *Ninth International Conference on Spoken Language Processing*.

Lameen Souag. 2019. Kabyle in arabic script: A history without standardisation. *Creating Standards*, page 273.

Houcemeddine Turki, Emad Adel, Tariq Daouda, and Nassim Regragui. 2016. A conventional orthography for maghrebi arabic. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC), Portoroz, Slovenia*.

Yongqiang Wang, Abdelrahman Mohamed, Due Le, Chunxi Liu, Alex Xiao, Jay Mahadeokar, Hongzhao Huang, Andros Tjandra, Xiaohui Zhang, Frank Zhang, et al. 2020. Transformer-based acoustic modeling for hybrid speech recognition. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6874–6878. IEEE.

Yu Wang, Xie Chen, Mark JF Gales, Anton Ragni, and Jeremy Heng Meng Wong. 2018. Phonetic and graphemic systems for multi-genre broadcast transcription. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5899–5903. IEEE.

Xiang Yu, Ngoc Thang Vu, and Jonas Kuhn. 2020a. Ensemble self-training for low-resource languages: grapheme-to-phoneme conversion and morphological inflection. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 70–78.

Xiang Yu, Ngoc Thang Vu, and Jonas Kuhn. 2020b. Ensemble self-training for low-resource languages: Grapheme-to-phoneme conversion and morphological inflection. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 70–78, Online. Association for Computational Linguistics.

Ouissam Zealouk, Mohamed Hamidi, Hassan Satori, and Khalid Satori. 2020. Amazigh digits

speech recognition system under noise car environment. In *Embedded Systems and Artificial Intelligence*, pages 421–428. Springer.

Imed Zitouni. 2014. *Natural language processing of semitic languages*. Springer.

Machine Translation for Zero and Low-resourced Dialects using a New Extended Version of the Dialectal Parallel Corpus (Padic v2.0)

Mohamed Lichouri
Algiers, Algeria
medlichouri@gmail.com

Mourad Abbas
High Council of Arabic Language
Algiers, Algeria
abb.mourad@gmail.com

Abstract

In this paper we present a set of experiments performing machine translation related to low-resourced Arabic dialects in addition to a zero-resourced dialect (Berber). For this, we extended the parallel PADIC corpus by adding the Berber dialect corpus and translating manually more than 6000 Arabic sentences. We applied both Rule-based Machine Translation (RBMT) and Statistical Machine Translation (SMT) with and without a transliteration process. The average overall BLEU score is 42.68% with RBMT and 61.94% with SMT.

1 Introduction

Over the past years, research has seen remarkable progress on dialectal processing of the Arab region (Darwish et al., 2021), like dialect identification in text (Abbas et al., 2019; Lichouri and Abbas, 2020; Lichouri et al., 2021) and speech (Ali et al., 2021). This can be considered as a big challenge since all Arabic dialects have been spoken in the past and rarely written unlike Modern Standard Arabic (MSA). This makes processing very difficult in the absence of needed resources. This challenge is multiplied when it comes to deal with certain vernacular dialects (Lichouri et al., 2018), because they are not considered as Arabic dialects due to the obvious difference with Arabic on one side, and that they have never been written on the other side, which is the case of Berber dialects.

In this paper, we introduce PADIC v2.0, a recent version that we extended from PADIC v1.0 [1] (Meftouh et al., 2015), enriching it with new parallel texts related to a zero-resourced Berber dialect: Kabyle. To our knowledge, this is the first time that resources are developed for such a vernacular, zero-resourced dialect, and devoted to NLP and particularly machine translation. The first study that seems to be necessary and obvious to do is calculating the closeness between Berber dialect and the other Arabic dialects, Maghrebi and Levantine ones. As a natural extension to the previous studies that used PADIC (Harrat et al., 2014, 2015; Meftouh et al., 2015), we focus in this paper mainly on experiments of machine translation between Berber (Kabyle) Dialect and Arabic (MSA), as well as between the remaining Arabic dialects. The rest of this article is organized as follows, we first present related work in section 2. In section 3, we describe how we enriched PADIC corpus, followed by measuring distances between the different dialects. In section 4, we present the evaluation methods and the experimental results, and finally, we conclude in section 5.

2 Related Work

Low-resource and zero-resource languages are considerably lacking in works especially on Machine Translation (MT). For instance, for Arabic Language and its dialects, most of the work done on MT, focused on translation into English, as in (Sawaf, 2010) where an hybrid approach between rule-based and statistical methods was presented. The authors evaluated their approach on the NIST MT08 WB Arabic dataset comprising MSA and 15 colloquial Arabic dialects from almost all the Arab countries (except Algeria). In (Salloum and Habash, 2011), the authors proposed a technique to solve the problem posed by out-of-vocabulary (OOV) words and low frequency words in Arabic-English SMT. For that they adopted a paraphrasing approach of (OOV) words in Arabi Dialectal Text to produce Modern Standard Arabic (MSA) paraphrases of dialectal Arabic that are input to a phrase-based SMT system. This approach permitted the authors to implement Elissa which is

[1] PADIC v1.0 is a parallel Arabic multi-dialectal textual corpus composed of six Arabic dialects: Syrian, Tunisian, Moroccan, Palestinian and two Algerian dialects (of Algiers and Annaba cities), in addition to MSA.
https://sites.google.com/site/torjmanepnr/Home
https://sourceforge.net/projects/padic/

Arabic dialect into English Translator by pivoting on MSA (Salloum and Habash, 2012, 2013). (Zbib et al., 2012) conducted MT for (MSA-English), (Levantine-English), and (Egyptian-English). The authors found surprisingly that translating from Egyptian and Levantine dialects into English outperformed the couple of language (MSA-English) by 6.3 and 7.0 of BLEU, respectively. (Sajjad et al., 2013) attempted to narrow down the gap between Egyptian and MSA by applying an automatic character-level transformational model that changes Egyptian to a format similar to MSA, which reduced the out-of-vocabulary (OOV) words from 5.2% to 2.6% and gives a gain of 1.87 BLEU points. For Iraqi Dialect, in order to resolve the lack of dialectal parallel data, authors presented in (Kirchhoff et al., 2015) how they extracted parallel data from out-of-domain corpora related to different Arabic dialects and MSA. By applying deep neural network on Machine Translation, (Zoph et al., 2016) presented an approach based on transfer learning for the benefit of low-resource languages. In another context, (Almahairi et al., 2016) conducted experiments on Arabic Neural Machine Translation (NMT) and have concluded that in spite of the big need of tremendous amount of data for NMT, in comparison to Phrase-based Statistical Machine Translation, the NMT system outperforms the statistical one in case of an out-of-domain test set, making it attractive for real-world deployment. A comparison between statistical and NMT was conducted in (Guellil et al., 2017) for MSA and one of its dialects that had been extracted from PADIC corpus. Another study presented in (Alrajeh, 2018), having the same objective as that mentioned in (Guellil et al., 2017), which is comparing between phrase-based SMT and NMT, has been reached using three parallel MSA-ENG corpora: UN, ISI and Ummah. Their findings show that tuning a model trained on the whole data using a small high quality corpus like Ummah gives a substantial improvement and that training a neural system with a small Arabic-English corpus is competitive to a traditional phrase-based system. Another aspect that is not taken into account by most current models is that a sentence can have multiple translations. For this, as to solve the problem of this kind of variation in parallel corpus, (Schulz et al., 2018) applied a deep generative model of machine translation which incorporates a chain of latent variables, in order to account for local lexical and syntactic variation in parallel corpora.

3 Description of PADIC v2.0

The difference between PADIC v1.0 and PADIC v2.0 is that PADIC v2.0 has been enriched with Kabyle (an Algerian zero-resourced dialect). Kabyle is one of the Berber dialects; it is a branch of the Afro-Asiatic language phylum which covers parts of North Africa, stretching from Morocco to Yemen, including Libya, Egypt and Somalia. In the following, we will present how we developed PADIC v2.0, as well as the linguistic similarities between Arabic and Berber that can be on all levels: phonology, morphology, syntax, and lexicon.

3.1 Enrichment of PADIC with Berber Dialect

We solicited a couple of native speakers of Kabyle, a variant of Berber dialect, from Tizi-Ouzou city. These native speakers translated the 6400 sentences of PADIC from Algiers dialect (ALG), writing these sentences with Arabic letters. Hence, the new PADIC version is composed of seven Arabic dialects: ALG (Algiers), ANB (Annaba), TUN (Tunisia), PAL (Palestine), SYR (Syria), MOR (Moroccan), and KAB (Kabyle), in addition to MSA.

3.2 Distances between Dialects

In order to quantify the closeness between the studied dialects, we used a set of distances belonging to five measure classes[2]:

Edit based: Hamming, MLIPNS, Levenshtein, Damerau-Levenshtein, Jaro-Winkler, Strcmp95, Needleman-Wunsch, Gotoh and Smith-Waterman (Navarro, 2001).

Token based: Jaccard index, Overlap coefficient and Cosine similarity.

Sequence based: Longest common substring similarity and Ratcliff-Obershelp similarity.

Phonetic: MRA and Editex.

Compression NCD-based: BZ2, LZMA and ZLib.

The choice of these measures is explained by the fact that each of them has its own calculation algorithms and therefore each has a specific

[2]https://pypi.org/project/textdistance/

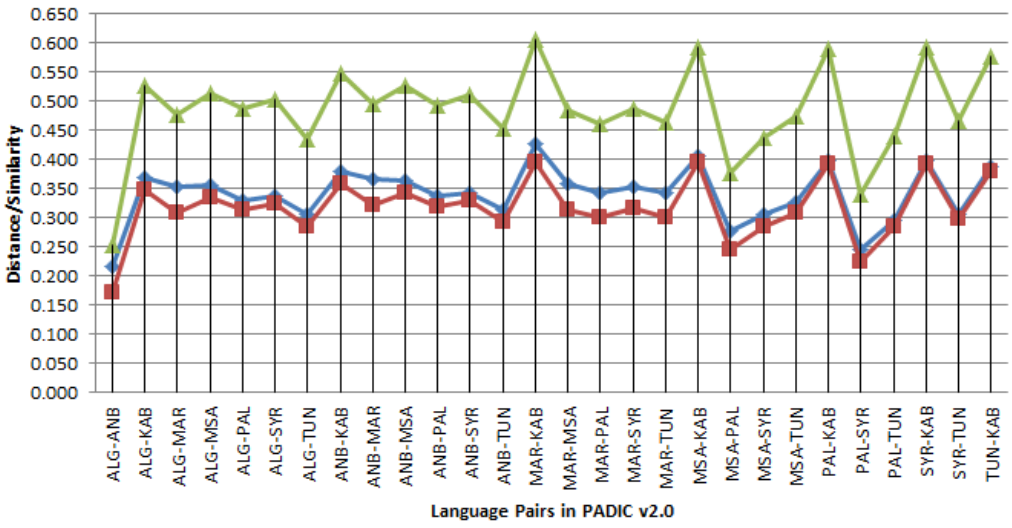


Figure 1: Distance Measures between Language Pairs in PADIC v2.0 by Compression NCD-based metric(0=Equal,1=Different). Metrics Bz2-NCD (Blue), LzMA-NCD (Red) and zLIB-NCD (Green).

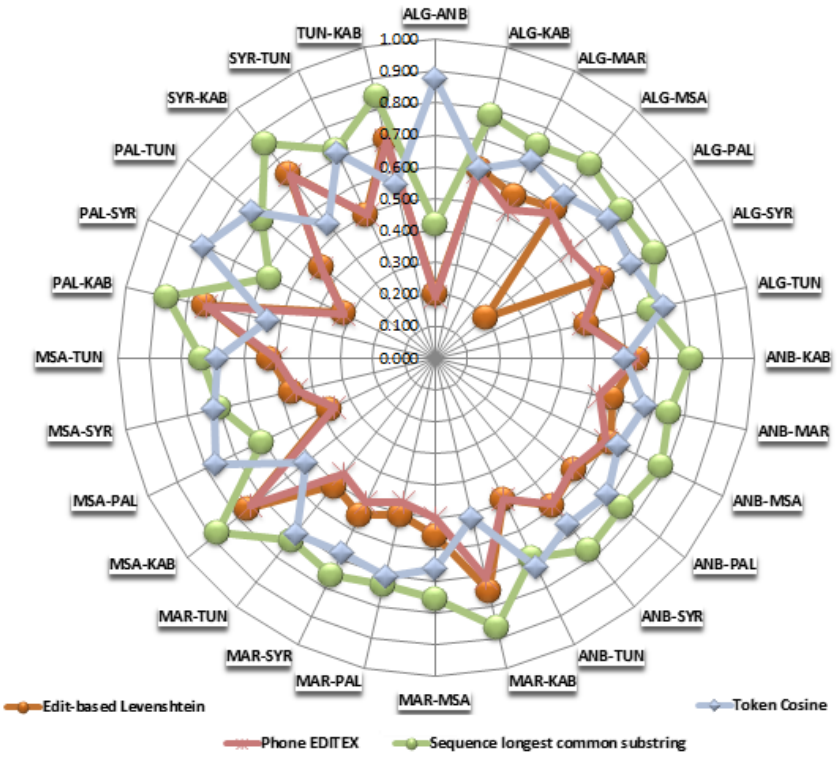


Figure 2: A sample of Distance Measures between Language Pairs in PADIC v2.0 by Edit-based, Phonetic, Sequence-based and Token-based metrics (0=Equal,1=Different)

purpose to deploy[3]. Because of the variation of these different measures, we used the normalized similarity for sequences (Vitányi, 2011), that returns a float between **0** and **1** (Cilibrasi and Vitányi, 2005), where **1** means totally different, and **0** means equal[4]. Based on Figure 1, we can see that the results obtained using Bz2-NCD and LzMA-NCD are very close, and 0.15 lower than zLIB-NCD. Note that the three curves have relatively a similar behavior, this reinforces the differences and similarities that actually exist between these dialects.

[3]https://www.kdnuggets.com/2019/01/comparison-text-distance-metrics.html

[4]https://articles.orsinium.dev/p/notes-other/ncd/

	ALG	ANB	KAB	MOR	PAL	SYR	TUN
From MSA (Simple)	65.93	65.93	62.91	71.24	73.72	71.76	71.67
From MSA (Translit)	09.56	09.63	09.44	09.63	09.67	09.41	09.63
To MSA (Simple)	69.56	70.29	64.06	80.14	72.12	64.63	57.56
To MSA (Translit)	15.01	16.87	33.81	14.56	39.32	16.65	30.40

Table 1: Comparison results of Rule-based MT for PADIC v2.0 from/to MSA with and without transliteration process.

	ALG	ANB	KAB	MOR	PAL	SYR	TUN
From MSA (Simple)	52.51	47.55	35.52	60.36	81.12	70.39	59.91
From MSA (Translit)	57.66	54.63	44.89	65.52	87.35	73.71	68.81
To MSA (Simple)	71.71	70.46	25.53	66.53	83.05	69.45	72.16
To MSA (Translit)	51.62	51.37	25.89	63.84	88.96	69.97	63.96

Table 2: Comparison results of SMT for PADIC v2.0 from/to MSA with and without transliteration process.

For example, Algiers dialect (ALG) is the closest to the Algerian Annaba's dialect (ANB), Which is very reasonable and expected, since these two dialects are spoken in the same country and share up to 60% of words (Meftouh et al., 2015). However, some results are unforeseen if one takes into account the geographical parameter. The appropriate example for this case, is the pairs of Arabic dialects TUN/ALG and TUN/PAL. Indeed, Tunisian has small distances with both Algiers dialect and Palestinian. The closeness with ALG is understandable because Tunisia borders Algeria, which is not the case for Palestine. Another interesting and unexpected result is that Moroccan dialect (MOR) is closer to Palestinian than Tunisian or Algerian, though Morocco borders Algeria. For Levantine dialects, Syrian is close to Palestinian, which is not surprising because of the geographical proximity, whereas Palestinian is closer to MSA than Syrian. For the Berber variant (Kabyle), it is clearly shown in Figure 1, that it has the farthest distance with all the Arabic dialects.

4 Experiments and Results

We applied two well-known MT approaches using PADIC v2.0: Rule Based MT (RBMT) and Statistical MT (SMT). We decided to use two versions of PADIC, one with Arabic letters, and the other one by applying Buckwalter transliteration [5]. The results are evaluated using BLEU score [6].

4.1 Rule Based MT

For achieving a simple rule based machine translation, we adopted the same model used in in the work by Niyongabo & College [7]. The obtained results are presented in table 1. The best BLEU scores are obtained for the couples (MSA-Pal) (73.72%) and (MOR-MSA) (80.14%). In general, translation from MSA into (MOR, PAL, SYR, ANB) yielded close BLEU scores, around 71%, and from MSA into Algerian dialects (Alg, ANB, Kab) the scores are around 63%. Whereas, we recorded surprisingly, the same BLEU score (around 64%) when translating into MSA,from KAB and SYR. On the other hand, (ALG, ANB, PAL) have an overall score of 69%. The worst BLEU for Translation into MSA is the one recorded from ANB: 57.56%. The impact of transliteration was very negative, the BLEU score is around 9.5% for (dialects-MSA) and ranges for (MSA-dialects) from 15% to 23.8% .

4.2 SMT

We used Moses2.0 to train the SMT model. The obtained results are presented in table 2. We can say that without transliteration, the performance of translation from MSA achieved by SMT is lower than RBMT, except for Palestinian (MSA-PAL) that has the best BLEU score (81.12%). However, Translation into MSA using SMT outperforms RBMT except for Kabyle dialect (MSA-KAB) Contrarywise, as shown in Table 2, the Buckwalter transliteration has a positive impact on SMT performance in most of cases except for the three

[5] http://www.qamus.org/transliteration.htm

[6] https://github.com/cshanbo/Smooth_BLEU/blob/master/BLEU.py

[7] https://github.com/pniyongabo/kinyarwandaRBMT

pairs (ANB-MSA), (ALG-MSA) and (ANB-MSA) . Note that the best BLEU score obtained in our experiments is 88.96% for (PAL-MSA). On the other hand, SMT provides the worst results for Kabyle dialect (44.89% for MSA-KAB) and (25.89% for KAB-MSA), though the rule based method yielded promising and surprising results (62.91% for MSA-KAB) and (64.06% for KAB-MSA).

5 Conclusion

In this paper, we presented a new extension to the Parallel Arabic Dialect Corpus PADIC by adding a zero-resourced dialect, namely: Algerian Kabyle dialect. We tested rule based and statistical machine translation models. We studied the impact of using Buckwalter transliteration on the performance of the trained models. The results are promising, we believe that we can further enhance the performance by applying some preprocessing steps as sentence tokenization, and using Neural MT.

References

Mourad Abbas, Mohamed Lichouri, and Abed Alhakim Freihat. 2019. St madar 2019 shared task: Arabic fine-grained dialect identification. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 269–273.

Ahmed Ali, Shammur Chowdhury, Mohamed Afify, Wassim El-Hajj, Hazem Hajj, Mourad Abbas, Amir Hussein, Nada Ghneim, Mohammad Abushariah, and Assal Alqudah. 2021. Connecting arabs: bridging the gap in dialectal speech recognition. *Communications of the ACM*, 64(4):124–129.

Amjad Almahairi, Kyunghyun Cho, Nizar Habash, and Aaron C. Courville. 2016. First result on arabic neural machine translation. *CoRR*, abs/1606.02680.

Abdullah Alrajeh. 2018. A recipe for arabic-english neural machine translation. *CoRR*, abs/1808.06116.

Rudi Cilibrasi and Paul MB Vitányi. 2005. Clustering by compression. *IEEE Transactions on Information theory*, 51(4):1523–1545.

Kareem Darwish, Nizar Habash, Mourad Abbas, Hend Al-Khalifa, Huseein T Al-Natsheh, Houda Bouamor, Karim Bouzoubaa, Violetta Cavalli-Sforza, Samhaa R El-Beltagy, Wassim El-Hajj, et al. 2021. A panoramic survey of natural language processing in the arab world. *Communications of the ACM*, 64(4):72–81.

Imane Guellil, Faical Azouaou, and Mourad Abbas. 2017. Neural vs statistical translation of algerian arabic dialect written with arabizi and arabic letter.

Salima Harrat, Karima Meftouh, Mourad Abbas, Salma Jamoussi, Motaz Saad, and Kamel Smaili. 2015. Cross-dialectal arabic processing. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 620–632. Springer.

Salima Harrat, Karima Meftouh, Mourad Abbas, and Kamel Smaili. 2014. Building resources for algerian arabic dialects. In *15th Annual Conference of the International Communication Association Interspeech*.

Katrin Kirchhoff, Bing Zhao, and Wen Wang. 2015. Exploiting out-of-domain data sources for dialectal arabic statistical machine translation. *arXiv preprint arXiv:1509.01938*.

Mohamed Lichouri and Mourad Abbas. 2020. Simple vs oversampling-based classification methods for fine grained arabic dialect identification in twitter. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 250–256.

Mohamed Lichouri, Mourad Abbas, Abed Alhakim Freihat, and Dhiya El Hak Megtouf. 2018. Word-level vs sentence-level language identification: Application to algerian and arabic dialects. *Procedia Computer Science*, 142:246–253.

Mohamed Lichouri, Mourad Abbas, Khaled Lounnas, Besma Benaziz, and Aicha Zitouni. 2021. Arabic dialect identification based on a weighted concatenation of tf-idf features. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 282–286.

Karima Meftouh, Salima Harrat, Salma Jamoussi, Mourad Abbas, and Kamel Smaili. 2015. Machine translation experiments on padic: A parallel arabic dialect corpus. In *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation*, pages 26–34.

Gonzalo Navarro. 2001. A guided tour to approximate string matching. *ACM computing surveys (CSUR)*, 33(1):31–88.

Hassan Sajjad, Kareem Darwish, and Yonatan Belinkov. 2013. Translating dialectal arabic to english. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 1–6.

Wael Salloum and Nizar Habash. 2011. Dialectal to standard arabic paraphrasing to improve arabic-english statistical machine translation. In *Proceedings of the first workshop on algorithms and resources for modelling of dialects and language varieties*, pages 10–21. Association for Computational Linguistics.

Wael Salloum and Nizar Habash. 2012. Elissa: A dialectal to standard arabic machine translation system. *Proceedings of COLING 2012: Demonstration Papers*, pages 385–392.

Wael Salloum and Nizar Habash. 2013. Dialectal arabic to english machine translation: Pivoting through modern standard arabic. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 348–358.

Hassan Sawaf. 2010. Arabic dialect handling in hybrid machine translation. In *Proceedings of the conference of the association for machine translation in the americas (amta), denver, colorado*.

Philip Schulz, Wilker Aziz, and Trevor Cohn. 2018. A stochastic decoder for neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1243–1252.

Paul M.B. Vitányi. 2011. Compression-based similarity. In *2011 First International Conference on Data Compression, Communications and Processing*, pages 111–118.

Rabih Zbib, Erika Malchiodi, Jacob Devlin, David Stallard, Spyros Matsoukas, Richard Schwartz, John Makhoul, Omar F Zaidan, and Chris Callison-Burch. 2012. Machine translation of arabic dialects. In *Proceedings of the 2012 conference of the north american chapter of the association for computational linguistics: Human language technologies*, pages 49–59. Association for Computational Linguistics.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. *arXiv preprint arXiv:1604.02201*.

Improving BERT Performance for Aspect-Based Sentiment Analysis

Akbar Karimi **Leonardo Rossi** **Andrea Prati**

University of Parma
{akbar.karimi, leonardo.rossi, andrea.prati}@unipr.it

Abstract

Aspect-Based Sentiment Analysis (ABSA) addresses the problem of extracting sentiments and their targets from opinionated data such as consumer product reviews. Analyzing the language used in a review is a difficult task that requires a deep understanding of the language. In recent years, deep language models, such as BERT, have shown great progress in this regard. In this work, we propose two simple modules called Parallel Aggregation and Hierarchical Aggregation to be utilized on top of BERT for two main ABSA tasks namely Aspect Extraction (AE) and Aspect Sentiment Classification (ASC). With the proposed modules, we show that the intermediate layers of the BERT architecture can be utilized for the enhancement of the model performance[1].

1 Introduction

In an industry setting, it is extremely important to have a valid conception of how consumers perceive the products. Nowadays, they communicate their perception through their comments on the products, using mostly social networks. They might have positive opinions which can lead to the success of a business or negative ones possibly leading to its demise. Due to the abundance of these views in many areas, their analysis is a time-consuming and labor-intensive task which is why a variety of machine learning techniques such as Support Vector Machines (SVM) (Cortes and Vapnik, 1995; Kiritchenko et al., 2014; Basari et al., 2013), Maximum Entropy (Jaynes, 1957; Nigam et al., 1999), Naive Bayes (Duda et al., 1973; Gamallo and Garcia, 2014; Dinu and Iuga, 2012), and Decision Trees (Quinlan, 1986; Wakade et al., 2012) have been proposed to perform opinion mining.

In recent years, Deep Learning (DL) techniques have been widely utilized due to the increase in computational power and the huge amount of freely available data on the Web (Zhang et al., 2015; Liu et al., 2015; Wang et al., 2016). One of the areas on which these techniques have had a great impact is Natural Language Processing (NLP) where modeling (i.e. understanding) the language plays a crucial role. BERT (Devlin et al., 2019) is a state-of-the-art model of this kind which has become widely utilized in many NLP tasks (Kantor et al., 2019; Davison et al., 2019) as well as in other fields (Peng et al., 2019; Alsentzer et al., 2019). It has been trained on a large corpus of Wikipedia documents and books in order to *learn* the language syntax and semantics from the context. The main component of its architecture is called the transformer (Vaswani et al., 2017) block consisting of attention heads. These heads have been designed to pay particular attention to parts of the input sentences that correspond to a particular given task (Vig and Belinkov, 2019). In this work, we utilize BERT for Aspect-Based Sentiment Analysis (ABSA) tasks.

Our main contribution is the proposal of two simple modules that can help improve the performance of the BERT model. In our models we opt for Conditional Random Fields (CRFs) for the sequence labeling task which yield better results. In addition, our experiments show that training BERT for more number of epochs does not cause the model to overfit. However, after a certain number of training epochs, the learning seems to stop.

2 Related Work

Recently, there has been a large body of work which utilizes the BERT model for various tasks in NLP in general such as text classification (Sun et al., 2019b), question answering (Yang et al.,

[1] https://github.com/IMPLabUniPr/BERT-for-ABSA

2019), summarization (Liu, 2019) and, in particular, ABSA tasks (Hoang et al., 2019).

Using Graph Convolutional Networks (GCNs), Zhao et al. (2020) take into account sentiment dependencies in a sequence. In other words, they show that when there are multiple aspects in a sequence, the sentiment of one of them can affect that of the other one. Making use of this information can increase the performance of the model. Some studies convert the Aspect Extraction (AE) task into a sentence-pair classification task. For instance, Sun et al. (2019a) construct auxiliary sentences using the aspect terms of a sequence. Then, utilizing both sequences, they fine-tune BERT on this specific task.

Word and sentence level representations of a model can also be enriched using domain-specific data. Xu et al. (2019) show this by post-training the BERT model, which they call BERT-PT, on additional restaurant and laptop data. In our experiments, we use their pre-trained model for the initialization of our models. Due to the particular architecture of the BERT model, extra modules can be attached on top of it. Li et al. (2019) add different layers such as an RNN and a CRF layer to perform ABSA in an end-to-end fashion. In our work, we use the same layer modules from the BERT architecture and employ the hidden layers for prediction as well.

3 Aspect-Based Sentiment Analysis Tasks

Two of the main tasks in ABSA are Aspect Extraction (AE) and Aspect Sentiment Classification (ASC). While the latter deals with the semantics of a sentence as a whole, the former is concerned with finding which word that sentiment refers to. We briefly describe them in this section.

3.1 Aspect Extraction

In AE, the goal is to extract a specific aspect of a product towards which some type of sentiment is expressed in a review. For instance, in the sentence, "*The laptop has a good battery.*", the word *battery* is the aspect which is extracted. Sometimes, the aspect words can be multiple in which case all of them need to be labeled accordingly. This task can be seen as a sequence labeling task, where the words are assigned a label from the set of three letters namely $\{B, I, O\}$. Each word in the sequence can be the beginning word of aspect terms (B), among the aspect terms (I), or not an aspect term (O). The classification of each word into one of these three classes, is accomplished using a fully connected layer on top of the BERT architecture and applying the Softmax function.

3.2 Aspect Sentiment Classification

In this task, the goal is to extract the sentiment expressed in a review by the consumer. Given a sequence, one of the three classes of *Positive*, *Negative*, and *Neutral* is extracted as the class of that sequence. The representation for this element is embodied in the architecture of the BERT model. For each sequence as input, there are two extra tokens that are used by the BERT model:

$$[CLS], w_1, w_2, ..., w_n, [SEP]$$

where w_i are the sequence words and $[CLS]$ and $[SEP]$ tokens are concatenated to the sentence in the input stage. While the $[CLS]$ token is there to store the sentiment representation of the sentence, the $[SEP]$ token is used to separate input sequences in case there are more than one (e.g. in a question answering task). In the final layer of the architecture, a Softmax function is applied to the $[CLS]$ embedding and the class probability is computed.

4 Proposed Model

Deep models can capture deeper knowledge of the language as they grow. As shown by Jawahar et al. (2019), the initial to middle layers of BERT can extract syntactic information, whereas the language semantics are represented in higher layers. Since extracting the sentence sentiment is semantically demanding, we expect to see this in higher layers of the network. This is the intuition behind our models where we exploit the final layers of the BERT model.

The two models that we introduce here are similar in principle, but slightly differ in implementation. Also, for the two tasks, the losses are computed differently. While for the ASC task we utilize cross-entropy loss, for the AE task, we make use of CRFs. The reason for this choice is that the AE task can be treated as sequence labeling. Therefore, taking into account the previous labels in the sequence is of high importance, which is exactly what the CRF layer does.

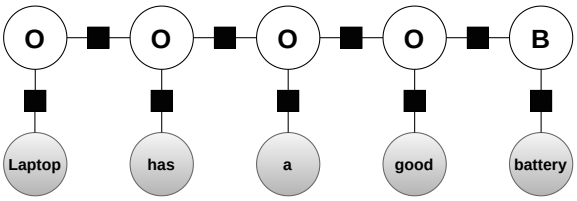


Figure 1: An example of representing a sentence with its word labels using CRFs.

4.1 Conditional Random Fields

CRFs (Lafferty et al., 2001) are a type of graphical models and have been used both in computer vision (e.g. for pixel-level labeling (Zheng et al., 2015)) and in NLP for sequence labeling.

Since AE can be considered a sequence labeling task, we opt for using a CRF layer in the last part of our models. The justification for the use of a CRF module for AE is that doing so helps the network to take into account the joint distribution of the labels. This can be significant since the labels of sequence words are dependent on the words that appear before them. For instance, as is seen in Figure 1, the occurrence of the adjective *good* can give the model a clue that the next word is probably not another adjective. The equation with which the joint probability of the labels is computed is as follows:

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{t=1}^{T} \exp\left\{ \sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, \mathbf{x}_t) \right\} \quad (1)$$

In Formula 1, $\mathbf{x}$ is the observed sequence, $\mathbf{y}$ is the sequence of labels, and k and t are the indices for feature functions and time steps in the sequence, respectively. The relations between sequence words are represented by using feature functions $\{f_k\}$. These relations can be strong or weak, or non-existent at all. They are controlled by their weights $\{\theta_k\}$ which are computed during the training phase. Finally, $Z(\mathbf{x})$ is a normalization factor.

4.2 Parallel Aggregation

Rossi et al. (2020) showed that the hidden layers of deep models can be exploited more to extract region specific information. Inspired by their work, we propose a model called P-SUM applying BERT layer modules on each one of the best performing BERT layers. Figure 2 shows the details of this model. We exploit the last four layers of the BERT model by adding one more BERT layer plus a fully connected layer and calculating the loss of that branch on the input data, using a Softmax function

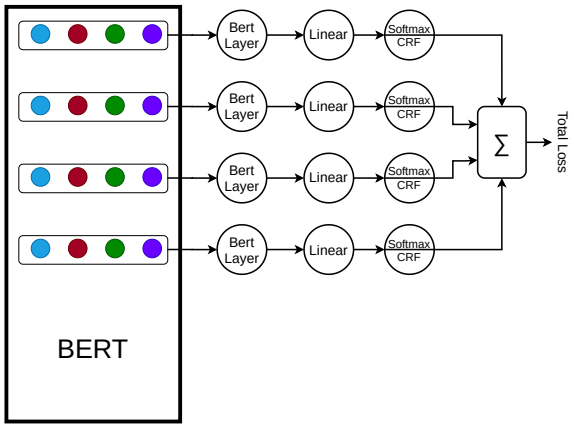


Figure 2: Parallel aggregation (P-SUM)

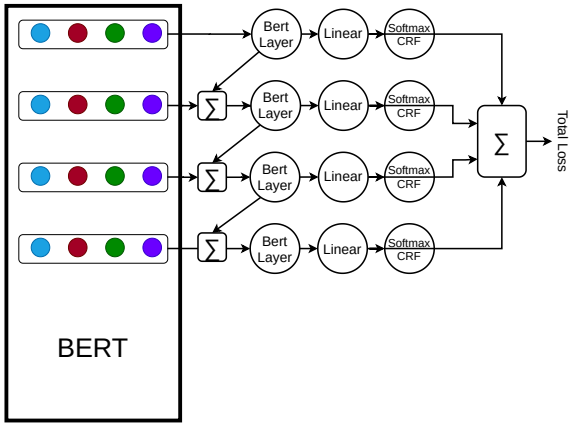


Figure 3: Hierarchical aggregation (H-SUM)

and a conditional random fields layer. The reason is that all deeper layers contain most of the related information regarding the task. Therefore, extracting this information from each one of them and combining them can produce richer representations of the semantics. In order to calculate the total loss, the loss values of all branches are summed up which is indicated with Σ notation in the diagram. This is done so, in order to take all the losses into account when optimizing the parameters. However, to compute the network's output $logits$, we average over the output $logits$ of the four branches.

4.3 Hierarchical Aggregation

Our hierarchical aggregation (H-SUM) model is inspired by the use of Feature Pyramid Networks (FPNs) (Lin et al., 2017). The goal is to extract more semantics from the hidden layers of the BERT model. The architecture of the H-SUM model can be seen in Figure 3. Here, after applying a BERT layer on each one of the hidden layers, the output is aggregated (element-wise) with the previous

	Train		Test	
Dataset	S	A	S	A
LPT14	3045	2358	800	654
RST16	2000	1743	676	622

Table 1: Laptop (LPT14) and restaurant (RST16) datasets from SemEval 2014 and 2016, respectively, for AE. S: Number of sentences; A: Number of aspects.

	Train				Test			
Dataset	S	Pos	Neg	Neu	S	Pos	Neg	Neu
LPT14	2313	987	866	460	638	341	128	169
RST14	3102	2164	805	633	1120	728	196	196

Table 2: Laptop (LPT14) and restaurant (RST14) datasets from SemEval 2014 for ASC. S: Number of all sentences; Pos, Neg, Neu: Number of positive, negative, and neutral sentiments, respectively.

layer. At the same time, similar to the P-SUM, each branch produces a loss value which contributes to the total loss equally since the total loss is the summation of all of them.

5 Experiments

In order to carry out our experiments, we use the same codebase as Xu et al. (2019). We ran the experiments on a GPU (GeForce RTX 2070) with 8 GB of memory using batches of 16 for both our models and the BERT-PT model as the baseline. For training, Adam optimizer was used and the learning rate was set to $3e-5$. From the distributed training data, we used 150 examples as the validation. To evaluate the models, the official scripts were used for the AE tasks and the script from the same codebase was used for the ASC task. Results are reported in F1 for AE and in Accuracy and MF1 for ASC. While F1 score is the harmonic mean of precision and recall, MF1 score is the average of F1 score for each class.

5.1 Datasets

In our experiments, we utilized laptop and restaurant datasets from SemEval 2014 (Pontiki et al., 2014) Subtask 2 and 2016 (Pontiki et al., 2016) Subtask 1. The collections consist of user reviews which have been annotated manually. Tables 1 and 2 show the statistics of these datasets. In choosing the datasets, we opted for the ones utilized in previous works (Karimi et al., 2020; Xu et al., 2019) so that we can draw a reliable comparison between the performance of our models and those ones.

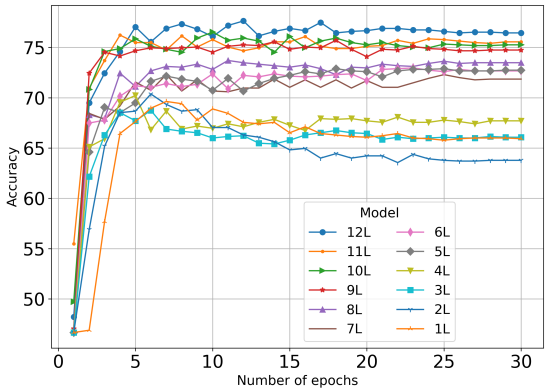


Figure 4: Performance of BERT layers initialized by BERT-PT weights for ASC on RST14 validation data. Each model is the BERT model using the specified number of layers. 1L means using the first layers, 2L means using the first 2 layers, etc. Accuracy values are percentages.

5.2 Performance of BERT Layers

Depending on the depth of the network, it can perform differently. Therefore, we carried out experiments to find out how each layer of the BERT model performs. The results are shown in Figure 4. As can be seen, better performance is achieved in the deeper layers, especially the last four. Therefore, our modules operate on these four layers to achieve an improved model.

5.3 Increasing Training Epochs

More training can lead to a better performance of the network. However, one risks the peril of overfitting especially when the number of training examples are not considered to be large compared to the number of parameters contained in the model. However, in the case of BERT, as was also observed by Li et al. (2019), it seems that with more training the model does not overfit although the number of the training data points is relatively small. The reason behind this could be the fact that we are using an already pre-trained model which has seen an enormous amount of data (Wikipedia and Books Corpus). Therefore, we can expect that by performing more training, the model will still be able to generalize.

The same observation can be made by looking at the validation losses in Figure 5. In case of an overfit, we would expect the losses to go up and the performance to go down. However, we see that with the increase in loss after the second epoch, the

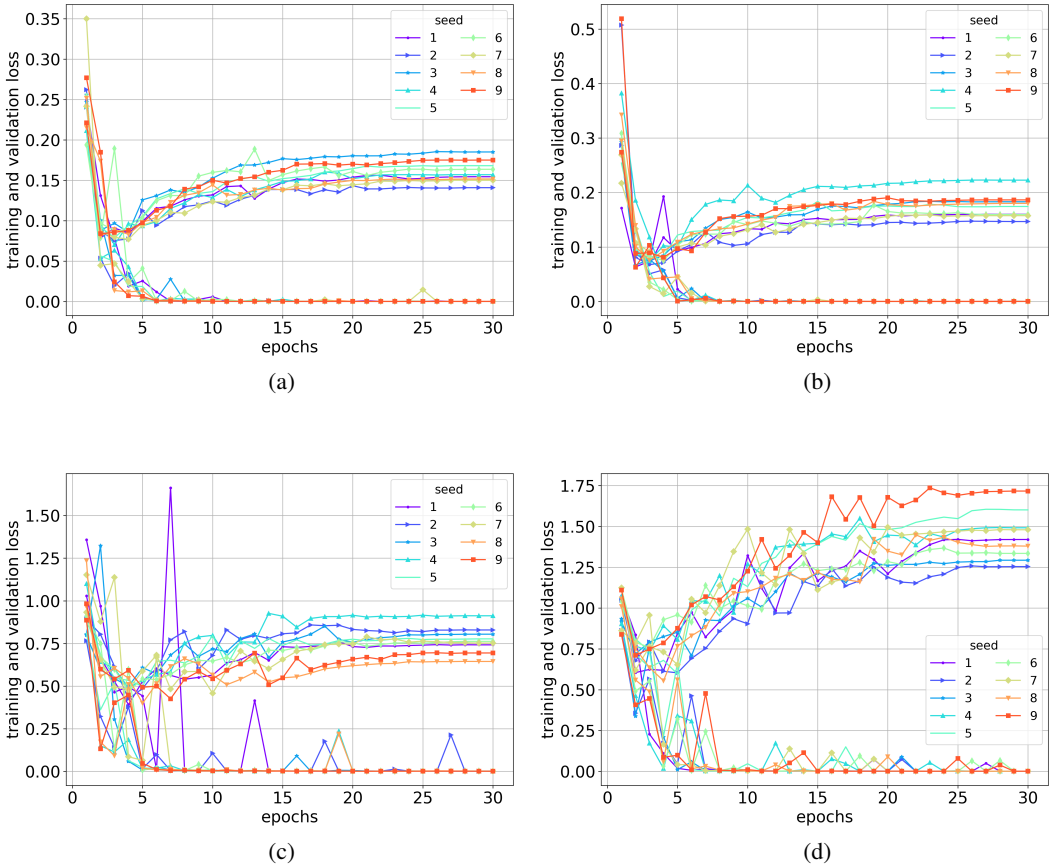


Figure 5: Training and validation losses of the 12-layer BERT model initialized with BERT-PT weights for AE (laptop (a) and restaurant (b)) and ASC (laptop (c) and restaurant (d)). In each figure, the upper lines are validation losses and the bottom lines are training losses, each line corresponding to a seed number.

performance still improves for a couple of epochs and then fluctuates in the subsequent ones (Figure 4). This suggests that with more training, the network weights continue to change until they remain almost stable in later epochs, indicating that there is no more learning. From Figure 4, we see that with 4 or 5 training epochs we get near the maximum performance. Although some later epochs such as 12 yield better results for the 12-layer version, it can be considered negligible.

6 Results

Our experimental results show that with the increase of the training epochs the BERT model also improves. These results can be seen in Table 3. To compare our proposed models with Xu et al. (2019), we perform the same model selection for both of them. Unlike Xu et al. (2019) and Karimi et al. (2020) who select their best models based on the lowest validation loss, we choose the models trained with four epochs after observing that accuracy goes up on the validation sets (Figure 4). Therefore, in Table 3, we report the original BERT-PT scores as well as the ones for our model selection. From Table 3, it can also be seen that the proposed models outperform the newly selected BERT-PT model in both datasets and tasks with improvements in MF1 score as high as **+1.78** and **+2** for ASC on laptop and restaurant, respectively.

It is also worth noting that, in terms of accuracy, the H-SUM module performs better than the P-SUM module in most cases. This could be attributed to the hierarchical structure of the module and the fact that each branch of this module benefits from the information processed in the preceding branch.

7 Conclusion

We proposed two simple modules utilizing the hidden layers of the BERT language model to produce

	AE		ASC			
	LPT14	RST16	LPT14		RST14	
Models	F1	F1	Acc	MF1	Acc	MF1
BERT	79.28	74.10	75.29	71.91	81.54	71.94
DE-CNN (Xu et al., 2018)	81.59	74.37	-	-	-	-
BERT-PT (Xu et al., 2019)	84.26	77.97	78.07	75.08	84.95	76.96
BAT (Karimi et al., 2020)	85.57	81.50	79.35	76.50	86.03	79.24
BERT-PT*	85.57	81.57	78.21	75.03	85.43	77.68
P-SUM	**85.94**	**81.99**	**79.55**	**76.81**	**86.30**	**79.68**
H-SUM	**86.09**	**82.34**	**79.40**	**76.52**	**86.37**	**79.67**

Table 3: Comparison of the results for Aspect Extraction (AE) and Aspect Sentiment Classification (ASC). BERT-PT* is the original BERT-PT model using our model selection. The boldfaced numbers show the outperforming models using the same settings. Each score in the table is the average of 9 runs. Results for the cited papers are reported from the corresponding paper. The other models are run for 4 epochs. LPT: Laptop, RST: Restaurant, Acc: Accuracy , MF1: Macro-F1. Values are percentages.

deeper semantic representations of input sequences. The layers are once aggregated in a parallel fashion and once hierarchically. For each branch of the architecture built on top of the selected hidden layers, we compute the loss separately. These losses are then aggregated to produce the final loss of the model. We address aspect extraction using conditional random fields which helps to take into account the joint distribution of the sequence labels to achieve more accurate predictions. Our experiments show that the proposed approaches outperform the post-trained vanilla BERT model.

References

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78.

Abd Samad Hasan Basari, Burairah Hussin, I Gede Pramudya Ananta, and Junta Zeniarja. 2013. Opinion mining of movie review using hybrid method of support vector machine and particle swarm optimization. *Procedia Engineering*, 53:453–462.

Corinna Cortes and Vladimir Vapnik. 1995. Support vector machine. *Machine learning*, 20(3):273–297.

Joe Davison, Joshua Feldman, and Alexander M Rush. 2019. Commonsense knowledge mining from pretrained 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 1173–1178.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*.

Liviu P Dinu and Iulia Iuga. 2012. The naive bayes classifier in opinion mining: in search of the best feature set. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 556–567. Springer.

Richard O Duda, Peter E Hart, et al. 1973. *Pattern classification and scene analysis*, volume 3. Wiley New York.

Pablo Gamallo and Marcos Garcia. 2014. Citius: A naive-bayes strategy for sentiment analysis on english tweets. In *Proceedings of the 8th international Workshop on Semantic Evaluation (SemEval 2014)*, pages 171–175.

Mickel Hoang, Oskar Alija Bihorac, and Jacobo Rouces. 2019. Aspect-based sentiment analysis using bert. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 187–196.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657.

Edwin T Jaynes. 1957. Information theory and statistical mechanics. *Physical review*, 106(4):620.

Yoav Kantor, Yoav Katz, Leshem Choshen, Edo Cohen-Karlik, Naftali Liberman, Assaf Toledo, Amir Menczel, and Noam Slonim. 2019. Learning to combine grammatical error corrections. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 139–148.

Akbar Karimi, Leonardo Rossi, Andrea Prati, and Katharina Full. 2020. Adversarial training for aspect-based sentiment analysis with BERT. *arXiv preprint arXiv:2001.11316*.

Svetlana Kiritchenko, Xiaodan Zhu, Colin Cherry, and Saif Mohammad. 2014. Nrc-canada-2014: Detecting aspects and sentiment in customer reviews. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 437–442.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

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.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125.

Pengfei Liu, Shafiq Joty, and Helen Meng. 2015. Fine-grained opinion mining with recurrent neural networks and word embeddings. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1433–1443.

Yang Liu. 2019. Fine-tune BERT for extractive summarization. *arXiv preprint arXiv:1903.10318*.

Kamal Nigam, John Lafferty, and Andrew McCallum. 1999. Using maximum entropy for text classification. In *IJCAI-99 workshop on machine learning for information filtering*, volume 1, pages 61–67.

Yifan Peng, Shankai Yan, and Zhiyong Lu. 2019. Transfer learning in biomedical natural language processing: An evaluation of BERT and ELMo on ten benchmarking datasets. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 58–65.

Maria Pontiki, Dimitrios Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pages 27–35.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, AL-Smadi Mohammad, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 19–30.

J. Ross Quinlan. 1986. Induction of decision trees. *Machine learning*, 1(1):81–106.

Leonardo Rossi, Akbar Karimi, and Andrea Prati. 2020. A novel region of interest extraction layer for instance segmentation. *arXiv preprint arXiv:2004.13665*.

Chi Sun, Luyao Huang, and Xipeng Qiu. 2019a. Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. In *Proceedings of NAACL-HLT*, pages 380–385.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019b. How to fine-tune BERT for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer.

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.

Jesse Vig and Yonatan Belinkov. 2019. Analyzing the structure of attention in a transformer language model. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 63–76.

Shruti Wakade, Chandra Shekar, Kathy J Liszka, and Chien-Chung Chan. 2012. Text mining for sentiment analysis of twitter data. In *Proceedings of the International Conference on Information and Knowledge Engineering (IKE)*, page 1. The Steering Committee of The World Congress in Computer Science, Computer ....

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 606–615.

Hu Xu, Bing Liu, Lei Shu, and S Yu Philip. 2018. Double embeddings and cnn-based sequence labeling for aspect extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 592–598.

Hu Xu, Bing Liu, Lei Shu, and S Yu Philip. 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.

Wei Yang, Yuqing Xie, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019. Data augmentation for BERT fine-tuning in open-domain question answering. *arXiv preprint arXiv:1904.06652*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1*, pages 649–657.

Pinlong Zhao, Linlin Hou, and Ou Wu. 2020. Modeling sentiment dependencies with graph convolutional networks for aspect-level sentiment classification. *Knowledge-Based Systems*, 193:105443.

Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. 2015. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1529–1537.

MAPLE – MAsking words to generate blackout Poetry using sequence-to-sequence LEarning

Aditeya Baral
PES University, Bangalore, India
aditeya.baral@gmail.com

Himanshu Jain
PES University, Bangalore, India
nhimanshujain@gmail.com

Deeksha D
PES University, Bangalore, India
deekshad132@gmail.com

Mamatha HR
PES University, Bangalore, India
mamathahr@pes.edu

Abstract

Poetry has morphed rapidly over changing times with non-traditional forms stirring the creative minds of people today. One such type of poetry is blackout poetry. Blackout poetry is a form of poetry in which words in a passage are masked, except for a few which when combined together in order to convey some meaning. With the recent developments in Natural Language Processing aiming to simulate human creativity, we propose a novel approach to blackout poetry generation employing deep learning. We explore four different architectures, namely an encoder-decoder with Bidirectional Long Short-Term Memory (LSTM) and Attention, a Bidirectional LSTM Conditional Random Fields (LSTM-CRF) architecture, Bidirectional Encoder Representations from Transformers (BERT) and Robustly Optimized BERT Pre-training Approach (RoBERTa). The first architecture employs abstractive summarization and the remaining employed sequence labelling to generate poetry. The Transformer based architectures prove to be the best working models, and were also able to pass a Turing Test as well.

1 Introduction

Poems are seen as an outlet through which a poet can express their creativity and emotions and deliver strong and vibrant messages to the readers. Some forms of poetry use rhyming schemes of two or more lines while some other poetry forms impress the reader through the beauty of the words selected and their arrangement. The latter type of poems are free-form, and they don't follow any formal structures.

Blackout poetry (Miller, 2017) is the most recent form of poetry in which one picks out words from a passage to generate free-form poems. Such poems may provide a completely different sense as opposed to the meaning of the passage. This art of forming poems from any passage has swiftly gained popularity over the last decade.

Our work aims to generate blackout poetry from any given passage. We use existing state-of-the-art architectures to generate these free-form, blackout poems using techniques like abstractive summarization and sequence labelling. The results have shown that Transformers are very effective in generation of such poems but no single model is capable of producing satisfactory results. Evaluation of our model was done by performing the Turing Test (Wikipedia, b) where we compared the poems generated by humans and machines.

2 Background

Blackout poetry is a recently established form of poetry that rose to popularity in 2005. A passage consisting of words is taken and "blackened" or masked out, except for a few words such that these leftover words when combined together convey some meaning. Often, instead of simply masking out the words, blackout poets tend to draw patterns related to the poem. It is also seen as a way to repurpose old newspapers and magazines. This form of poetry was popularized by Austin Kleon (Kleon) (see Figure 1), who created such poetry from old newspapers. The New York Times also features a digital blackout poetry generator (Times, 2014) that allows visitors to generate blackout poems on their website.

3 Previous Work

The first and only known work of automated blackout poetry generation was observed during the National Novel Generation Month (NaNoGenMo) (Month) 2016. Liza Daly's (Daly) work was able to generate blackout poems from any given passage of text by looking up sequences of words that followed a given set of parts-of-speech grammar

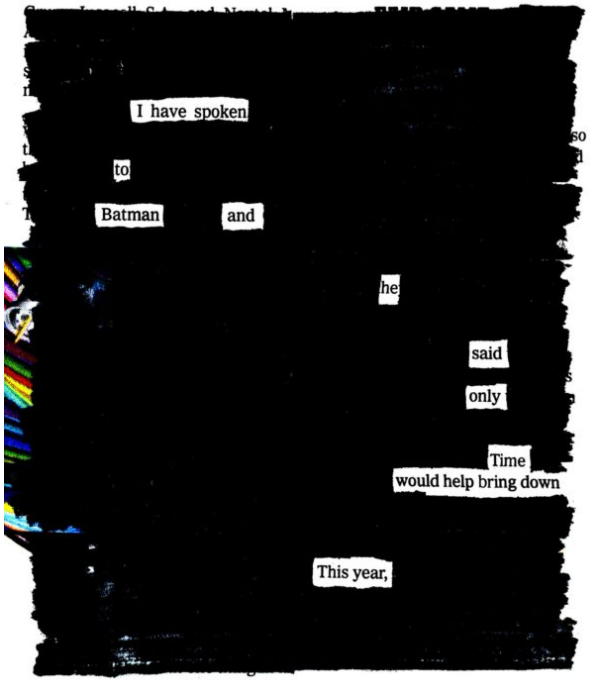


Figure 1: Blackout poetry by Austin Kleon for the New York Times

rules (see Table 1). Although her approach was rule-based, on rare occasions it was able to pick out sequences of words that were able to convey some meaning. However, the approach is very restricted, since it only looks for a given number of grammar-rules and extracts the first sequence of words matching a rule. Additionally, there is no way to verify whether the generated poem is syntactically or semantically correct.

4 Proposed System

Our ultimate objective is to apply deep learning to generate blackout poems which match or possibly beat Liza Daly's model (which use fixed grammar-rules) in terms of human-nature and readability. We look at two major Natural Language Processing techniques – abstractive summarization (Gupta) and sequence labelling (Wikipedia, a). Abstractive summarization is the process of generating summaries of any given passage, such that the lines in the summary are not chosen from the passage itself, and are thus rewritten by the machine. Sequence labelling , also known as token classification is a method to assign a class or a label to every word or token in a given sequence of text.

Passage	**Generated Blackout Poem**
Loving you feels like the soft touch of velvet rain.Over the hills we walk together again. Violet eyes, ruby lips, a beautiful smile. Every step a together lips smile	-
The skies can't keep their secret! They tell it to the hills The hills just tell the orchards. And they the daffodils! A bird, by chance, that goes that way Soft overheard the whole.	secret hills tell daffodils
The mountain sat upon the plain In his eternal chair, His observation omnifold, His inquest everywhere. The seasons prayed around his knees, Like children round a sire: Grandfather of the days is he, Of dawn the ancestor.	sat his chair
Summer for thee grant I may be When summer days are flown! Thy music still when whippoorwill And oriole are done! For thee to bloom, I 'll skip the tomb And sow my blossoms o'er!	when flown blossoms
Mine enemy is growing old I have at last revenge. The palate of the hate departs; If any would avenge, Let him be quick, the viand flits, It is a faded meat. Anger as soon as fed is dead;	mine enemy makes

Table 1: Poems generated using Liza Daly's rules

5 Workflow

To obtain satisfactory results, the process of data collection as well as pre-processing had to be paid significant attention since data was scarce. We experiment and try four state-of-the-art architectures for the two major approaches to generating blackout poetry.

5.1 Data Collection

Due to the lack of any publicly available dataset containing passages and extracted blackout poems, we resort to using Liza Daly's method to synthetically generate a dataset.

To ensure that our generated data possesses high level of creativity as well as to enforce artistic nature, we use a large collection of traditional poems written and submitted to a public archive (kag). These poems were used as our passages for input. We also restrict the size of the passage to betweeen 8 and 120 words for computational reasons.

The grammar rules employed are different from Liza Daly's work. To obtain a set of systematic grammar rules that would ensure a higher number of sequences being extracted, we choose frequently used grammar rules in poetry. Although it would not completely remove the issue of generating nonsensible data, it did help produce more syntactically accurate data. To do this, a large number of haikus – short, 3 line poems that can be read as a single sentence – were obtained, and the parts-of-speech tags for its constituent words was retrieved. We analysed the repeating nature of the sequence of the parts-of-speech and choose the most frequent rules (see Table 2) (see Table 3).

Parts-of-Speech Grammar Rule	Frequency
PROPN PROPN DET ADJ NOUN ADP DET NOUN	7
PROPN PROPN DET ADJ NOUN ADP DET NOUN NOUN	4
ADJ NOUN DET NOUN VERB ADP DET NOUN	3
VERB PROPN DET NOUN NOUN	2
NOUN ADJ NOUN ADP DET NOUN	2
VERB DET NOUN ADP DET NOUN	2
VERB DET NOUN ADP PROPN NOUN	2
NOUN NOUN DET ADJ NOUN ADP DET NOUN	2
NOUN NOUN NOUN VERB ADP DET NOUN	2

Table 2: Parts-of-Speech Grammar rules

5.2 Data Pre-Processing

We retain only those poems which are at least 5 words in length. We pre-process our data by first removing any form of bad symbols (characters apart from letters, numbers and punctuation) in the generated poems as well as passages. The choice to convert the data into lowercase is decided based on the model architecture being used (see Table 4). We further remove any duplicate passages and poems by sampling unique pairs from the dataset (see Table 5).

5.3 Evaluation Metric

Since we are attempting to simulate creativity in a machine, a quantitative metric cannot be used to evaluate our model. We resort to using a Turing Test to gauge the quality of our poems. A questionnaire was constructed with 8 human written poems and the best machine generated poems each and were randomly shuffled. We asked the audience to choose among these three options – written by a human, machine or unable to draw a conclusion. The questionnaire was shared with people in the age group 18-22 and observed 120 responses.

5.4 Models

5.4.1 Abstractive Summarization using Bidirectional LSTM and Attention

We employ abstractive summarization by using an encoder-decoder architecture using a Bidirectional LSTM (Hochreiter and Schmidhuber, 1997) (Rumelhart and McClelland, 1997) and unidirectional LSTM respectively. We also include Bahdanau's Attention (Bahdanau et al., 2014) between the encoder and the decoder to increase performance on large sequences. fastText (Bojanowski et al., 2017) Word embeddings trained on passages are used to initialise the fixed embedding layer. The latent dimensions of the context vector are set to 1024 and the size of the embedding layer is set to 100.

The model uses the Adam (Kingma and Ba, 2014) optimiser and sparse categorical cross-

Passage	Generated Blackout Poem
Loving you feels like the soft touch of velvet rain.Over the hills we walk together again. Violet eyes, ruby lips, a beautiful smile. Every step a blessing walking down the aisle.	violet ruby a beautiful smile
The skies can't keep their secret! They tell it to the hills The hills just tell the orchards. And they the daffodils! A bird, by chance, that goes that way Soft overheard the whole.	secret hills the daffodils by that way
The mountain sat upon the plain In his eternal chair, His observation omnifold, His inquest everywhere. The seasons prayed around his knees, Like children round a sire: Grandfather of the days is he, Of dawn the ancestor.	mountain seasons a sire days of the ancestor
Summer for thee grant I may be When summer days are flown! Thy music still when whippoorwill And oriole are done! For thee to bloom, I 'll skip the tomb And sow my blossoms o'er!	summer days music
Mine enemy is growing old I have at last revenge. The palate of the hate departs; If any would avenge, Let him be quick, the viand flits, It is a faded meat. Anger as soon as fed is dead;	mine enemy is revenge

Table 3: Blackout Poetry generated using statistically obtained rules

Dataset Attribute	Value
Number of Passages	54629
Vocabulary Size (Cased)	114562
Vocabulary Size (Uncased)	100820
Maximum Passage Length	120
Minimum Poem Length	5

Table 4: Dataset attributes after pre-processing

Dataset Attributes	Total	Train	Test
Number of Passages	16903	15222	1681
Vocabulary Size (Cased)	114562		
Vocabulary Size (Uncased)	100820		

Table 5: Dataset Attributes after sampling

entropy as the loss function. The model is trained for 10 epochs. Our training data consists of the passage as the input, and the poem as the expected output (see Table 6).

5.4.2 Bidirectional LSTM-CRF

Our second model uses a Bidirectional LSTM with a Conditional Random Field (Lafferty et al., 2001) layer to perform sequence labelling. A single Bidirectional LSTM layer with 512 units is used with a feed-forward layer with softmax as the activation function. The CRF layer is initialised with 2 classes and is connected to the Bidirectional LSTM stack. The embedding layer is once again initialised with the weights from fastText word embeddings obtained from the passages. The model was trained for 5 epochs with Adam as the optimiser and CRF loss (negative log-likelihood for linear chain CRF) as the loss function. Our training data consisted of passages as input and index based position-labels from the poem as the expected output (see Table 7).

Attribute	Value
embedding_size	100
Bidirectional LSTM units	1024
activation function	softmax
epochs	10
loss	sparse-categorical-crossentropy
optimiser	adam

Table 6: Chosen Parameters for Abstrative Summarization using Bidirectional LSTM with Attention

Attribute	Value
embedding_size	100
Bidirectional LSTM units	512
activation function	softmax
epochs	5
loss	crf_loss
optimiser	adam

Table 7: Chosen Parameters for Bidirectional LSTM-CRF

Attribute	Value
model_id	bert-base-cased, bert-base-uncased
epochs	1

Table 8: Chosen Parameters for BERT

5.4.3 BERT

We apply the bidirectional learning of the Transformer architecture for sequence labelling. Two vanilla BERT (Devlin et al., 2018) pre-trained architectures are chosen, which have been fine-tuned on cased as well as uncased data. The dataset is was converted to lowercase for the cased BERT architectures and is used to fine-tune the model. The base BERT model is used for both mentioned architectures and is fine-tuned for 1 epoch (see Table 8).

5.4.4 RoBERTa

The base RoBERTa (Liu et al., 2019) pre-trained architecture is chosen, which has been pre-trained on cased data. This model was fine-tuned for 1 epoch (see Table 9).

5.5 Post-Processing

Post-processing of the generated output is performed to enhance them and bring back the writing style of the passage. These include steps such as prepending skipped articles before a word, appending punctuation from the original passage after a word and capitalisation of the first letters of words which were converted to lowercase during pre-processing.

Attribute	Value
model_id	roberta-base
epochs	1

Table 9: Chosen Parameters for RoBERTa

6 Results

Our results show a significant improvement in quality over the poems generated by Liza Daly. The GPT-2 language model was used to measure the perplexity of the generated poems. Our model was able to obtain an average perplexity score of 5758.87, while Liza Daly's poems obtained an average perplexity score of 6511.533. Since a lower perplexity score indicates a more probable sequence, we can conclude that our model was capable of generating better and more probable poems.

Out of 56k randomly generated poems, only 0.1% of Liza Daly's poems formed valid grammatical sequences while 3% of our generated poems were valid. Although it is to be noted that this comparison is baseless since blackout poems are often grammatically incorrect.

However, we do observe that all our models are highly inconsistent, and no model is able to consistently generate good quality results. This is due to the nature of the training dataset being used, which is synthetically generated using a set of grammar rules and hence contains quite a few bad examples of poetry. We observe that the Transformer based architectures (see Table 10) (see Table 11) (see Table 12) perform nearly the same (but with BERT being more consistent) and both outperform the sequence model based architectures used by a significant margin. The Bidirectional LSTM-CRF model performs the worst, with the model not generating any kind of output.

7 Turing Test Analysis

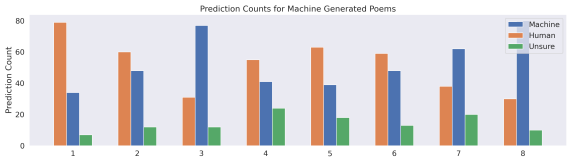


Figure 2: Prediction counts of machine generated poems

Figure 3: Prediction counts of human written poems

We observe that for a few machine generated poems, people were easily able to make the right

Passage	Generated Blackout Poem
You want someone to hear you run your mouth you talk loud now my aggravations got the best of me you tell your stories i try to wrap my brain I've lost count I pick the best and write your fiction it makes for a laugh or something to fill a void shut your mouth	mouth aggravations the best stories about a laugh
So the steering wheel showed a ship in my dad's coupe from years ago cars in boys' mind brakes just won't slip so the steering wheel showed a ship the fresh minted smell brewed air sip glow flown style blur torn roam wild show so the steering wheel showed a ship in my dad's coupe from years ago	forest grass the thorn flowers like a shore
I am loosed, I am free I have no responsibility no longer to be found the shackles and chains that had me bound a new life waits ahead it is the unknown what I most dread though I'm free as a dove I'd surrender if bound by your love let not go is my plea without you I am lost hold onto me no matter the cost & responsibility shackles that new life	responsibility shackles that new life as a dove
I wonder if the river gets tired, it runs and runs but never stops, foaming swirling round the rocks, it mustn't be tired because if it were it surely would rest, it only runs past to be admired, so rivers never do get tired, though lazy sometimes, yes when rain isn't giving her best but when the rain is feeling well, the river rushes madder still to get to the ocean blue	foaming swirling the tired rivers to the ocean
Every spring after the rain, new life comes again, when seeds are sprouting, and even the smallest petals of grain	spring seeds the smallest flower

Table 10: Poems generated using Abstractive Summarization

Passage	Generated Blackout Poem
Ghosts are many in the stories But quite rare in realities, Yet children are afraid of those Cry at night dreams sometimes	Ghosts are many But rare in realities, sometimes
A pocketful of sympathy Is really rather wonderful. To stop a scratch from stinging. Or a bruise from black and bluing. A pocketful of sympathy - Can stop a heart from hurting, Or catch a tear that's falling Like a raindrop down a cheek. A pocketful of sympathy Costs absolutely nothing, It's the cheapest kind of plaster That you'll ever ever find. And a pocketful of sympathy Is like Lindsay's Magic Pudding 'Cos the more of it you give away The more you leave behind.	pocketful of sympathy Is the cheapest of Magic
Fire never dies, just smoulders, like love you need to fan it, to keep the flame alive. Like the smouldering embers, that needs only a little attention, to become a flame again. So the parting lovers, need only to kiss to ignite the flame, and start the passion again.	Fire smoulders, the embers, a little flame
In the city of sorrows, Is where we see so much Racism, Against Men and women, In the city of sorrows, There are so much injustice.	city sorrows much Racism, the city much injustice
Night whispers in the dark makes eerie sounds with the wind making it so comforting. I hear night sounds like an owl hooting far sitting alone in the dark moonlight still.	Night whispers the dark eerie sounds with the wind

Table 11: Poems generated using BERT

Passage	Generated Blackout Poem
Innocence and desire illusions of my thoughts forsaken into realization, hollow like our ears is the sifting destiny.	innocence illusions my thoughts into realization
When you're lonely I wish you company, when you're sad I wish you happiness, when you're heartbroken I wish you eternal love, when all is chaotic I wish you inner silence, when all seems empty I wish you hope.	heartbroken love is chaotic silence
My soul, sit thou a patient looker-on; judge not the play before the play is done: her plot hath many changes; every day speaks a new scene; the last act crowns the play.	my soul hath many changes; every new scene
The red sun rises without intent and shines the same on all of us. we play like children under the sun. one day, our ashes will scatter— it doesn't matter when. now the sun finds our innermost hearts, fills us with oblivion intense as the forest, winter and sea.	sun rises and shines our hearts
I am loosed, I am free I have no responsibility no longer to be found the shackles and chains that had me bound a new life waits ahead it is the unknown what I most dread though I'm free as a dove I'd surrender if bound by your love let not go is my plea please bring again my captivity without you I am lost hold onto me no matter the cost	responsibility shackles that new life

Table 12: Poems generated using RoBERTa

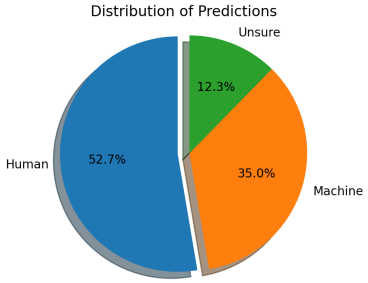


Figure 4: Distribution of predictions for machine generated poems predicted as human written

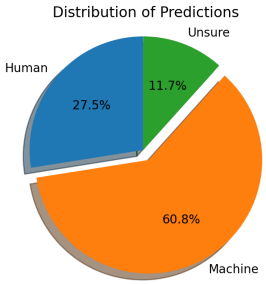


Figure 5: Distribution of predictions for machine generated poems predicted as machine generated

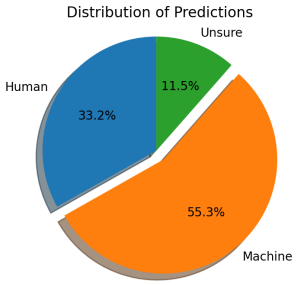


Figure 6: Distribution of predictions for human written poems predicted as machine generated

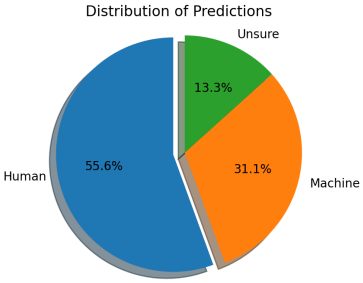


Figure 7: Distribution of predictions for human written poems predicted as human written

prediction (see Figure 8). However, for all cases we observe that the number of predictions passing the Turing Test are only a few more than the number of predictions failing the Turing Test (see Figure 2) (see Figure 3) (see Figure 4). This suggests that although the models were able to pass the Turing Test, they were neither poor to be labelled as machine generated nor great to have a higher prediction count for the other class (see Figure 5) (see Figure 6) (see Figure 7). We also observe that the average number of unsure predictions across all four cases remain about the same (see Figure 9).

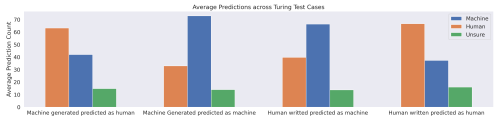


Figure 8: Average Predictions across Turing Test Cases

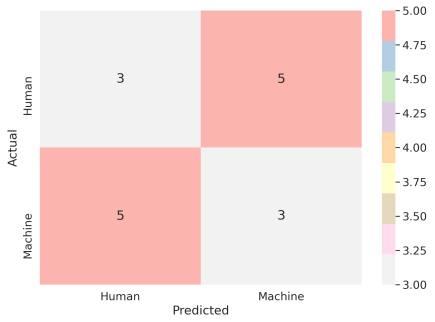


Figure 9: Confusion Matrix of Predictions

8 Conclusion

We thus show through our work how it is possible to generate free-form blackout-poetry using both abstractive summarization as well as sequence labelling techniques. Although the poems were able to pass a Turing Test, the models are highly inconsistent in their results. The Transformer architectures were observed to be the best working models, producing the best results both in terms of a syntactical as well as semantic sense.

The quality of the training dataset has a huge role to play in the result, since the dataset itself was generated synthetically using a set of grammar rules. Replacing this dataset with an actual blackout poetry dataset would greatly improve the performance of the models. Additionally, the poems generated can also be filtered using various linguistic tools to check for valid sequences.

References

Kaggle. https://www.kaggle.com/.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate.

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.

Liza Daly. Blackout. https://github.com/lizadaly/blackout.

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*.

Som Gupta. Abstractive summarization: An overview of the state of the art. https://www.sciencedirect.com/science/article/abs/pii/S0957417418307735.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory.

Diederik Kingma, P. and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

Austin Kleon. Newspaper blackout poems. https://austinkleon.com/category/newspaper-blackout-poems/.

John Lafferty, Andrew Mccallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

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.

E. CE Miller. 2017. What is blackout poetry? https://www.bustle.com/p/what-is-blackout-poetry-these-fascinating-poems-are-created-from-existing-art-78781.

National Novel Generation Month. Nanogenmo. https://nanogenmo.github.io/.

David E. Rumelhart and James L McClelland. 1997. Learning internal representations by error propagation.

New York Times. 2014. Searching for poetry in prose. https://www.nytimes.com/interactive/2014/multimedia/blackout-poetry.html.

Wikipedia. a. Sequence labelling. https://en.wikipedia.org/wiki/Sequence_labeling.

Wikipedia. b. Turing test. https://en.wikipedia.org/wiki/Turing_test.

Beyond Voice Activity Detection: Hybrid Audio Segmentation for Direct Speech Translation

Marco Gaido[†,*], **Matteo Negri**[†], **Mauro Cettolo**[†], **Marco Turchi**[†]
[†]Fondazione Bruno Kessler
[*]University of Trento
{mgaido,cettolo,negri,turchi}@fbk.eu

Abstract

The audio segmentation mismatch between training data and those seen at run-time is a major problem in direct speech translation. Indeed, while systems are usually trained on manually segmented corpora, in real use cases they are often presented with continuous audio requiring automatic (and sub-optimal) segmentation. After comparing existing techniques (VAD-based, fixed-length and hybrid segmentation methods), in this paper we propose enhanced hybrid solutions to produce better results without sacrificing latency. Through experiments on different domains and language pairs, we show that our methods outperform all the other techniques, reducing by at least 30% the gap between the traditional VAD-based approach and optimal manual segmentation.

1 Introduction

Speech-to-text translation (ST) consists in translating utterances in one language into text in another language. From the architectural standpoint, ST systems are traditionally divided in cascade and direct. Cascade solutions first transcribe the audio via automatic speech recognition (ASR) and then translate the generated transcripts with a machine translation (MT) component. In direct ST, a single end-to-end model operates without intermediate representations. This allows reducing error propagation and latency, as well as exploiting more information (e.g. speaker's vocal traits and prosody).

Different from MT, where sentence-level splits represent a natural (though not necessarily optimal) input segmentation criterion, handling audio data is more problematic. Existing training corpora (Cattoni et al., 2021; Iranzo-Sánchez et al., 2020) split continuous speech into utterances according to strong punctuation marks in the transcripts (which are known in advance), reflecting linguistic criteria related to sentence well-formedness. This (*manual*) segmentation is optimal, as it allows ST systems to potentially generate correct outputs even for languages with different syntax and word order (e.g. subject-verb-object vs subject-object-verb). At run-time, though, audio transcripts are not known in advance and *automatic* segmentation techniques have to be applied. The traditional approach is to adopt a Voice Activity Detection (VAD) tool to break the audio on speaker silences (Sohn et al., 1999), considered as a proxy of clause boundaries. However, since the produced segmentation is not driven by syntactic information (unlike that of the training corpora), final performance on downstream tasks degrades considerably (Sinclair et al., 2014).

The impact of a syntax-unaware segmentation can be limited in cascade systems by means of dedicated components that re-segment the ASR transcripts, so to feed MT with well-formed sentences (Matusov et al., 2006). The absence of intermediate transcripts makes this solution unfeasible for direct systems, whose performance is therefore highly sensitive to sub-optimal audio segmentation. This has been shown in the 2020 IWSLT evaluation campaign (Ansari et al., 2020), where the best direct ST system had a key feature in the segmentation algorithm (Potapczyk and Przybysz, 2020). In the same evaluation setting, the second-best direct system (Bahar et al., 2020) exploited an external ASR model to segment the audio (with a +10% BLEU gain compared to its VAD-based counterpart). This solution, however, formally makes it closer to a cascade architecture, losing the advantage of the reduced latency of direct systems. For this reason, while in §4 we compare with the state-of-the-art method proposed in (Potapczyk and Przybysz, 2020), we will not consider approaches needing additional models (e.g. ASR) like the one in (Bahar et al., 2020).

So far, no work analyzed in depth the strengths

and weaknesses of different audio segmentation methods in the context of direct ST. To fill this gap, we study the behavior of the existing techniques and, based on the resulting observations, we propose improved hybrid methods that can also be applied to streaming audio. Through experiments in two domains (TED and European Parliament talks) and two target languages (German and Italian), we show that our solutions outperform the others in all conditions, reducing the gap with optimal manual segmentation by at least 30% compared to VAD systems.

2 Audio Segmentation Methods

2.1 Existing Methods

VAD systems. VAD tools are classifiers that determine whether a given audio frame contains speech or not. Based on this, a VAD-based segmentation considers a sequence of consecutive speech frames as a segment, filtering out non-speech frames. In this work, we evaluate two widely used open source VAD tools: LIUM (Meignier and Merlin, 2010) and WebRTC's VAD.[1] For LIUM, we apply the configuration employed in the IWSLT campaign (Ansari et al., 2020). WebRTC takes as parameters the *frame size* (10, 20 or 30ms) and the *aggressiveness mode* (an integer in the range [0, 3], 3 being the most aggressive). We select three configurations based on the segmentation they produce on the MuST-C test set (Cattoni et al., 2021), one of the test sets used in our experiments (see §4). Specifically, we consider those not generating too many (more than two times the segments of the manual segmentation) or too long segments (more than 60s). They are: *(3, 30ms)*, *(2, 20ms)* and *(3, 20ms)*. The statistics computed for the two segmentation tools on the MuST-C test set are presented in Table 1, along with those corresponding to manual segmentation. To better understand the impact of different VADs on translation quality, the tools are compared on MuST-C and Europarl-ST data. Table 2 reports preliminary translation results for English-German (en-de) and English-Italian (en-it), obtained with the systems described in §3. LIUM and the most aggressive WebRTC configuration *(3, 20ms)* are significantly worse than the other two WebRTC configurations. As *(2, 20ms)* achieves comparable BLEU performance to *(3, 30ms)* on MuST-C and better on Europarl-ST, it is used in the rest of the paper.

[1] http://webrtc.org/. We use the Python interface http://github.com/wiseman/py-webrtcvad.

System	Man.	LIUM	WebRTC		
Aggress.			3	2	3
Frame size			30ms	20ms	20ms
% filtered	14.66	0.00	11.27	9.53	15.58
Num segm.	2,574	2,725	3,714	3,506	5,005
Max len (s)	51.97	18.63	48.84	58.62	46.76
Min len (s)	0.05	2.50	0.60	0.40	0.40
Avg len (s)	5.82	6.44	4.19	4.53	2.96

Table 1: Statistics for different segmentations of the MuST-C test set. "Man." = sentence-based segmentation. "% filtered" = percentage of audio discarded.

VAD System	**MuST-C**		**Europarl-ST**	
	BLEU (↑)	**TER (↓)**	**BLEU (↑)**	**TER (↓)**
English-German				
LIUM	19.55	76.21	15.39	94.06
WebRTC 3, 30ms	**21.90**	66.96	16.23	89.35
WebRTC 3, 20ms	19.48	72.25	14.07	99.32
WebRTC 2, 20ms	21.87	**66.72**	**18.51**	**78.12**
English-Italian				
LIUM	21.29	67.50	18.88	73.73
WebRTC 3, 30ms	**22.46**	**64.99**	19.85	72.28
WebRTC 3, 20ms	20.09	68.62	17.35	78.18
WebRTC 2, 20ms	22.34	66.12	**20.90**	**69.54**

Table 2: Results of the VAD systems on MuST-C and Europarl-ST for en-de and en-it.

Fixed-length. A simple approach is splitting the audio at a predefined segment length (Sinclair et al., 2014), without considering the content. In contrast with VAD, this naive method has the benefit of ensuring that the resulting segments are not too long or too short, which are typically hard conditions for ST systems. However, the split points are likely to break sentences in critical positions such as between a subject and a verb or even in the middle of a word. Unlike VAD, this method does not filter the non-speech frames from the input audio, which is entirely passed to the ST system. Fig. 1 shows that, with fixed segmentation, translation quality improves with the duration of the segments (slightly for values $>=$16s) up to 20s, after which it decreases. 20 seconds is the maximum segment length in our training data due to memory limits: we can conclude that longer segments produce better translations, but models can effectively translate only sequences whose length does not exceed the maximum observed in the training set.

SRPOL-like segmentation. The method described in (Potapczyk and Przybysz, 2020) takes into account both audio content (silences) and target segments' length (i.e. the desired length of the generated segments) to split the audio. It recursively divides the audio segments on the longest

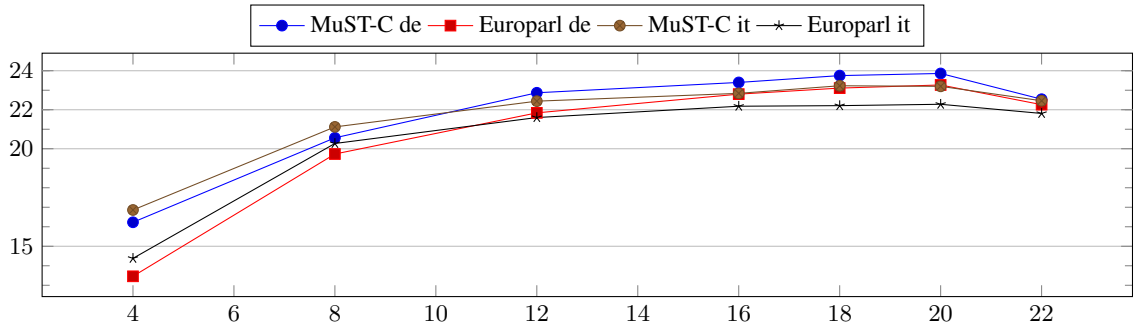


Figure 1: BLEU scores (Y axis) with different fixed-length segmentations (in seconds – X axis).

silence, until either there are no more silences in a segment, or the segment itself is shorter than a threshold. It is important to notice that, in (Potapczyk and Przybysz, 2020), the silences are detected with a manual operation, making the approach hard to reproduce and not scalable. In this paper, we replicate the logic, but we rely on WebRTC to automatically identify silences. For this reason, our results might be slightly different than the original ones, but the segmentation is automatic and easy to reproduce. Another major problem of this method is that it requires the full audio to be available for splitting it. So it is not applicable to audio streams and online use cases. Based on the previous considerations drawn from Fig.1, in our experiments we set the maximum length threshold to 20s, so that the model is fed with sequences that are not longer than the maximum seen at training time. The resulting segments have an average length of 7-8s.

2.2 Proposed hybrid segmentation

Similar to (Potapczyk and Przybysz, 2020), our method is hybrid as it considers both the audio content and the target segments' length. However, unlike (Potapczyk and Przybysz, 2020), we give more importance to the target segments' length than to the detected pauses (we motivate this choice in §4). Specifically, we split on the longest pause in the interval (minimum and maximum length), if any, otherwise we split at maximum length. Maximum and minimum segment lengths are controlled by two hyper-parameters (*MAX_LEN* and *MIN_LEN*). Unlike the SRPOL-like approach, ours can operate on audio streams, as it does not require the full audio to start the segmentation procedure. Moreover, the latency is controlled by *MAX_LEN* and *MIN_LEN*, which can be tuned to trade translation quality for lower latency.

We tested different values for *MIN_LEN* and we chose 17s for our experiments, because it resulted in the best score on the MuST-C dev set. As in the other methods, and for the same reasons, *MAX_LEN* is set to 20s. The resulting segments have an average length slightly higher than 17s.

We also introduce a variant of this method that enforces splitting on pauses longer than *550ms*. In (Karakanta et al., 2020), this threshold is shown to often represent a *terminal juncture*: a break between two utterances, usually corresponding to clauses. Splitting on such pauses should hence enforce separating different clauses. As a result, segments can be shorter than *MIN_LEN*, but we still ensure they are not longer than *MAX_LEN*. With this variant, the segments are much shorter, as their average length is 8s, similar to the SRPOL-like technique.

3 Experimental Settings

We use a Transformer (Vaswani et al., 2017) whose encoder is modified for ST. The encoder starts with two 2D convolutional layers that reduce the length of the Mel-filter-bank sequence by a factor of 4. The resulting tensors are passed to a linear layer that maps them into the dimension used by the following encoder Transformer layers. A logarithmic distance penalty (Di Gangi et al., 2019) is applied in all the encoder Transformer layers.

The ST models have 11 encoder Transformer layers and 4 decoder Transformer layers. We use 8 attention heads, 512 attention hidden units and 2,048 features in the FFNs' hidden layer. We set dropout to 0.1. The optimizer is Adam (Kingma and Ba, 2015) with betas *(0.9, 0.98)*. The learning rate is scheduled with inverse square root decay after 4,000 warm-up updates, during which it increases linearly from $3 \cdot 10^{-4}$ up to $5 \cdot 10^{-3}$. The update frequency is set to 8 steps; we train on 8

Segm. method	MuST-C en-de		Europarl en-de		MuST-C en-it		Europarl en-it	
	BLEU (↑)	TER (↓)	BLEU (↑)	TER (↓)	BLEU (↑)	TER (↓)	BLEU (↑)	TER (↓)
Manual segm.	27.55	58.84	26.61	60.99	27.70	58.72	28.79	59.16
Best VAD	21.87	66.72	18.51	78.12	22.34	66.12	20.90	69.54
Best Fixed (20s)	23.86	**61.29**	23.27	64.01	23.20	64.24	22.28	64.57
SRPOL-like	22.26	71.10	20.49	77.61	23.12	66.27	23.26	66.19
Pause in 17-20s	**24.39**	61.35	**23.78**	**63.15**	**23.50**	**63.76**	22.86	63.44
+ force split	23.17	66.20	22.52	68.56	23.45	63.79	**24.15**	**63.31**

Table 3: Comparison between manual and automatic segmentations: VAD, fixed-length and hybrid approaches.

GPUs and each mini-batch is limited to 12,000 tokens or 8 sentences, so the resulting batch size is slightly lower than 512. We initialize the convolutional and the first encoder Transformer layers with the encoder of a model trained on ASR data. We pre-train our ST models on the ASR corpora with synthetic targets generated by an MT model fed with the known transcripts (Jia et al., 2019) and we fine-tune on the ST corpora. Both these trainings adopt knowledge distillation (Hinton et al., 2015) with the MT model as teacher (Liu et al., 2019; Gaido et al., 2020). Finally, we fine-tune on the ST corpora with label-smoothed cross-entropy (Szegedy et al., 2016). In all the three steps, we use SpecAugment (Park et al., 2019) and time stretch (Nguyen et al., 2020) as data augmentation techniques.

The ASR model is similar to ST models, but we use 8 encoder layers and 6 decoder layers. For MT, instead, the Transformer attentions has 16 heads and hidden-layer features are two times those of ST and ASR models.

We experimented with translation from English speech into two target languages: German and Italian. To train the MT model used for knowledge distillation, we employed the WMT 2019 datasets (Barrault et al., 2019) and the 2018 release of OpenSubtitles (Lison and Tiedemann, 2016) for English-German and OPUS (Tiedemann, 2016) for English-Italian. All the data were cleaned with Modern MT (Bertoldi et al., 2017). The ASR model, whose encoder was used to initialize that of the ST model, was trained on TED-LIUM 3 (Hernandez et al., 2018), Librispeech (Panayotov et al., 2015), Mozilla Common Voice[2], How2 (Sanabria et al., 2018), and the audio-transcript pairs of the ST corpora. The ST corpora were MuST-C (Cattoni et al., 2021) and Europarl-ST (Iranzo-Sánchez et al., 2020) for both target languages. We filtered out samples with input audio longer than 20s to avoid out-of-memory errors. The text is encoded using BPE (Sennrich et al., 2016) with 8,000 merge rules (Di Gangi et al., 2020).

[2]https://voice.mozilla.org/

4 Results

We compute ST results in terms of BLEU (Papineni et al., 2002) and TER (Snover et al., 2006) on the MuST-C and Europarl-ST test sets for en-de and en-it. In MuST-C the two test sets are identical with regard to the audio, while in Europarl-ST they contain different recordings.

As shown in Table 3, fixed-length segmentation always outperforms the best VAD, both in terms of BLEU and TER. This may be surprising but it confirms previous findings in ASR (Sinclair et al., 2014): also in ST, VAD is more costly and less effective than a naive fixed-length segmentation. Besides, it suggests that the resulting segments' length is more important than the precision of the split times. This observation motivates the definition of our proposed techniques. Compared to fixed-length segmentation, the SRPOL-like method provides better results for en-it, but worse for en-de, indicating that the syntactic properties of the source and target languages are an important factor for audio segmentation (see §5).

Our proposed method (*Pause in 17-20s* in Table 3) outperforms the others on all test sets but Europarl en-it, in which SRPOL-like has a higher BLEU (but worse TER). The version with forced splits on 550ms pauses is inferior to the version without forced splits on the German test sets, but it is on par for MuST-C en-it and superior on Europarl en-it, on which it is the best segmentation overall by a large margin. Moreover, its scores are always better than the ones obtained by the SRPOL-like approach, although the length of the produced segments is similar. These results suggest that, although the best version depends on the syntax and the word order of the source and target languages, our method can always outperform the others in terms of both BLEU and TER. Noticeably, it does not introduce latency, since it does not require the

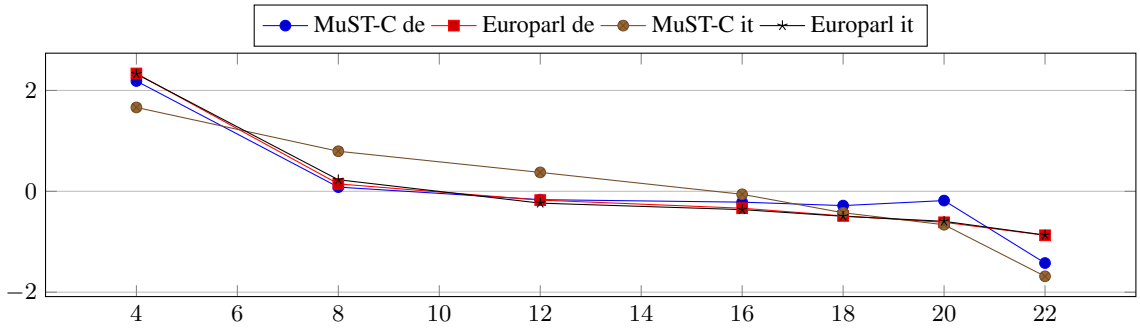


Figure 2: Z-score normalized output lengths (number of words) according to the input segments length.

(a) Hallucinations with non-speech audio	
Audio	*Music and applause.*
4s segments	**[Chinesisch] [Hawaiianischer Gesang] // Chris Anderson: Du bist ein Idiot. // Nicole: Nein.** *[Chinese] [Hawaiian song] // Chris Anderson: You are an idiot. // Nicole: No.*
(b) Hallucinations with sub-sentential utterances	
Audio	Now, chimpanzees are well-known for their aggression. // (Laughter) // But unfortunately, we have made too much of an emphasis of this aspect (...)
Reference	Schimpansen sind bekannt für ihre Aggressivität. // (Lachen) // Aber unglücklicherweise haben wir diesen Aspekt überbetont (...)
4s segments	**Publikum: Nein.** Schimpansen sind bekannt. // **Ich bin für** ihre Aggression gegangen. // Aber leider haben **wir zu viel Coca-Cola gemacht**. // **Das ist eine** wichtige Betonung dieses Aspekts (...) ***Audience: No.*** *Chimpanzees are known.* ***// I went for*** *their aggression.* *// But unfortunately* ***we made too much Coca-Cola.*** *//* ***This is an*** *important emphasis of this aspect (...)*
20s segments	Schimpansen sind bekannt für ihre Entwicklung. // Aber leider haben wir zu viel Schwerpunkt auf diesem Aspekt (...) *Chimpanzees are known for their development. // But unfortunately, we have expressed too much emphasis on this aspect (...)*
(c) Hallucinations and bad translation with sub-sentential utterances	
Audio	(...) where the volunteers supplement a highly skilled career staff, you have to get to the fire scene pretty early to get in on any action.
Reference	(...) in der Freiwillige eine hochqualifizierte Berufsfeuerwehr unterstützten, muss man ziemlich früh an der Brandstelle sein, um mitmischen zu können.
4s segments	(...) **wo die Bombenangriffe auf dem Markt waren. // Man muss bis zu 1.000 Angestellte in die USA, nach Nordeuropa kommen**. *(...)* ***where the bombings were on the market. // You have to come up to 1,000 employees in the USA, to Northern Europe****.*
20s segments	(...) in der die Freiwilligen ein hochqualifiziertes Karriere-Team ergänzen, muss man ziemlich früh an die Feuerszene kommen, um in irgendeiner Aktion zu gelangen. *(...) where the volunteers complement a highly qualified career team, you have to get to the fire scene pretty early in order to get into any action.*
(d) Final portions of long segment ignored	
Audio	But still it was a real footrace against the other volunteers to get to the captain in charge to find out what our assignments would be. // When I found the captain, (...)
Reference	Aber es war immer noch ein Wettrennen gegen die anderen Freiwilligen **um den verantwortlichen Hauptmann zu erreichen und herauszufinden was unsere Aufgaben sein würden**. // Als ich den Hauptmann fand (...)
22s segments	(...) Es war immer noch ein echtes Fussrennen gegen die anderen Freiwilligen. // Als ich den Kapitän fand, (...) *(...) It was still a real footrace against the other volunteers. // When I found the captain, (...)*

Table 4: Translations affected by errors caused by too short – *(a)*, *(b)*, *(c)* – or too long – *(d)* – segments. The symbol "//" refers to a break between two segments. The breaks might be located in different positions in the different segmentations. Over-generated – in examples *(a)*, *(b)*, *(c)* – and missing – in *(d)* – content is marked in **bold** respectively in system's outputs and in the reference.

full audio to be available for splitting it, as the SR-POL-like technique does. In particular, averaged on the two domains, our best results (respectively with and without *forced splits*) reduce the gap with the manual segmentation by 54.71% (en-de) and 30.95% (en-it) compared to VAD-based segmentation.

5 Analysis

A first interesting consideration regards the length of the produced translations. In particular, we analyze the case of fixed-length segmentation (see Fig. 2): in presence of short input segments the output is longer, while it gets shorter in case of seg-

ments longer than 20s. To understand this behavior, we performed a manual inspection of the German translations produced by fixed-length segmentation with 4s, 20s and 22s.

The analysis revealed two main types of errors: overly long (*hallucinations* (Lee et al., 2018)) and overly short outputs. The first type of error occurs when the system is fed with small, sub-sentential segments. In this case, trying to generate well-formed sentences, the system "completes" the translation with text that has no correspondence with the input utterance. The second type of error occurs when the system is fed with segments that exceed the maximum length observed in the training data. In this case, part of the input (even complete clauses, typically towards the end of the utterance) is not realized in the final translation.

Table 4 provides examples of all these phenomena. The first three examples show cases of hallucinations in short (4s) segments, while the last one shows an incomplete translation of a long (22s) segment. In particular:

(a) shows the generation of text not related to the source when the audio contains only noise or silence (e.g. at the beginning of a TED talk recording).

(b) presents the addition of non-existing content in the translation of a sub-sentential segment.

(c) is related to a sub-sentential utterance as well, but in this case the output of the system is affected by both hallucinations and poor translation quality due to the lack of enough context.

(d) reports a segment whose last portion ignored.

The length of the generated outputs also helps understanding the different results obtained by the variants of our method on the two target languages. Indeed, the introduction of forced splits (+ *force split*) produces audio segments that are much shorter ($\sim$8s vs $\sim$17s) and hence, according to the previous consideration, the resulting translation is overall longer. For German, the difference in terms of output length is high ($> 8.5\%$), while for Italian it is much lower (4.33% on MuST-C and 2.49% on Europarl-ST). So, the German results are penalized by the additional hallucinations, while, for Italian translations, the beneficial separation of clauses delimited by terminal juncture dominates.

This different behavior relates to the different syntax of the source and target languages. Indeed, translating from English (an SVO language) into German (an SOV language) requires long-range reorderings (Gojun and Fraser, 2012), which can also span over sub-clauses. The Italian phrase structure, instead, is more similar to English. This is confirmed by the shifts counted in TER computation, which are 20% more in German than in Italian. Moreover, in Italian their number does not change between our method with and without forced splits, while in German the version with forced splits has 5-10% more shifts.

6 Conclusions

We studied different segmentation techniques for direct ST. Despite its wide adoption, VAD-based segmentation resulted to be underperforming. We showed that audio segments' length is a crucial factor to obtain good translations and that the best segmentation approach depends on the structural similarity between the source and target languages. In particular, we demonstrated that the resulting segments should be neither longer than the maximum length of the training samples nor too short (especially when the target language has a different structure). Inspired by these findings, we proposed two variants of a hybrid method that significantly improve on different test sets and languages over the VAD baseline and the other techniques presented in literature. Our approach was designed to be also applicable to audio streams and to allow controlling latency, hence being suitable even for online use cases.

References

Ebrahim Ansari, Amittai Axelrod, Nguyen Bach, Ondřej Bojar, Roldano Cattoni, Fahim Dalvi, Nadir Durrani, Marcello Federico, Christian Federmann, Jiatao Gu, Fei Huang, Kevin Knight, Xutai Ma, Ajay Nagesh, Matteo Negri, Jan Niehues, Juan Pino, Elizabeth Salesky, Xing Shi, Sebastian Stüker, Marco Turchi, Alexander Waibel, and Changhan Wang. 2020. FINDINGS OF THE IWSLT 2020 EVALUATION CAMPAIGN. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 1–34, Online. Association for Computational Linguistics.

Parnia Bahar, Patrick Wilken, Tamer Alkhouli, Andreas Guta, Pavel Golik, Evgeny Matusov, and Christian Herold. 2020. Start-Before-End and End-to-End: Neural Speech Translation by AppTek and RWTH Aachen University. In *Proceedings of 17th International Workshop on Spoken Language Translation (IWSLT)*, Virtual.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Gra-

ham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Nicola Bertoldi, Roldano Cattoni, Mauro Cettolo, Amin Farajian, Marcello Federico, Davide Caroselli, Luca Mastrostefano, Andrea Rossi, Marco Trombetti, Ulrich Germann, and David Madl. 2017. MMT: New Open Source MT for the Translation Industry. In *Proceedings of the 20th Annual Conference of the European Association for Machine Translation (EAMT)*, pages 86–91, Prague, Czech Republic.

Roldano Cattoni, Mattia A. Di Gangi, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2021. MuST-C: A multilingual corpus for end-to-end speech translation. *Computer Speech & Language*, 66:101155.

Mattia A. Di Gangi, Marco Gaido, Matteo Negri, and Marco Turchi. 2020. On target segmentation for direct speech translation. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 137–150, Virtual. Association for Machine Translation in the Americas.

Mattia A. Di Gangi, Matteo Negri, and Marco Turchi. 2019. Adapting Transformer to End-to-End Spoken Language Translation. In *Proceedings of Interspeech 2019*, pages 1133–1137.

Marco Gaido, Mattia A. Di Gangi, Matteo Negri, and Marco Turchi. 2020. On Knowledge Distillation for Direct Speech Translation. In *Proceedings of the Seventh Italian Conference on Computational Linguistics, CLiC-it 2020, Bologna, Italy, March 1-3, 2021*, volume 2769 of *CEUR Workshop Proceedings*.

Anita Gojun and Alexander Fraser. 2012. Determining the placement of German verbs in English–to–German SMT. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 726–735, Avignon, France.

François Hernandez, Vincent Nguyen, Sahar Ghannay, Natalia Tomashenko, and Yannick Estève. 2018. TED-LIUM 3: Twice as Much Data and Corpus Repartition for Experiments on Speaker Adaptation. In *Proceedings of the Speech and Computer - 20th International Conference (SPECOM)*, pages 198–208, Leipzig, Germany. Springer International Publishing.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the Knowledge in a Neural Network. In *Proceedings of NIPS Deep Learning and Representation Learning Workshop*, Montréal, Canada.

Javier Iranzo-Sánchez, Joan A. Silvestre-Cerdà, Javier Jorge, Nahuel Roselló, Giménez. Adrià, Albert Sanchis, Jorge Civera, and Alfons Juan. 2020. Europarl-ST: A Multilingual Corpus For Speech Translation Of Parliamentary Debates. In *Proceedings of 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8229–8233, Barcelona, Spain.

Ye Jia, Melvin Johnson, Wolfgang Macherey, Ron J. Weiss, Yuan Cao, Chung-Cheng Chiu, Naveen Ari, Stella Laurenzo, and Yonghui Wu. 2019. Leveraging Weakly Supervised Data to Improve End-to-End Speech-to-Text Translation. In *Proceedings of the 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7180–7184, Brighton, UK.

Alina Karakanta, Matteo Negri, and Marco Turchi. 2020. Is 42 the Answer to Everything in Subtitling-oriented Speech Translation? In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 209–219, Online.

Diederik Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of 3rd International Conference on Learning Representations (ICLR)*, San Diego, California.

Katherine Lee, Orhan Firat, Ashish Agarwal, Clara Fannjiang, and David Sussillo. 2018. Hallucinations in Neural Machine Translation. In *Proceedings of NIPS 2018 Interpretability and Robustness for Audio, Speech and Language Workshop*, Montréal, Canada.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles. In *Proceedings of the Tenth International Language Resources and Evaluation Conference (LREC)*, pages 923–929, Portoroz, Slovenia.

Yuchen Liu, Hao Xiong, Jiajun Zhang, Zhongjun He, Hua Wu, Haifeng Wang, and Chengqing Zong. 2019. End-to-End Speech Translation with Knowledge Distillation. In *Proceedings of Interspeech 2019*, pages 1128–1132, Graz, Austria.

Evgeny Matusov, Arne Mauser, and Hermann Ney. 2006. Automatic Sentence Segmentation and Punctuation Prediction for Spoken Language Translation. In *Proceedings of International Workshop on Spoken Language Translation (IWSLT) 2006*, pages 158–165, Kyoto, Japan.

Sylvain Meignier and Teva Merlin. 2010. LIUM SpkDiarization: An Open Source Toolkit For Diarization. In *Proceedings of the CMU SPUD Workshop*, Dallas, Texas.

Thai-Son Nguyen, Sebastian Stueker, Jan Niehues, and Alex Waibel. 2020. Improving Sequence-to-sequence Speech Recognition Training with On-the-fly Data Augmentation. In *Proceedings of the 2020*

International Conference on Acoustics, Speech, and Signal Processing – IEEE-ICASSP-2020, Barcelona, Spain.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An ASR corpus based on public domain audio books. In *Proceedings of 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210, South Brisbane, Queensland, Australia.

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.

Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. 2019. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. In *Proceedings of Interspeech 2019*, pages 2613–2617, Graz, Austria.

Tomasz Potapczyk and Pawel Przybysz. 2020. SR-POL's system for the IWSLT 2020 end-to-end speech translation task. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 89–94, Online. Association for Computational Linguistics.

Ramon Sanabria, Ozan Caglayan, Shruti Palaskar, Desmond Elliott, Loïc Barrault, Lucia Specia, and Florian Metze. 2018. How2: A Large-scale Dataset For Multimodal Language Understanding. In *Proceedings of Visually Grounded Interaction and Language (ViGIL)*, Montréal, Canada. Neural Information Processing Society (NeurIPS).

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Mark Sinclair, Peter Bell, Alexandra Birch, and Fergus Mcinnes. 2014. A semi-Markov model for speech segmentation with an utterance-break prior. In *Proceedings of Interpseech 2014*, pages 2351–2355, Singapore.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas,*, pages 223–231, Cambridge.

Jongseo Sohn, Nam Soo Kim, and Wonyong Sung. 1999. A statistical model-based voice activity detection. *IEEE Signal Processing Letters*, 6(1):1–3.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the Inception Architecture for Computer Vision. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, Las Vegas, Nevada, United States.

Jörg Tiedemann. 2016. OPUS – Parallel Corpora for Everyone. *Baltic Journal of Modern Computing*, page 384. Special Issue: Proceedings of the 19th Annual Conference of the European Association of Machine Translation (EAMT).

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 *Proceedings of Advances in Neural Information Processing Systems 30 (NIPS)*, pages 5998–6008, Long Beach, California.

A Sample-Based Training Method for Distantly Supervised Relation Extraction with Pre-Trained Transformers

Mehrdad Nasser,[1] Mohamad Bagher Sajadi,[2] and Behrouz Minaei-Bidgoli[1]
[1]School of Computer Engineering, Iran University of Science and Technology
[2]Central Tehran Branch, Islamic Azad University
mehrdadnasser94@gmail.com, moh.sajadi.eng@iauctb.ac.ir
b_minaei@iust.ac.ir

Abstract

Multiple instance learning has become the standard learning paradigm for distantly supervised relation extraction. However, relation extraction is performed at bag level in this learning paradigm and thus has significant hardware requirements for training when coupled with large sentence encoders such as deep transformer neural networks. In this paper, we propose a novel sample-based training method for distantly supervised relation extraction that relaxes these hardware requirements. In the proposed method, we limit the number of sentences in a batch by randomly sampling sentences from the bags in the batch. However, this comes at the cost of losing valid sentences from bags. To alleviate the issues caused by random sampling, we use an ensemble of trained models for prediction. We demonstrate the effectiveness of our approach by using our proposed learning setting to fine-tuning BERT on the widely NYT dataset. Our approach significantly outperforms previous state-of-the-art methods in terms of AUC and P@N metrics.

1 Introduction

Relation extraction (RE) is an essential part of Natural Language Processing (NLP) and benefits downstream tasks such as knowledge base population. The main goal of RE is to identify the semantic relationship between two entities in text. For example, based on the sentence " ***Elon Musk*** is the founder of ***SpaceX***", entities ***Elon Musk*** and ***SpaceX*** express the *founderOf* relation. Conventional supervised relation extraction methods rely on manually labeled datasets (Hendrickx et al., 2010; Walker et al., 2006) for training. The construction of such datasets on a large scale requires considerable human effort and is often impractical. Distant supervision for relation extraction (Mintz et al., 2009) addresses this problem by automatically labeling entity pairs in a sentence based on their relationship in a knowledge base, such as freebase (Bollacker et al., 2008), removing the need for manual labeling. However, not every pair of entities in a sentence express their corresponding relation in a knowledge base; thus, distant supervision suffers from noisy labels. For example, if (***Elon Musk***, *CEOof*, ***SpaceX***) is a fact in knowledge base , distant supervision would label the aforementioned example sentence as *CEOof*, which would be an incorrect label.

Recent works have adopted multiple instance learning (MIL) framework, along with additional denoising methods to address the noisy labeling problem. In MIL, each sample in the dataset is a bag of sentences that share the same entity pair, as opposed to the conventional supervised relation extraction methods where each instance is a single sentence. Additional denoising steps, such as selective attention (Lin et al., 2016), are then taken to aggregate all the sentences in a bag into a single high-quality representation for that bag.

Distant supervision is used to generate large-scale datasets, and thus some bags will consist of a large number of sentences. These bags can not be split into multiple batches and have to be processed at once during training to construct a single bag-level representation. As a result, MIL is more resource-intensive than fully supervised relation extraction. For example, if there are bags with more than 100 sentences in the training dataset, even by setting the batch size to 1, we require enough hardware memory to pass at least 100 sentences through the sentence encoder to get the bag representation in a step of training. Due to the aforementioned problem, current state-of-the-art methods adopt light-weight and efficient deep neural networks, such as convolutional neural networks (CNN), as the sentence encoder and focus mainly on mitigating the noisy labeling problem.

Deep Transformer neural networks (Vaswani et al., 2017) pre-trained on large corpora (De-

vlin et al., 2019; Radford et al., 2018, 2019) have demonstrated superior capabilities in capturing a contextual semantic representation of words and have achieved state-of-the-art results in many NLP tasks, including supervised relation extraction (Soares et al., 2019; Wu and He, 2019). However, these models often have a large number of parameters. They have been shown to capture even better representations as they increase in size (Radford et al., 2019; Brown et al., 2020), and thus have significant hardware requirements when training under the MIL framework. To address this issue, we propose a new training method for distantly supervised relation extraction (DSRE). Unlike previous methods that use all the sentences in bags to construct the bag representations, we propose a random instance sampling (RIS) method that limits the number of sentences in a mini-batch by randomly sampling sentences from bags in the mini-batch. Limiting the total number of sentences allows us to leverage deep language representation models such as BERT (Devlin et al., 2019) as the sentence encoder despite limited hardware and produce higher quality sentence representations. However, due to this approach's randomness, using RIS will result in less robust predictions in the inference stage. To mitigate this issue, we train multiple models and then use an ensemble of these models in the inference stage by averaging over prediction probabilities. We adopt selective attention as the denoising mechanism and use BERT to encode the relation between entity pairs in sentences.

The contributions of this paper can be summarized as follows:

- We propose a new training method for DSRE that relaxes the hardware requirements of MIL by using a random subset of bags in the training phase. This results in a smaller number of sentences in a batch of bags and thus allows us to use larger transformer models as sentence encoders in the distantly supervised setting.
- We present two sampling methods for RIS, a baseline that preserves the relative size of the bags after sampling and another method that samples an equal number of sentences from all bags in a mini-batch regardless of their sizes. Our experiments demonstrate the superiority of the latter sampling approach.
- We propose the use of an ensemble of different trained models to mitigate the effects of randomness in our new training method. Our experiments demonstrate the effectiveness of this approach.
- We use our new training method, coupled with selective attention for bag denoising, to fine-tune BERT on the widely used NYT dataset. Our model achieves an AUC value of 61.4, significantly outperforming previous state-of-the-art methods despite using a simple denoising method.

2 Related Work

Mintz et al. (2009) proposed distant supervision as a way to generate labels for large-scale data for relation extraction automatically. This was done by aligning entities in a knowledge base. However, some of these labels did not match the relation expressed by their corresponding sentences, and thus these noisy labels became the main challenge in DSRE. Subsequent works adopted the MIL paradigm to alleviate the noisy label problem (Riedel et al., 2010; Hoffmann et al., 2011; Surdeanu et al., 2012), which considered each bag as a sample instead of each sentence. However, these methods used hand-crafted features to encode sentences into vector representations, which limited their performance.

Zeng et al. (2015) adopted the piecewise convolutional neural network as the sentence encoder and selected only a single sentence in each bag to use as bag-level representation. Lin et al. (2016) proposed selective attention, which uses a weighted average of sentence representations as the bag-level representation. Liu et al. (2017) proposed a soft-label method in which bag labels could change depending on the bag-level representation. Qin et al. (2018) and Feng et al. (2018) both trained reinforcement learning agents to detect and remove or relabel noisy sentences in bags. Ye and Ling (2019) proposed two novel attention mechanisms, intra-bag attention that considers all relations instead of just the bag's relation to compute the bag-level representation, and inter-bag attention mechanism that aggregates multiple bag representations into a single representation to alleviate the noisy bag problem, i.e., bags with all noisy sentences.

Previous works have incorporated the self-attention mechanism (Vaswani et al., 2017) in their methods (Huang and Du, 2019; Li et al., 2020) to address the limitations of piecewise convolutional

neural networks (PCNN) in learning sentence representations.

Alt et al. (2019) extended the OpenAI Generative Pre-trained Transformer (GPT) (Radford et al., 2018) to bag-level relation extraction and used selective attention to compute bag representations. They chose GPT over other transformer models like BERT due to its more reasonable hardware requirements. Our method is similar to Alt et al. (2019) as we leverage a pre-trained transformer neural network in the distantly supervised setting and use selective attention as the denoising method. However, we use our proposed RIS module during training, which allows us to use BERT as a sentence encoder while requiring much less memory during training.

3 Proposed Method

In this section, we present the different steps of the training procedure in our proposed method. In the standard distant supervision setting, the sentences of each bag in a mini-batch are transformed into vector representations using sentence encoders. In the next step, each bag is transformed into a single representation using a denoising method and is then fed into a classification layer. In our proposed training method, a new RIS step is added to the beginning. The new bags computed using RIS are used in the encoding phase instead of the original ones. An essential property of RIS is that it is independent of other steps of training and thus can be integrated into any other distant supervision method. Overview of our proposed training method is demonstrated in figure 1.

3.1 Random Instance Sampling

To address the issue of hardware resource requirements of the MIL framework for distant supervision, we propose a new method called Random Instance Sampling (RIS). Unlike standard supervised relation extraction, batch size in bag-level relation extraction does not control the number of sentences in a mini-batch but only determines the number of bags. Thus, if we are using a sentence encoder with a large number of parameters, such as BERT, it is important to have control over the maximum number of sentences in each step of training to avoid exceeding the available memory. Let $\mathrm{B} = \{B_1, ..., B_n\}$ denote a mini-batch of bags in a step of training, and n is the batch size and N_{max} denote the maximum number of sentences allowed in a single batch. If the total number of sentences in a mini-batch is less than N_{max}, the output mini-batch of RIS will be the same as the original one.

$$N_{\mathrm{B}} = Sum\{Size(B_1), ..., Size(B_n)\} \quad (1)$$

where $Size(B_i)$ denotes the size of bag B_i and N_{B} is the total number of sentences in a mini-batch. However, if $N_{\mathrm{B}} > N_{\text{max}}$, due to memory limitations, we will not be able to pass all the sentences through the sentence encoder. Here, we propose two variations of RIS. A baseline method and another approach that improves upon the baseline.

3.1.1 RIS-baseline

In the first variation, all bags participate in the sampling regardless of their sizes and the relative bag sizes in a batch are preserved. For all bags, we sample fraction f of their sentences, where f is computed by simply dividing N_{max} by N_{B}. Formally, for each mini-batch B in every step of training, RIS-baseline creates a new mini-batch $\mathrm{B}' = \{B'_1, ..., B'_n\}$ as follows:

$$f = \frac{N_{\text{max}}}{N_{\mathrm{B}}} \quad (2)$$

$$t_i = Size(B_i) \times f \quad (3)$$

$$B'_i = Sample(B_i, t_i) \quad (4)$$

where t_i denotes how many sentences should be sampled from bag B_i and B'_i is the new bag after randomly sampling t_i sentences from bag B_i and mini-batch B' will be used instead of mini-batch B in the current step of training. While this is a simple and straightforward approach, it has a significant shortcoming. Distant supervision for relation extraction is based on the assumption that at least one valid sentence exists in each bag. When sampling sentences from small and large bags with equal fractions, the probability that valid sentences in the small bags get removed due to the sampling is higher than that of larger bags, for example, if we have bags with 14 and 2 sentences in a batch of 2 bags with $N_{\text{max}} = 8$, then using the RIS-baseline approach we will have bags with 7 and 1 sentences respectively. The probability of removing valid sentences when sampling 1 sentence from a bag of size 2 is higher than when sampling 7 sentences from a bag of size 14. Moreover, the number of valid sentences in bags is usually low, and thus the

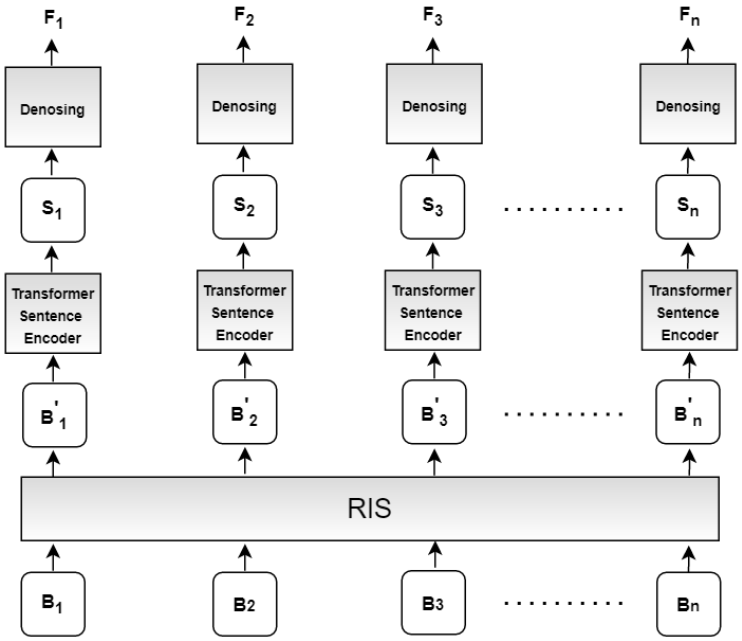


Figure 1: Overview of our proposed sample-based training method

probability of forming noisy bags (bags without valid sentences) due to sampling is higher for small bags.

3.1.2 RIS-equal

To address the aforementioned issues of RIS-baseline, we propose another approach called RIS-equal. In this approach, we sample an equal number of sentences, denoted by N_{sample}, from each bag regardless of its size, and thus smaller bags will lose a smaller proportion of their sentences due to sampling than larger bags and bags with sizes less than the aforementioned fixed number will not participate in the sampling. As a result, the overall probability of losing valid sentences will be less than that of RIS-baseline. We set N_{sample} to the maximum possible value, which can be calculated by dividing N_{max} by the batch size. After the sampling, the total number of sentences in a mini-batch will be less than N_{max} if there are bags whose sizes are smaller than N_{sample}. Let N_{diff} be the difference between N_{max} and the total number sentences after sampling N_{sample} from each bag. If N_{diff} is non-zero, we can increase the size of bags whose original sizes before sampling were larger than N_{sample} by sampling sentences from their corresponding leftover bags (remaining sentences in bags after sampling N_{sample} sentences from them) and adding these sentences to them. We set the number of sentences sampled from each leftover bag to be proportional to the leftover bag size.

Formally, for each mini-batch B in every step of training, RIS-equal creates a new mini-batch $\text{B}' = \{B'_1, ..., B'_n\}$ as follows:

$$N_{\text{sample}} = \frac{N_{\text{max}}}{n} \tag{5}$$

$$c_i = \left\lfloor \frac{Max\{0, Size(B_i) - N_{\text{sample}}\}}{\sum_j Max\{0, Size(B_j) - N_{\text{sample}}\}} \right\rfloor \times N_{\text{diff}} \tag{6}$$

$$t_i = N_{\text{sample}} + c_i \tag{7}$$

$$B'_i = Sample(B_i, t_i) \tag{8}$$

where N_{sample} denotes how many sentences we initially sample from each bag, t_i denotes how many sentences should be sampled from bag b_i if N_{diff} is non-zero and we can sample extra sentences from the leftover bags. Finally, B'_i is the new bag after randomly sampling t_i sentences from bag B_i and mini-batch B' will be used instead of batch B in the current step of training.

3.2 Sentence Encoder

We follow the approach of Soares et al. (2019) to encode sentences into relation representations using deep transformers. Similar to Soares et al. (2019), we adopt BERT (Devlin et al., 2019) for encoding sentences. After tokenizing the sentences

in the dataset, each sentence X can be represented as a sequence of tokens as follows:

$$X = [x_0...x_i...x_j...x_l...x_m...x_n]$$

where $x_0 = [CLS]$ and $x_n = [SEP]$ are special tokens indicating the start and end of the sequence. Sequences $[x_i...x_j]$ and $[x_l...x_m]$ represent tokens for head and tail entities respectively. BERT's output hidden state corresponding to the $[CLS]$ token is used as the sentence representation in task such as sentiment analysis (Devlin et al., 2019). However, it is not a good representation for relations as it does not make use of the position of entity tokens. Soares et al. (2019) propose adding special markers before and after head and tail entities as follows:

$$X' = [x_0...[H_1]x_i...x_j[H_2]...[T_1]x_l...x_m[T_2]...x_n]$$

Let S_H and S_T denote the output hidden states corresponding to $[H_1]$ and $[T_1]$ respectively. Final vector representation of the relation expressed by each sentence will be computed by concatenating S_H and S_T into a single vector S. Formally, each bag $B = \{X_1, ..., X_m\}$ will become a bag of sentence representations S $= \{S_1, ..., S_m\}$ after the encoding stage.

3.3 Selective Attention

In order to train a relation classifier in the MIL framework for distant supervision, we need to compute a vector representation for each bag in the dataset. Following Lin et al. (2016) and Alt et al. (2019), we use selective attention to aggregate sentence representations in a bag into a single bag representation. Let S denote a bag of sentence representations that mention the same entity pair and r be the corresponding label provided by distant supervision. Selective attention assigns a weight to each sentence representation in a bag and then computes a weighted average of them to produce the final bag representation. Valid sentences are assigned higher weights, and thus contribute more to the final bag representation whereas noisy sentences will receive lower weights. Final representation of a bag using selective attention can be formulated as follows:

$$\beta_i = S_i \mathbf{r} \tag{9}$$

$$\alpha_i = \frac{exp(\beta_i)}{\sum_j exp(\beta_j)} \tag{10}$$

$$F = \sum_i \alpha_i S_i \tag{11}$$

where $\mathbf{r}$ is a learnable embedding for relation r, β_i is the similarity score between r and sentence representation S_i, α_i is the weight assigned to S_i and F is the representation of bag S.

Each bag representation is then fed into a dense layer with softmax activation to compute the probability distribution over all the relations.

$$d = \mathbf{W}F + \mathbf{b} \tag{12}$$

$$P(\mathbf{r}|S, \theta) = softmax(d) \tag{13}$$

where $\mathbf{W}$ and $\mathbf{b}$ are the learnable parameters of the Dense layer, θ denotes the model's parameters and $P(r|S, \theta)$ is the probability distribution over relation labels.

We formulate the objective function of the training as follows:

$$\mathcal{J}_D = \sum_i^{|D|} P(r_i|\mathrm{S}_i, \theta) \tag{14}$$

Where $|D|$ denotes the number of bags in the training set.

3.4 Ensemble Modeling for Prediction

Using RIS for training will cause two main issues:

- During training, we apply RIS to each mini-batch at each training step. Thus some valid sentences may not participate in the construction of their corresponding bag's representation. This negatively affects the model's convergence in training and thus, impacts the model's performance in the evaluation stage.
- Each time we train the model, it is practically trained on a slightly different dataset due to RIS. Thus, the performance will vary each time we train the model from scratch.

Due to the issues mentioned above, we propose training several models and using an ensemble of those models for evaluation. An ensemble method mitigates the first issue because specific valid sentences that are ignored during training for one model may take part in the training of other models. For the second issue, using an ensemble of multiple models for evaluation will reduce the predictions' variance.

Parameter	Value
Optimizer	Adam
Learning Rate	5e-6
Batch Size	12
Max sentence in Batch	36
Max Sentence Length	96

Table 1: Hyper-parameters used in our experiments

Let M $= \{m^1, ..., m^n\}$ denote a list of predictions from multiple trained models, corresponding to a bag in the test set, and n is the number of models. Let $m^i = \{p_1^i, ..., p_l^i\}$ denote the scores predicted by model m^i, and l is the number of relations. Then, the final score predicted by the ensemble of n models for each relation is computed by taking the unweighted average of the scores predicted by all the models for that relation.

$$p_j^{\text{Ens}} = \frac{\sum_{i=1}^{n} p_j^i}{n} \tag{15}$$

where p_j^{Ens} denotes the score predicted by the ensemble for relation j for a bag in the test set.

4 Experiments

4.1 Dataset

We evaluate our model on the widely used NYT dataset (Riedel et al., 2010) which was generated by aligning freebase with the New York Times corpus. Articles from 2005 to 2006 were used for the training set, and articles from 2007 were used for the test set. There are 53 distinct relation types in the dataset, including the special NA relation which indicates the lack of semantic relation between entity pairs. The training set contains 570K sentences, and the test set contains 170K sentences.

4.2 Evaluation Metrics

Following previous works, we use the area under the curve (AUC), Precision@N (P@N) and precision-recall (PR) curves to evaluate our proposed method on the held-out test set of the NYT dataset.

4.3 Implementation Detail

We extend the OpenNRE framework (Han et al., 2019) to implement our model. We use $\text{BERT}_{\text{Large}}$ pre-trained model[1] released by Devlin et al. (2019) to initialize BERT. It has 24 encoder layers, 16 attention heads, and 1024 hidden state size. For hyper-parameter tuning, we use 20 percent of the training set as validation set and selected hyper-parameters that result in the best AUC value on the validation set. Table 1 shows the hyper-parameters used in our experiments. We train our model on the full training set for 2 epochs using these hyper-parameters. We use an ensemble of 5 models for prediction in the test stage, and all reported results are the average of 5 different runs. All our experiments were conducted using a single Tesla T4 GPU.

[1] https://github.com/google-research/bert

4.4 Baselines

We compare our model with the following baselines:

- Mintz (Mintz et al., 2009) is the original distant supervision model that uses hand-crafted features and a logistic regression classifier.
- PCNN+ATT (Lin et al., 2016) adopts the PCNN for sentence encoding and uses the selective attention mechanism to compute bag representations.
- PCNN+ATT_RA+BAG_ATT (Ye and Ling, 2019) adopts the PCNN as sentence encoder and uses inter-bag and intra-bag attention mechanisms for denoising.
- DISTRE (Alt et al., 2019) adopts the GPT as sentence encoder and performs bag-level relation extraction using selective attention.
- SeG (Li et al., 2020) adopts a self-attention enhanced PCNN along with entity-aware embeddings to represent sentences and uses a selective gate mechanism for denoising.

4.5 Evaluation Results

Table 4 shows AUC and P@N values of our proposed method and different baselines. The last two rows of the table show the results of our model when used with the two proposed sampling methods. We used an ensemble of 5 different trained models in our experiments. We observe that RISequal achieves the highest AUC value and significantly outperforms the previous state-of-the-art method by 0.104. Our method also outperforms all other baselines in P@N for all the values of N up to 2000 and in almost all recall levels. Figure

N_{max}	AUC	P@100	P@200	P@300	P@500	P@1000	P@2000
24	0.598	92.1	88.6	86.2	80.3	73.4	57.9
36	0.608	93.6	90.0	87.8	83.0	74.2	58.3
48	0.594	91.9	88.4	85.8	80.0	72.9	57.6

Table 2: P@N and AUC values for different values of N_{max} when using RIS-baseline

N_{max}	AUC	P@100	P@200	P@300	P@500	P@1000	P@2000
24	0.603	93.1	89.0	86.6	81.5	73.9	58.4
36	0.614	94.1	92.5	90.4	84.6	75.2	58.6
48	0.602	94.7	89.9	87.5	82.3	74.2	58.0

Table 3: P@N and AUC values for different values of N_{max} when using RIS-equal

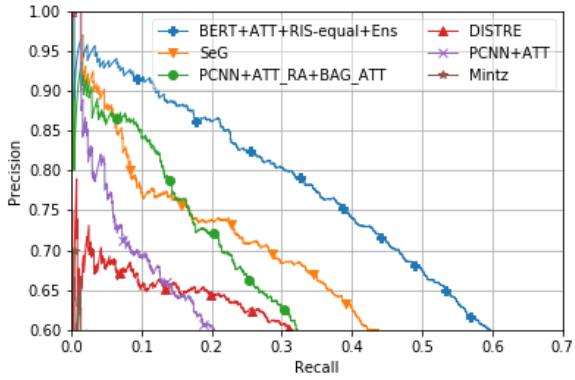


Figure 2: PR curves comparison of our proposed method and different baselines

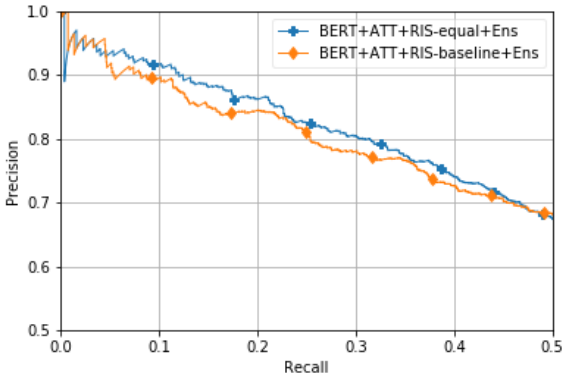


Figure 3: PR curves comparison of our model when used with our two proposed sampling methods

2 shows the PR curves of RIS-equal and baseline models.

We also observe that RIS-equal outperforms RIS-baseline in both AUC and P@N, which proves the effectiveness of sampling larger proportions from smaller bags. Figure 3 shows the PR curves of RIS-equal and RIS-baseline.

These results demonstrate that we can achieve state-of-the-art performance even when using a subset of bags in training in the distantly supervised setting.

Despite the similarities of our method with DISTRE, our evaluation results show a significant gap in the performance of the two methods. This shows the difference between the quality of language representations produced by GPT and $\text{BERT}_{\text{Large}}$. While $\text{BERT}_{\text{large}}$ has three times the number of parameters of GPT, our model requires much less memory for training compared with DISTRE, which shows the effectiveness of using RIS in training.

Our method also achieves much better results than SeG and PCNN+ATT_RA+BAG_ATT despite using a less effective denoising method. This indicates the importance of using better sentence representations compared with using better denoising methods.

4.6 Effectiveness of Ensemble Modeling

In this section, we conduct extensive experiments to show the effectiveness of using Ensembles for alleviating the randomness of RIS. We report P@N and AUC values for different numbers of trained models used for ensemble in Table 5. BERT+ATT+RIS-baseline and BERT+ATT+RIS-equal indicate the two methods without using ensemble modeling. As shown in the table, Increasing the number of trained models results in increased AUC and P@N for both RIS-baseline and RIS-equal. Overall, using more trained models for ensemble results in more training and evaluation time, and thus the choice of the number of models used for the ensemble is a trade-off between performance and speed.

Method	AUC	P@100	P@200	P@300	P@500	P@1000	P@2000
Mintz	0.106	51.8	50.0	44.8	39.6	33.6	23.4
PCNN+ATT	0.336	76.3	71.1	69.4	63.9	52.7	39.1
PCNN+ATT_RA+BAG_ATT	0.429	87.0	86.5	82.0	72.8	61.1	45.1
DISTRE	0.422	68.0	67.0	65.3	65.0	60.2	47.9
SeG	0.51	93.0	90.0	86.0	73.5	67.0	51.6
BERT+ATT+RIS-baseline+Ens	0.608	93.6	90.0	87.8	83.0	74.2	58.3
BERT+ATT+RIS-equal+Ens	**0.614**	**94.1**	**92.5**	**90.4**	**84.6**	**75.2**	**58.6**

Table 4: P@N and AUC values of different models

Method	AUC	P@100	P@200	P@300	P@500	P@1000	P@2000
BERT+ATT+RIS-baseline	0.544	89.1	85.1	82.7	78.0	69.5	54.6
+ Ensemble (n=2)	0.578	91.7	89.0	85.1	80.5	72.1	56.6
+ Ensemble (n=3)	0.596	93.2	90.5	86.8	81.8	73.9	57.5
+ Ensemble (n=4)	0.604	93.2	89.7	87.0	82.4	74.0	58.1
+ Ensemble (n=5)	0.608	93.6	90.0	87.8	83.0	74.2	58.3
BERT+ATT+RIS-equal	0.555	89.7	86.1	83.9	80.1	70.5	55.0
+ Ensemble (n=2)	0.581	91.5	89.3	87.8	81.9	72.1	56.5
+ Ensemble (n=3)	0.601	93.9	91.4	89.7	84.1	73.8	57.6
+ Ensemble (n=4)	0.606	95.0	91.9	89.0	83.4	74.2	58.2
+ Ensemble (n=5)	0.614	94.1	92.5	90.4	84.6	75.2	58.6

Table 5: P@N and AUC values for different number of trained models used for ensemble

4.7 Effect of Maximum Number of Sentences in Batch

We conducted experiments with different values of $N_{\max}$. We tested three different values of 24, 36, and 48. Using a single GPU with 16 GBs of memory, 48 was the maximum value of $N_{\max}$ we could set. The results of the experiments for both BERT+ATT+RIS-equal and BERT+ATT+RIS-baseline are demonstrated in Table 3 and Table 2 respectively. We expected our models to perform better for higher values of $N_{\max}$ as sampling a higher number of sentences from bags would result in a lower probability of losing valid sentences from bags. However, we achieved the best results when setting $N_{\max}$ to 36 for both sampling methods. This could be attributed to selective attention because when the number of sentences in a bag is smaller, the effect of the valid sentences in the weighted sum increases. Thus, when we set $N_{\max}$ to a small value, the resulting bags become smaller, and better bag representations could be computed. However, the probability of losing valid sentences increases, which negatively affects the quality of bag representations. The opposite holds for high values of $N_{\max}$.

5 Conclusion

In this paper, we proposed a new sample-based training method for distantly supervised relation extraction that reduces the hardware requirements of the multiple instance learning paradigm by randomly sampling sentences from bags in a batch. We then alleviated the issues raised by this randomness by using an ensemble of multiple trained models. The reduced hardware requirements allowed us to leverage a pre-trained BERT model for relation extraction in the distantly supervised setting. Experimental results on the widely used NYT dataset demonstrated that our method significantly outperforms current state-of-the-art methods in terms of both AUC and P@N values.

References

Christoph Alt, Marc Hübner, and Leonhard Hennig. 2019. Fine-tuning pre-trained transformer language models to distantly supervised relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1388–1398.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring hu-

man knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, pages 1247–1250. ACM.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

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.

Jun Feng, Minlie Huang, Li Zhao, Yang Yang, and Xiaoyan Zhu. 2018. Reinforcement learning for relation classification from noisy data. *AAAI*.

Xu Han, Tianyu Gao, Yuan Yao, Deming Ye, Zhiyuan Liu, and Maosong Sun. 2019. OpenNRE: An open and extensible toolkit for neural relation extraction. In *Proceedings of EMNLP-IJCNLP: System Demonstrations*, pages 169–174.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38, Uppsala, Sweden. Association for Computational Linguistics.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 541–550.

Yuyun Huang and Jinhua Du. 2019. Self-attention enhanced cnns and collaborative curriculum learning for distantly supervised relation extraction. 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 389–398.

Yang Li, Guodong Long, Tao Shen, Tianyi Zhou, Lina Yao, Huan Huo, and Jing Jiang. 2020. Self-attention enhanced selective gate with entity-aware embedding for distantly supervised relation extraction. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:8269–8276.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2124–2133.

Tianyu Liu, Kexiang Wang, Baobao Chang, and Zhifang Sui. 2017. A soft-label method for noise-tolerant distantly supervised relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1790–1795.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011.

Pengda Qin, Weiran Xu, and William Yang Wang. 2018. Robust distant supervision relation extraction via deep reinforcement learning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2137–2147.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 455–465.

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.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus. *Linguistic Data Consortium, Philadelphia*, 57.

Shanchan Wu and Yifan He. 2019. Enriching pre-trained language model with entity information for relation classification. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2361–2364.

Zhi-Xiu Ye and Zhen-Hua Ling. 2019. Distant supervision relation extraction with intra-bag and inter-bag attentions. 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 2810–2819.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1753–1762.

Static Fuzzy Bag-of-Words: a Lightweight and Fast Sentence Embedding Algorithm

Matteo Muffo[1,2], **Roberto Tedesco**[1], **Licia Sbattella**[1] and **Vincenzo Scotti**[1]
[1]DEIB, Politecnico di Milano
Via Golgi 42, 20133, Milano (MI), Italy
[2]Indigo.ai
Via Torino 61, 20123, Milano (MI), Italy
matteo@indigo.ai roberto.tedesco@polimi.it
licia.sbattella@polimi.it vincenzo.scotti@polimi.it

Abstract

The introduction of embedding techniques has pushed forward significantly the Natural Language Processing field. Many of the proposed solutions have been presented for word-level encoding; anyhow, in the last years, new mechanisms to treat information at a higher level of aggregation, like at sentence- and document-level, have emerged. With this work, we address specifically the sentence embeddings problem, presenting the *Static Fuzzy Bag-of-Word* model. Our model is a refinement of the Fuzzy Bag-of-Words approach, providing sentence embeddings with a fixed dimension. SFBoW provides competitive performances in Semantic Textual Similarity benchmarks while requiring low computational resources.

1 Introduction

Natural Language Processing (NLP) has gained much traction in the last years, mainly thanks to learnt semantic representations. Those representations (usually) called embeddings are, in the textual context, real-valued vectors representing the semantic meaning of words, sentences or even documents in a Euclidean space. These vectors are features generated by models trained using a *self-supervised* approach on a vast corpus of unlabelled text. Leveraging features obtained through a self-supervised approach instead of "hand-selected" features is of crucial importance for NLP (Bengio et al., 2013).

Textual learnt representations immediately turned out to be significant in many NLP tasks, from more simple ones, like Part-Of-Speech (POS) tagging, Named Entity Recognition (NER), and language modelling (Collobert et al., 2011), to more complex problems such as Machine Translation (Sutskever et al., 2014), and even Conversational Systems (Sordoni et al., 2015). These representations significantly moved forward state of the art.

Results in the tasks mentioned above have been boosted mainly thanks to learnt word-level semantic vectors, i.e. *word embeddings*. Nevertheless, for many problems like web search, question answering and image captioning, having access to higher-level representations is crucial: this is where *sentence embeddings* find their usefulness (Yih et al., 2015).

Recent outcomes show that *contextual* representations, learnt through *Transformer Network* (Vaswani et al., 2017) Language Models (LMs) (Devlin et al., 2019; Radford et al., 2018), provide better performances in all those tasks and are slowly substituting "static" embeddings. Even though the results of these models are remarkable, their usability is strongly restricted because of their high demand in terms of computational resources; hence we decided to focus on more lightweight solutions.

With this work, we introduce the Static Fuzzy Bag-of-Words (SFBoW) model, an improvement of DynaMax Fuzzy Bag-of-Words model (Zhelezniak et al., 2019). Differently from its predecessor, the size of universe matrix (and thus the dimension of the generated embeddings) is fixed, hence the name *static*. SFBoW is characterised by low analysis time and interesting Semantic Textual Similarity (STS) results without demanding high computational power, making it an interesting solution for embedded systems and, in general, for applications where resources are limited and power consumption is a concern.

The rest of this document is organised as follows. In Section 2 we present the related works in the field of learnt semantic representations. In Section 3 we present the SFBoW model. In Section 4 we present the experiments to assess the quality of our model. In Section 5 we present the results of experiments. In Section 6 we sum up the entire work and propose possible future directions.

2 Related work

Our work revolves around the concept of *vector semantics*: the idea that the meaning of a word or a sentence can be modelled as a vector (Osgood et al., 1958).

The first steps on this subject were made in Information Retrieval (IR) context with the vector space model (Salton, 1971), where documents and queries were represented as high dimensional (vocabulary size) sparse embedding vectors. In this model, each dimension is used to represent a word, so that given a vocabulary $\mathcal{V}$:

- A word $w_i \in \mathcal{V}$, with $i \in [1, |\mathcal{V}|] \subseteq \mathbb{N}$, is expressed as a so called "one hot" binary vector $\mathbf{v}_{w_i} \in \mathbb{1}^{|\mathcal{V}|}$, where, calling $v_{w_i,j}$ the j-th element of the word vector, it holds that $v_{w_i,j} = 1 \iff j = i$.
- A sentence S is expressed as vector $\boldsymbol{\mu}_S \in \mathbb{N}^{|\mathcal{V}|}$, where $\mu_{S,i}$, the i-th element of vector $\boldsymbol{\mu}_S$, namely $c_{S,i}$, represents the number of times word w_i appears in sentence S.

The resulting sentence representation, used also for text documents, is called *Bag-of-Words* (BoW), and can be summarised as

$$\boldsymbol{\mu}_S = \sum_{i=1}^{|\mathcal{V}|} c_{S,i} \cdot \mathbf{v}_{w_i} \tag{1}$$

These representation models needed to be replaced because of the sparsity, which made them resource consuming, and the induced orthogonality among vectors with similar meanings.

2.1 Word and sentence embeddings

Word embeddings refer to the dense semantic vector representation of words. Approaches for word embeddings can be divided into: *prediction-based* and *count-based* (Baroni et al., 2014).

The former group identifies the embeddings obtained through the training of models for next/missing word prediction given a context. It encompasses models like *Word2Vec* (Mikolov et al., 2013a,b) and *fastText* (Bojanowski et al., 2017). The latter group refers to the embeddings obtained leveraging words co-occurrence counts in a corpus. One of the most recent solutions of this group is *GloVe* (Pennington et al., 2014).

All the models mentioned above belong to the class of *shallow* models, where the embedding of a word w_i can be extracted through lookup over the rows of the embedding matrix $\mathbf{W} \in \mathbb{R}^{|\mathcal{V}| \times d}$, with d being the desired dimensionality of the embedding space. Given the word (column) vector $\mathbf{v}_{w_i}$, the corresponding word embedding $\mathbf{u}_{w_i} \in \mathbb{R}^d$ can be computed as (see Section 2.2)

$$\mathbf{u}_{w_i} = \mathbf{W}^\top \cdot \mathbf{v}_{w_i} \tag{2}$$

More recently, the introduction of Transformer-based LMs, like *BERT* (Devlin et al., 2019) or *GPT* (Radford et al., 2018), has spread the concept of *contextual embeddings*; such embeddings proved to be particularly helpful for a wide variety of NLP problems, as shown by the leader-boards of NLP benchmarks (Wang et al., 2019; Rajpurkar et al., 2016).

The inherent hierarchical structure of the human language makes it hard to understand a text from single words; thus, the birth of higher-level semantic representations for sentences, which are the sentence embeddings, was just a natural consequence. As for the Word embeddings, also sentence embeddings are organised into two groups: *parametrised* and *non-parametrised*, depending on whether the model requires parameter training or not.

Clear examples of parametric model are the *Skip-Thoughts vectors* (Kiros et al., 2015) and *Sent2Vec* (Pagliardini et al., 2018), which generalises Word2Vec. Non-parametric models, instead, show that simply aggregating the information from pre-trained word embeddings, for example through averaging, as in *SIF weighting* (Arora et al., 2017), is sufficient to represent higher-level entities like sentences and paragraphs.

Transformer LMs are also usable at sentence level. An example is the parametric model *Sentence-BERT* (Reimers and Gurevych, 2019), obtained by fine-tuning on Natural Language Inference corpora.

All these models rely on the assumption that cosine similarity is the correct metric to compute "meaning distance" between sentences. This is why parametric models are explicitly trained to minimise this measure for similar sentences and maximise it for dissimilar sentences. However, this may not be the only and best measure. The *DynaMax* model (Zhelezniak et al., 2019) proposed to follow a fuzzy set representation of sentences and to rely on fuzzy Jaccard similarity instead of the cosine one. As a result, the DynaMax model outperformed many non-parametric models and performed comparably to parametric ones under cosine similarity

measurements, even if competitors were trained directly to optimise that metric, while the DynaMax approach was utterly unrelated to that objective.

The use of fuzzy sets to represent documents is not new, it was already proposed by Zhao and Mao (2018). With respect to DynaMax, previous results were inferior because of their approach to compute fuzzy membership.

2.2 Fuzzy Bag-of-Words and DynaMax for sentence embeddings

The *Fuzzy Bag-of-Words* (FBoW) model for text representation (Zhao and Mao, 2018) – and its generalised and improved variant DynaMax (Zhelezniak et al., 2019), which introduced a better similarity metric – represent the starting point of our work, which is described in Section 3.

The BoW approach, described at the beginning of Section 2, can be seen as a multi-set representation of text. It enables to measure similarity between two sentences with set similarity measures, like Jaccard, Otsuka and Dice indexes. These indexes share all a common pattern to measure the similarity σ between two sets A and B (Zhelezniak et al., 2019):

$$\sigma(A, B) = n_{shared}(A, B) / n_{total}(A, B) \quad (3)$$

where $n_{shared}(A, B)$ denotes the count of shared elements and $n_{total}(A, B)$ is the count of total elements. In particular, the Jaccard index is defined as

$$\sigma_{Jaccard}(A, B) = |A \cap B| / |A \cup B| \quad (4)$$

However, the simple set similarity is a rigid approach as it allows for some degree of similarity when the very same words appear in both sentences, but fails in the presence of synonyms. This is where *Fuzzy Sets theory* comes handy: in fact, fuzzy sets enable to interpret each word in $\mathcal{V}$ as a singleton and measure the degree of membership of any word to this singleton as the similarity between the two considered words (Zhao and Mao, 2018).

The FBoW model prescribes to work in this way (Zhao and Mao, 2018):

- Each word w_i is interpreted as a singleton $\{w_i\}$; thus, the membership degree of any word w_j in the vocabulary (with $j \in [1, |\mathcal{V}|] \subseteq \mathbb{N}$) with respect to this set is computed as the similarity σ between w_i and w_j. These similarities can be used to fill a $|\mathcal{V}|$-sized vector $\hat{\mathbf{v}}_{w_i}$ used to provide the fuzzy representation of w_i (the j-th element $\hat{\mathbf{v}}_{w_i,j}$ being $\sigma(w_i, w_j)$).
- A sentence S is simply defined through the fuzzy union operator, which is determined by the $\max$ operator over the membership degrees. In this case the S is represented by a vector of $|\mathcal{V}|$ elements.

The generalised FBoW approach (Zhelezniak et al., 2019), prescribes to computes the *fuzzy embedding* of a word singleton as

$$\hat{\mathbf{v}}_{w_i} = \mathbf{U} \cdot \mathbf{u}_{w_i} = \mathbf{U} \cdot \mathbf{W}^\top \cdot \mathbf{v}_{w_i} \quad (5)$$

to reduce the dimension of the output vector for S. Where, $\mathbf{W} \in \mathbb{R}^{|\mathcal{V}| \times d}$ is a word embedding matrix (defined as in Section 2.1), $\mathbf{u}_{w_i}$ is defined in Equation (2) and $\mathbf{U} \in \mathbb{R}^{u \times d}$ (with u being the desired dimension of the fuzzy embeddings) is the *universe matrix*, derived from the *universe set* U, which is defined as "the set of all possible terms that occur in a certain domain". The generalised FBoW produces vectors of u elements, where $u = |U|$.

Given the fuzzy embeddings of the words in a sentence S, the generalised FBoW representation of S is a vector $\hat{\boldsymbol{\mu}}_S$ whose j-th element $\hat{\mu}_{S,j}$ ($j \in [1, u] \subseteq \mathbb{N}$) can be computed as:

$$\hat{\mu}_{S,j} = \max_{w_i \in S} c_{S,i} \cdot \hat{v}_{w_i,j} \quad (6)$$

where $c_{S,i}$ and $\hat{v}_{w_i,j}$ are, respectively, the number of occurrences of word w_i in sentence S and the j-th element of the $\hat{\mathbf{v}}_{w_i}$ vector.

The universe set can be defined in different ways, same applies for the universe matrix (Zhelezniak et al., 2019). Among the possible solutions, the DynaMax algorithm for fuzzy sentence embeddings builds the universe matrix from the word embedding matrix, stacking solely the embedding vectors of the words appearing in the sentences to be compared.

Notice that in this way the resulting universe matrix is not unique, as a consequence neither are the embeddings. This condition can be noticed from the description of the algorithm and from the definition of the universe matrix: when comparing two sentences S_a and S_b, the universe set U used in their comparison is $U \equiv S_a \cup S_b$, so the resulting sentence embeddings have size $u = |U| = |S_a \cup S_b|$. In fact, the universe matrix is given by

$$\mathbf{U} = \left[\mathbf{u}_{w_i} \forall w_i \in U\right]^\top \quad (7)$$

This characteristic is unfortunate as, for example, in IR it requires a complete re-encoding of the entire document achieve for each query.

The real improvement of DynaMax is in the introduction of the fuzzy Jaccard index to compute the semantic similarity between two sentences S_a and S_b, rather than the generalisation of the FBoW, which replaced the original use of the cosine similarity (Zhao and Mao, 2018); see Equation (8):

$$\hat{\sigma}_{Jaccard}\left(\hat{\boldsymbol{\mu}}_{S_a}, \hat{\boldsymbol{\mu}}_{S_b}\right) = \\ = \frac{\sum_{i=1}^{u} \min\left(\hat{\mu}_{S_a,i}, \hat{\mu}_{S_b,i}\right)}{\sum_{i=1}^{u} \max\left(\hat{\mu}_{S_a,i}, \hat{\mu}_{S_b,i}\right)} \tag{8}$$

3 Static Fuzzy Bag-of-Words model

Starting from the DynaMax, which evolved from the FBoW model, we developed our follow up aimed at providing a unique matrix $\mathbf{U}$ and thus embeddings with a fixed dimension. In Figure 1 is represented the visualisation of our approach.

3.1 Word embeddings

Word embeddings play a central role in our algorithm as they also provide the start point of the construction of the universe matrix. For this work, we leveraged pre-trained shallow models (more details in Section 4.1) for two main reasons:

- The model is encoded in a matrix where each row corresponds to a word.
- We want to provide a sentence embedding approach that does not require training, easing its accessibility.

The vocabulary of these models, composed starting from all the tokens in the training corpora, is usually more extensive than the English vocabulary, as it contains named entities, incorrectly spelt words, non-existing words, URLs, email addresses, and similar. To reduce the computational effort needed to construct and use the universe matrix, we have considered some subsets of the employed word embedding model's vocabulary. Depending on the experiment, we work with either the 100 000 most frequently used terms, the 50 000 most frequently used terms (terms frequencies are given by the corpora used to train the word embedding model) or the subset composed of all the spell-checked terms present in a reference English dictionary (obtained through the Aspell English spell-checker[1]).

[1] http://aspell.net

In the following sections, the $\check{\mathbf{W}}$ symbol refers to these as *reduced* word embedding matrices/models.

3.2 Universe matrix

During the experiments, we tried three main approaches to build the universe matrix $\mathbf{U}$: the first two – proposed, but not explored, by the original authors of DynaMax (Zhelezniak et al., 2019) – consist, respectively, in the usage of a clustered embedding matrix and an identity matrix with the rank equal to the size of the word embeddings. Instead, the last approach consists of applying a multivariate analysis techniques to the word embedding matrix to build the universe one. In the following formulae, we refer to d as the dimensionality of the word embedding vectors, while the SFBoW embedding of the singleton of word w_i is represented as $\check{\mathbf{v}}_{w_i}$.

Clustering The idea is to group the embedding vectors into clusters and use their centroids; in this way, the fuzzy membership will be computed over the clusters – which are expected to host semantically similar words – instead of all the word singletons. The universe set is thus built out of abstract entities only, which are the centroids. Considering k centroids, the universe matrix $\mathbf{U} = \mathbf{K}^\top \in \mathbb{R}^{k \times d}$, and thus SFBoW k-dimensional embedding $\check{\mathbf{v}}_{w_i}$ of the singleton of word w_i is

$$\check{\mathbf{v}}_{w_i} = \mathbf{K}^\top \cdot \mathbf{u}_{w_i} = \left[\mathbf{k}_1, \ldots, \mathbf{k}_k\right]^\top \cdot \mathbf{u}_{w_i} = \\ = \mathbf{K}^\top \cdot \mathbf{W}^\top \cdot \mathbf{v}_{w_i} \tag{9}$$

where $\mathbf{k}_j$, the j-th (with $j \in [1, k] \subseteq \mathbb{N}$) column of $\mathbf{K}$, corresponds to the centroid of the j-th cluster. This approach generates k-dimensional word and sentence embeddings.

Identity Alternatively, instead of looking for a group of semantically similar words that may form a significant group, useful for semantic similarity, we consider the possibility of re-using the word embedding dimensions (features) to represent the semantic content of a sentence. So, we just use the identity matrix as the universe: $\mathbf{U} = \mathbf{I}$, with $|\mathbf{I}| = d \times d$, so that $\check{\mathbf{v}}_{w_i} \in \mathbb{R}^d$ is

$$\check{\mathbf{v}}_{w_i} = \mathbf{I} \cdot \mathbf{u}_{w_i} = \mathbf{I} \cdot \mathbf{W}^\top \cdot \mathbf{v}_{w_i} \tag{10}$$

This approach generates d-dimensional word embeddings and sentence embeddings.

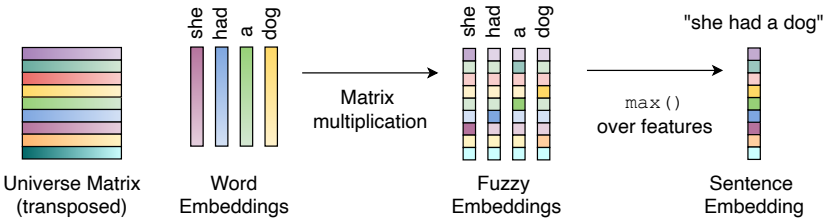


Figure 1: Visualisation of the Sentence Embedding computation process using SFBoW.

Multivariate analysis The same idea moves our multivariate analysis proposal. Judging by previous results, word embeddings aggregated correctly might be sufficient to provide a semantically valid representation of a sentence. What can bring better results might be as simple as roto-translate the reference system of the embedding representation. In this sense, we propose to use to compute the fuzzy membership, and hence the fuzzy Jaccard similarity index, over these dimensions resulting from roto-translation, expecting that this "new perspective" will expose better the semantic content. So, defining $\mathbf{U} = \mathbf{M}$, where $\mathbf{M}$ is the transformation matrix, with $|\mathbf{M}| = d \times d$, we have that $\check{\mathbf{v}}_{w_i} \in \mathbb{R}^d$ is

$$\check{\mathbf{v}}_{w_i} = \mathbf{M} \cdot \mathbf{u}_{w_i} = \mathbf{M} \cdot \mathbf{W}^\top \cdot \mathbf{v}_{w_i} \qquad (11)$$

Thus yielding d-dimensional word and sentence embeddings.

Clustering and multivariate analysis can be applied to the whole embedding vocabulary or the subsets of the vocabulary introduced in Section 3.1. Apart from reducing the computational time, we did so to see if these subsets are sufficient to provide a helpful representation.

4 Experiments

In order to find the best solution in terms of word embedding matrix and universe matrix, we explored various possibilities. Then, to measure the goodness of our sentence embeddings, we leveraged a series of STS tasks and compared the results with the preceding models.

4.1 Word embeddings

For what concerns the word embeddings, we have decided to work with a selection of four models:

- Word2Vec, with 300-dimensional embeddings;
- GloVe, with 300-dimensional embeddings;
- fastText, with 300-dimensional embeddings;
- Sent2Vec, with 700-dimensional embeddings.

As shown by the word embedding models list, we are also employing a Sent2Vec sentence embedding model. The embedding matrix of this model can be used for word embeddings too. During the experiments, we focused on the universe matrix construction. For this reason, we relied on pretrained models for word embeddings, available on the web.

4.2 Universe matrices

The universe matrices we considered are divided into three buckets, as described in Section 3.2.

Clustering Universe matrices built using clustering leverage four different algorithms: k-Means, Spherical k-Means, DBSCAN and HDBSCAN.

We selected k-Means and Spherical k-Means because they usually lead to good clustering results; the latter was specifically designed for textual purposes, with low demand in time and computation resources. For all algorithms, we considered the same values for k (the number of centroids), which were 100, 1000, 10 000 and 25 000. For all the values of k, we performed clustering on different subsets of the vocabulary: k-Means was applied on the whole English vocabulary as well as to the top 100 000 frequently used words subset, while Spherical k-Means was applied to the subset of the first 50 000 frequently used words (in order to reduce computational time).

We also explored density-based algorithms (DBSCAN and HDBSCAN), which do not require defining in advance the number of clusters, using euclidean and cosine distance between the word embedding. For DBSCAN with euclidean distance, we varied the radius of the neighbourhood ε between 3 and 8 and worked over the same

two subsets considered for k-Means, while for cosine distance ε was between 0.1 and 0.55 and it was applied over the subset of the first 50 000 frequently used words (for computational reasons, as we did for Spherical k-Means). Concerning HDBSCAN, we varied the smallest size grouping of clusters in the set $\{2, 4, 30, 50, 100\}$ and the minimum neighbourhood size of core samples in the set $\{1, 2, 5, 10, 50\}$. We considered this latter density-based algorithm since basic DBSCAN happens to fail with high-dimensional data.

Identity This approach consists of using the identity matrix as the universe, in this way, the singletons we use to compute the fuzzy membership are the dimensions of the word embeddings, which corresponds to the learnt features. This is the most lightweight method as it just requires to compute the word embeddings of a sentence and then the fuzzy membership over the exact d dimensions.

Multivariate analysis We adopted the Principal Component Analysis (PCA) to get a rotation matrix to serve as a universe matrix to the SFBoW. In fact, through PCA, the d-dimensional word embedding vectors are decomposed along with the d orthogonal directions of their variance. These components are then reordered to decrease explained variance and represent our fuzzy semantic sets.

The principal component of the reduced word embedding matrix $\check{\mathbf{W}}$ are described by the matrix $\mathbf{T} = \mathbf{P}^\top \cdot \check{\mathbf{W}}$, where $\mathbf{P}$ is a $d \times d$ matrix whose columns are the eigenvectors of the matrix $\check{\mathbf{W}}^\top \cdot \check{\mathbf{W}}$. With our approach, the matrix $\mathbf{P}^\top$, sometimes called the *whitening* or *sphering transformation matrix*, serves as universe matrix $\mathbf{U}$. In this way, the SFBoW embedding of a word singleton becomes

$$\check{\mathbf{v}}_{w_i} = \mathbf{P}^\top \cdot \mathbf{u}_{w_i} = \mathbf{P}^\top \cdot \check{\mathbf{W}}^\top \cdot \mathbf{v}_{w_i} \qquad (12)$$

As for the clustering approach, we experimented with both the whole vocabulary and the most 100 000 used words.

4.3 Data

We evaluated our SFBoW through a series of reference benchmarks; we selected the STS benchmark series, one of the tasks of the International Workshop on Semantic Evaluation (SemEval)[2].

SemEval is a series of evaluations on computational semantics; among these, the *Semantic Textual Similarity* STS benchmark[3] (Cer et al., 2017) has become a reference for scoring of sentence embedding algorithms. All the previous models we are considering for comparison have been benched against STS; this is because the benchmark highlights a model capability to provide a meaningful semantic representation by scoring the correlation between model's and human's judgements. For this reason, and also to allow comparisons, we decided to evaluate SFBoW on STS.

We worked only on the English language, using the editions of STS from 2012 to 2016 (Agirre et al., 2012, 2013, 2014, 2015, 2016). Each year, a collection of corpora coming from different sources has been created and manually labelled, in Table 1 is possible to have a reference in terms of support for each edition. Thanks to the high number of samples, we are confident about the robustness of our results.

Table 1: Support of the corpora of the STS benchmark series.

Edition	No. sentence pairs
STS 2012	5250
STS 2013	2250
STS 2014	3750
STS 2015	3000
STS 2016	1186
Total	15 436

To preprocess the input text strings, we lowecased each character and tokenised in correspondence of spaces and punctuation symbols. Then, from the resulting sequence, we retained only the tokens for which a corresponding embedding was found in the vocabulary known by the model. Finally, we calculated the SFBoW sentence embedding from the word embeddings of such tokens.

The samples constituting the corpora are pair of sentences with a human-given similarity score (the *gold labels*). The provided score is a real-valued index obtained averaging those of multiple crowd-sourced workers and is scaled in a $[0, 1] \in \mathbb{R}$ interval. The final goal of our work is to provide a model able to provide a score as close as possible

[2] https://aclweb.org/aclwiki/SemEval_Portal

[3] https://ixa2.si.ehu.eus/stswiki/index.php/Main_Page

to that of humans.

4.4 Evaluation approach

To assess the quality of our model, we used it to compute the similarity score between the sentence pairs provided by the five tasks, and we compared the output with the target labels. The results are computed as the correlation between the similarity score produced by SFBoW and the human one, using Spearman's ρ measure (Reimers et al., 2016). SFBoW employs fuzzy Jaccard similarity index (Zhelezniak et al., 2019) to compute word similarity.

To have terms of comparison, we establish a baseline through the most straightforward models possible, the average word embedding in a sentence, leveraging three different word embedding models: Word2Vec, GloVe and fastText. We also provide results from more complex models: SIF weighting (applied to GloVe), Sent2Vec, DynaMax (built using Word2Vec, GloVe and fastText) and Sentence-BERT.

All the embedding models except DynaMax and the baselines are scored using cosine similarity; DynaMax scores are obtained using fuzzy Jaccard similarity index.

5 Results

The results of the Spearman's ρ correlation in the STS benchmark of our SFBoW are reported in the last three rows of Table 2. The reported values belong to the FSBoW configurations that achieved the best score, among the variants we considered for the experiments, in at least one task.

FastText is the best among the four-word embeddings models, confirming the results of DynaMax. The best scores in terms of universe matrix are achieved either with Identity matrix or with PCA rotation matrix, highlighting how the features yield by word embeddings provide a better semantic content representation of sentences.

Clusterings results turned out to be very poor, independently of the starting embeddings. For this reason, we avoid discussing them.

As premised, we compare our results with three baseline models and other sentence embedding approaches, all reported in Table 2. The first group of scores is from the baselines, the second one is from other sentence embedding models and, finally, the last group is from our SFBoW model. Additionally, the best values in each column are highlighted in bold, while the second ones are underlined.

The key features about our model, which can be derived from the results, are the following:

- low number of parameters;
- faster inference time
- no training phase;
- results (in terms of ρ) comparable to similar models;
- fixed-size and easily re-usable embeddings.

About the number of parameters, we can notice that even if Sentence-BERT outperforms all the other models in every task, it relies on a much deeper feature extraction model and was trained on a much bigger corpus. Moreover, this model requires a considerably higher computational effort without an equally consistent difference in performances. BERT alone requires more than 100 million parameters just for its base version (and above 300 million for the large one), hence taking a lot of (memory) space, not to mention the amount of time necessary for the self-supervised training and the fine-tuning. On the other hand, non-parametric models (like SIF, DynaMax or SFBoW) or shallow parametric ones (Sent2Vec) require fewer parameters: just those for the embedding matrix $|\mathcal{V}| \times d$.

A similar discourse applies to inference speed. Even though Sentence-BERT achieves the best results on all tasks, SFBoW turns out to be four times faster at inferring the similarity, as can be noticed by the reported analysis times.

Being a non-parametric model, SFBoW does not require a training phase. It may require clustering the embeddings to build the universe matrix, but our experiments showed that clustering does not yield good results. Because of its simplicity, SFBoW can generally be easily deployed, requiring only the word embedding model to compute the sentence representation. Notice also that the SFBoW algorithm is agnostic to the word embedding model.

Regarding the results we obtained, compared to other models, SFBoW provided interesting figures: either considering the majority of tasks with higher Spearman's ρ rank or higher average score, it outperforms all the baselines, as well as SIF weighting and Sent2Vec. Finally, we see as our model performs closely to its predecessor, especially considering the weighted average of the results of the

Table 2: Comparison of results over the STS benchmark. SFBoW models are in the last block. Weighted averages are expressed as: *avg.*±*std.* Bold and underlined values represent, respectively, first and second best result of column. Inference time refers to to the time, in seconds, to carry out an evaluation on the entire STS corpus.

Model	**Results** (Spearman's ρ)						**Analysis time** (s)
	STS					**Total**	
	2012	**2013**	**2014**	**2015**	**2016**		
Word2Vec [a]	55.46	58.23	64.05	67.97	66.28	61.21±5.04	–
GloVe [a]	53.28	50.76	55.63	59.22	57.88	54.99±2.80	–
fastText [a]	58.82	58.83	63.42	69.05	68.24	62.65±4.20	–
SIF weighting [b]	56.04	<u>62.74</u>	64.29	69.89	70.71	62.84±5.54	–
Sent2Vec	56.26	57.02	65.82	74.46	69.01	63.21±7.13	–
DynaMax [c]	55.95	60.17	65.32	73.93	71.46	63.53±6.92	–
DynaMax [b]	57.62	55.18	63.56	70.40	71.36	62.25±5.85	–
DynaMax [d]	61.32	61.71	66.87	<u>76.51</u>	<u>74.71</u>	<u>66.71</u>±6.10	–
Sentence-BERT	**72.27**	**78.46**	**74.90**	**80.99**	**76.25**	**75.81**±3.27	218.3
SFBoW [d,e,f]	61.31	51.21	<u>67.47</u>	72.90	73.88	64.55±7.20	56.5
SFBoW [d,g,h]	<u>61.42</u>	51.36	66.44	72.74	73.72	64.32±7.00	56.8
SFBoW [d,g,i]	60.03	51.96	66.36	72.39	73.25	63.81±6.93	56.6

[a] Used as baseline. [b] Built upon a GloVe model for word embeddings. [c] Built upon a Word2Vec model for word embeddings. [d] Built upon a fastText model for word embeddings.
[e] Best average score. [f] Universe matrix is the identity matrix.
[g] Universe matrix is the PCA projection matrix. [h] Universe matrix is built from the English vocabulary.
[i] Universe matrix is built from the top 100 000 most frequent words.

single tasks. SFBoW bests out DynaMax in STS 2014 and gets almost the same results in STS 2012 (the difference is 0.01), which are the first two corpora in terms of samples; however, the difference in STS 2013 goes in favour of DynaMax.

About the comparison against DynaMax, it is worth underlining a few additional points. First of all, in both cases, fuzzy Jaccard similarity correlates better with human judgement as a measure of sentence similarity. Secondly, both models manage to achieve better results when using fastText word embedding, possibly underling that they lend better than other models at sentence level combination; the baseline performances also show this.

Finally, we remind that SFBoW generates embeddings with a fixed size, resulting in much easier applicability with respect to DynaMax.

6 Conclusion

With this work, we have proposed SFBoW, a refinement of the FBoW and DynaMax models for sentence embedding. Even if SFBoW does not achieve state-of-the-art performances on the considered STS benchmark, our solution performs comparably to its predecessor while enabling the possibility of re-usable embeddings as their dimension is fixed. Moreover, as can be seen from the results, it outperforms in most tasks all the other compared models except Sentence-BERT, without the need for specific training or fine-tuning on sentence similarity corpora and still being as lightweight and fast as possible. As a result, SFBoW seems a reasonable solution in low resources or constrained computational scenarios. In the future, we plan to investigate other clustering techniques and other methodologies for computing the universe matrix.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel M. Cer, Mona T. Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2015, Denver, Colorado, USA, June 4-5, 2015*, pages 252–263. The Association for Computer Linguistics.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel M. Cer, Mona T. Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. Semeval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014*, pages 81–91. The Association for Computer Linguistics.

Eneko Agirre, Carmen Banea, Daniel M. Cer, Mona T. Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*, pages 497–511. The Association for Computer Linguistics.

Eneko Agirre, Daniel M. Cer, Mona T. Diab, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of the 6th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2012, Montréal, Canada, June 7-8, 2012*, pages 385–393. The Association for Computer Linguistics.

Eneko Agirre, Daniel M. Cer, Mona T. Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *sem 2013 shared task: Semantic textual similarity. In *Proceedings of the Second Joint Conference on Lexical and Computational Semantics, *SEM 2013, June 13-14, 2013, Atlanta, Georgia, USA*, pages 32–43. Association for Computational Linguistics.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 238–247, Baltimore, Maryland. Association for Computational Linguistics.

Yoshua Bengio, Aaron C. Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomás Mikolov. 2017. Enriching word vectors with subword information. *Trans. Assoc. Comput. Linguistics*, 5:135–146.

Daniel M. Cer, Mona T. Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation, SemEval@ACL 2017, Vancouver, Canada, August 3-4, 2017*, pages 1–14. Association for Computational Linguistics.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537.

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.

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 3294–3302.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 3111–3119.

Charles Egerton Osgood, George J Suci, and Percy H. Tannenbaum. 1958. The measurement of meaning. *American Journal of Sociology*, 63(5):550–551.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2018. Unsupervised learning of sentence embeddings using compositional n-gram features. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 528–540. 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 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, pages 1532–1543. ACL.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *OpenAI Blog*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 2383–2392. The Association for Computational Linguistics.

Nils Reimers, Philip Beyer, and Iryna Gurevych. 2016. Task-oriented intrinsic evaluation of semantic textual similarity. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 87–96, Osaka, Japan. The COLING 2016 Organizing Committee.

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 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics.

Gerard Salton. 1971. *The SMART retrieval system: experiments in automatic document processing*. Prentice-Hall series in automatic computation. Prentice-Hall.

Alessandro Sordoni, Michel Galley, Michael Auli, Chris Brockett, Yangfeng Ji, Margaret Mitchell, Jian-Yun Nie, Jianfeng Gao, and Bill Dolan. 2015. A neural network approach to context-sensitive generation of conversational responses. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 196–205, Denver, Colorado. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

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.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3261–3275.

Wen-tau Yih, Xiaodong He, and Jianfeng Gao. 2015. Deep learning and continuous representations for natural language processing. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 6–8. The Association for Computational Linguistics.

Rui Zhao and Kezhi Mao. 2018. Fuzzy bag-of-words model for document representation. *IEEE Trans. Fuzzy Syst.*, 26(2):794–804.

Vitalii Zhelezniak, Aleksandar Savkov, April Shen, Francesco Moramarco, Jack Flann, and Nils Y. Hammerla. 2019. Don't settle for average, go for the max: Fuzzy sets and max-pooled word vectors. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

ITAcotron 2: Transfering English Speech Synthesis Architectures and Speech Features to Italian

Anna Favaro[1], **Licia Sbattella**[2], **Roberto Tedesco**[2] and **Vincenzo Scotti**[2]
[1]CIMeC, Università degli Studi di Trento
Corso Bettini 31, 38068, Rovereto (TN), Italy
[2]DEIB, Politecnico di Milano
Via Golgi 42, 20133, Milano (MI), Italy
anna.favaro@studenti.unitn.it licia.sbattella@polimi.it
roberto.tedesco@polimi.it vincenzo.scotti@polimi.it

Abstract

End-to-end deep learning models have pushed forward significantly many tasks of Natural Language Processing (NLP). However, most of these models are trained for languages providing many resources (such as English), and their behaviour is hardly studied in other languages due to resource shortage. To cope with these situations, it is common practice to employ *transfer learning*. With this work, we wanted to explore the cross-language transferability of a Text-to-Speech (TTS) architecture and the re-usability of the surrounding components that complete a speech synthesis pipeline. To do so, we fine-tuned an English version of the Tacotron 2 TTS, with speaker conditioning, to Italian (hence *ITAcotron 2*). The human evaluation –carried on 70 subjects– showed that the language adaptation was indeed successful.

1 Introduction

The development of Text-to-Speech (TTS) synthesis systems is one of the oldest problems in the Natural Language Processing (NLP) area and has a wide variety of applications (Jurafsky and Martin, 2009). Such systems are designed to output the waveform of a voice uttering the input text string. In the last years, the introduction of deep learning-based approaches, and in particular the end-to-end ones (Shen et al., 2018; Ping et al., 2018; Ren et al., 2019; Hsu et al., 2019), led to significant improvements.

Most of the evaluations carried out on these models are performed on languages with many available resources, like English. Thereby, it is hard to tell whether and how good these models and architectures are general across languages. With this work, we proposed to study how these models behave with less-resourced languages.

To evaluate the transferability of a TTS architecture to a different language, the effectiveness of training a new model starting from –and fine-tuning– another one, and to verify the effect on training convergence, we experimented with English and Italian languages. In particular, we started from the English TTS Tacotron 2 and fine-tuned its training on a collection of Italian corpora. Then, we extended the resulting model, with speaker conditioning; the result was an Italian TTS we named ITAcotron 2.

ITAcotron 2 was evaluated, through human assessment, on intelligibility and naturalness of the synthesised audio clips, as well as on speaker similarity between target and different voices. In the end, we obtained reasonably good results, in line with those of the original model.

We divide the rest of this paper into the following sections. In Section 2 we explain the problem and the available solutions. In Section 3 we present the corpora employed to train and test out the model. In Section 4 we explain the structure of the synthesis pipeline we are proposing and how we adapted it to Italian from English. In Section 5 we describe the experimental approach we followed to assess the model quality. In Section 6 we comment on the results of our model. In Section 7 we sum up our work and suggest possible future extensions.

2 Background

Modern, deep learning-based TTS pipelines are composed of two main blocks: a *spectrogram predictor* and a *vocoder* (Jurafsky and Martin, 2009). These components take care of, respectively, converting a string of characters to a (mel-scaled) spectral representation of the voice signal and converting the spectral representation to an actual waveform. Optionally, input text –apart from normalisation– undergoes phonemisation to present the input to the spectrogram predictor as a sequence of *phonemes* rather than *graphemes*.

Recent end-to-end solutions for spectrogam prediction are built with and *encoder-decoder* architecture (Wang et al., 2017; Shen et al., 2018; Ping et al., 2018; Ren et al., 2019). The encoder maps the input sequence to a hidden continuous space, and the decoder takes care of generating autoregressively the spectrogram from the hidden representation. To produce the alignment between encoder and decoder, an *attention mechanism* (Bahdanau et al., 2015) is introduced between these two blocks.

Among the available architectures for spectrogram prediction, *Tacotron* (Wang et al., 2017), and in particular its advanced version Tacotron 2 (Shen et al., 2018), seems to be the most flexible and re-usable.

Many works have been developed to introduce conditioning into Tacotron, obtaining a fine-grained control over different prosodic aspects. The *Global Style Token* (GST) approach enabled control over the speaking style in an unsupervised manner (Wang et al., 2018). Another controllable aspect is the speaker voice, introduced through additional *speaker-emebeddings* extracted through a speaker verification network (Jia et al., 2018). Finally Suni et al. (2020) proposed a methodology to control *prominence* and *boundaries* by automatically deriving prosodic tags to augment the input character sequence. It is also possible to combine multiple techniques into a single conditioned architecture, as shown by Skerry-Ryan et al. (2018).

Neural vocoders completed the deep learning TTS pipeline improving consistently the quality of synthesised voice (van den Oord et al., 2016; Kalchbrenner et al., 2018; Kumar et al., 2019; Yang et al., 2021). These vocoders substituted the Griffin-Lim algorithm (Griffin and Lim, 1983), which was characterised by artifacts and poor audio quality, especially if compared with newer neural approaches. These components, differently from the spectrogram predictors, do not strictly depend on the input language. Their primary role is to invert a spectral representation into the time domain; thus, they are thought to be *language-agnostic*.

As premised, the available models are primarily trained and evaluated on English corpora due to data availability. A general solution for data scarcity is to leverage a technique called *transfer learning* (Yosinski et al., 2014), which consists of re-using the hidden layers of a pre-trained deep neural network as inputs for a different task. For our work, we applied a variant of transfer learning called *fine-tuning*, where we used the pre-trained weights of the network as initialisation for the actual training on the new task (Yosinski et al., 2014).

3 Corpora

For the scope of this work, we considered three different corpora of Italian speech. All corpora are composed of read speech. We reported the main statistics about the corpora in Table 1. All clips were re-sampled at 22 050 Hz.

Mozilla Common Voice[1] (MCV) is a publicly available corpus of crowd-sourced audio recordings (Ardila et al., 2020). Contributors can either donate voice by reading a prompted sentences or validate clips by listening to others' recordings. The samples in this corpus have a sample rate of 48 000 Hz.

VoxForge[2] (VF) is a multilingual open-source speech database that includes audio clips collected from speaker volunteers. The samples in this corpus have a sample rate of 16 000 Hz.

Ortofonico (Ort.) is a subset of the CLIPS[3] corpus, a corpus of Italian speech collected for a project funded by the Italian Ministry of Education, University and Research. Audio recordings come from radio and television programs, map task dialogues, simulated conversations, and text excerpts read by professional speakers. The samples in this corpus subset have a sample rate of 22 050 Hz.

Apart from the three presented corpora, we used some clips from a private collection of audiobooks in the human evaluation step. We reported further details in Section 5.

4 ITAcotron 2 synthesis pipeline

The model we proposed and evaluated is called ITAcotron 2. It is an entire TTS pipeline, complete with speaker conditioning, based on Tacotron 2 (Shen et al., 2018; Jia et al., 2018). The pipeline is composed of a phonemiser, a speaker encoder (used for the conditioning step), a spectrogram predictor, and a neural vocoder. We reported a scheme of the pipeline in Figure 1.

The core part of the model we are presenting is the spectrogram predictor. We referred to the Tacotron 2 implementation and weights provided

[1] https://commonvoice.mozilla.org
[2] http://www.voxforge.org
[3] http://www.clips.unina.it

Table 1: Statistics on the considered corpora for the Italian fine tuning of the spectrogram predictor: Mozilla Common Voice (MCV), VoxForge (VF) and Ortofonico (Ort.).

Corpus	Time (h)			Clips			Speakers		
	Train	Validation	Test	Train	Validation	Test	Train	Validation	Test
MCV	79.07	26.45	26.42	50 322	16 774	16 775	5151	3719	3743
VF	13.62	1.74	1.75	7176	913	918	903	584	597
Ort.	2.94	0.36	0.32	1436	164	159	20	20	20

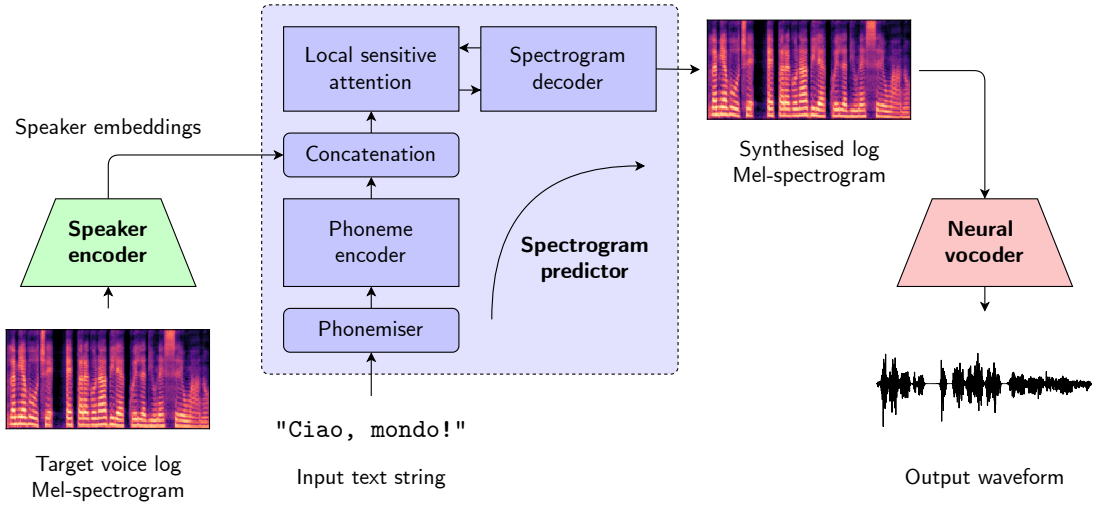


Figure 1: ITAcotron 2 synthesis pipeline.

by Mozilla[4] (Gölge, 2020). The model uses a *phoneme encoder* to represent the input sequence to utter, and an *autoregressive decoder* to generate the target spectrogram; an intermediate attention mechanism provides the input-output alignment. With respect to the original implementation, we only extended the employed phonemiser[5] to accommodate Italian's accented vowels as additional input characters. Code and pre-trained weights for speaker encoder and vocoder came from the Tacotron 2 same source.

We divided the fine-tuning process of the spectrogram predictor into two steps. In this way, we iteratively improved the output quality.

The former used only the data coming from the MCV corpus, which constituted the majority of the available data. Due to the low quality of the input audio recordings, we leveraged this step mostly to drive the network's weights towards the target language. The noisy and sometimes poorly uttered clips of this corpus resulted in an awful quality of the synthesised clips, which sometimes were impossible to understand. This fine tuning was performed for 52 271 update steps (identified trough validation) on mini-batches containing 64 clips each (Other hyper-parameters were left unchanged from the reference implementation).

The latter fine-tuning leveraged both VF and Ort. corpora. Audio clips in these corpora had a noticeable higher quality than those of MCV in terms of audio cleaning and speaker articulation. As a result, the outputs of this final stage had significantly less background noise, and the content was highly intelligible. We performed this second fine-tuning for 42 366 update steps (identified trough validation) on mini-batches containing 42 clips each (Other hyper-parameters were left unchanged from the reference implementation).

To achieve speaker conditioning, we concatenated the encoder representation of the spectrogram predictor with a *speaker embedding*. These embeddings are extracted from a speaker verification model (Chung et al., 2020), similar to that of the reference work by Jia et al. (2018). For the

[4]Repository link: https://github.com/mozilla/TTS, reference commit link: https://github.com/mozilla/TTS/tree/2136433

[5]https://pypi.org/project/phonemizer/

vocoder, instead, we adopted the more recent *Full-Band MelGAN* (FB-MelGAN) vocoder (Yang et al., 2021).

Notice that while we fine-tuned the spectrogram synthesis network, we did not apply the same process to the speaker embedding and neural vocoder networks. We did so because we wanted to observe the zero-shot behaviour of these networks in the new language. In this way, we could assess whether the two models are language-agnostic.

5 Evaluation approach

Similarly to Jia et al. (2018), we divided the evaluation process of the fine-tuned model into two listening tasks:

- evaluation of *Intelligibility and Naturalness* (I&N) of the speaker-conditioned synthesised samples;
- evaluation of *Speaker Similarity* (SS) of the speaker-conditioned synthesised samples.

For both tasks we asked subjects to rate different aspects in a 1 to 5 scale, with 0.5 increments (ITU-T Recommendation, 1999), of the various stimuli (i.e. audio clips). We divided the 70 participants into 20 experimental groups for both listening tasks. We prompted participants of each group with the same stimuli.

In the I&N tasks, we assigned each group with 4 clip pairs, for a total of 160 clips among all groups. Each clip pair was composed of a real clip (ground truth) coming from one of the corpora (including an additional private corpus of audio-books) and a synthetic clip generated in the voice of the ground truth, but with different speech content (i.e. the same voice uttered a different sentence). At this step, we asked subjects to rate the intelligibility and naturalness of each clip separately. Clips were presented in a random order (to avoid biases) and were rated right after listening.

In the SS tasks, we assigned each group with with 16 clips split into 4 subsets, for a total of 160 clips among all groups. We divided the SS task into three further sub-tasks. Each subset was composed of a synthetic clip and three real clips. Subjects compared the synthetic clip to each of the other three real clips:

1. real clip containing an utterance in the voice of the same speaker of the synthetic one (*same speaker* comparison sub-task);
2. real clip containing an utterance in the voice of a different speaker having the same gender of the speaker of the synthetic one (*same gender* comparison sub-task);
3. real clip containing an utterance in the voice of a different speaker having different gender of the speaker of the synthetic one (*different gender* comparison sub-task).

At this step, we asked subjects to rate how similar the synthetic voice was to the one we paired it with (knowing that the fixed clip was synthetic and the other three real). Real clips were presented in a random order (to avoid biases), and subjects rated the similarity after listening to a synthetic-real pair.

6 Results

Table 2: Results of the listening tasks. MOS values are reported as *average* $\pm$ *standard deviation*.

Task	Sub-task	Model	MOS
I&N	Intelligibility	ITAcotron 2	4.15 ± 0.78
		Ground truth	4.43 ± 0.74
	Naturalness	ITAcotron 2	3.32 ± 0.97
		Ground truth	4.28 ± 0.86
SS	Same speaker	ITAcotron 2	3.45 ± 1.07
	Same gender	ITAcotron 2	2.78 ± 1.01
	Different gender	ITAcotron 2	1.99 ± 1.08

We reported the Mean Opinion Score (MOS) of each task in Table 2. The overall scores were satisfying and reflected the intentions and the expectations underlying this research.

Concerning the I&N evaluation, the first thing that jumps to the eye is the high intelligibility score, very close to real clips. This high score provides clear evidence of how easy it was to understand the linguistic content of the synthetic clips. The naturalness score is lower than that of intelligibility, meaning that it is still possible to distinguish between real and fake clips.

Concerning the SS evaluation, instead, the thing that jumps to the eye is the progressive drop in the MOS value. This reduction is precisely the expected behaviour: changing the speaker should lead to lower similarity, especially when the two speakers have different gender. The value obtained for the same speaker sub-task seems promising. The reduction in speaker similarity observed in different speaker sub-task showed that the synthetic

clips' voice is distinguishable from those of the same gender. The further drop observed in different speaker similarity evaluations underlined that the network learned to separate even better these aspects, as we expected considered the general difference in pitch ranges between the two genders (Leung et al., 2018).

The figures we obtained are quite similar to those obtained by Jia et al. (2018) on similar tasks for English. However, we choose not to report a direct comparison against the work mentioned above as it focuses on English and the tasks are not perfectly comparable with ours. Nevertheless, obtaining scores that are similar to the ones provided by that work, is a hint that our approach seems sound.

7 Conclusion

This paper showed the approach we followed in our work to adapt a speech synthesis pipeline from English to Italian. The procedure is language-agnostic; however, the spectrogram prediction network requires fine-tuning data in the target language. To show how some pipeline components can be used out-of-the-box (i.e. without language adaptation), we also introduced a speaker embedding network (to achieve speaker conditioning) and a neural vocoder. Opinion scores from a human evaluation session showed that the adaptation was successful in terms of intelligibility and naturalness. Concerning speaker conditioning, the result was not as sharp as for the first evaluation, yet we obtained a satisfying similarity score, matching that of the reference model.

In future work, to derive speaker discriminative representations, we could refine the speaker encoder on Italian multi-speaker speech data. In doing so, we will assess the impact of employing a network refined on a target language for deriving descriptive features for speakers of that language. Finally, since ITAcotron 2 is not completely able to isolate the speaker voiceprint from the prosody of the reference audio, we suggest conditioning its generative performance on independent auxiliary representations as in Skerry-Ryan et al. (2018) and Wang et al. (2018). For instance, one intended to capture the speaker's accent and one the speaker's voiceprint.

Acknowledgments

This work was partially supported by the European Union's Horizon 2020 project *WorkingAge* (grant agreement No. 826232).

References

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis M. Tyers, and Gregor Weber. 2020. Common voice: A massively-multilingual speech corpus. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 4218–4222. European Language Resources Association.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Joon Son Chung, Jaesung Huh, Seongkyu Mun, Minjae Lee, Hee-Soo Heo, Soyeon Choe, Chiheon Ham, Sunghwan Jung, Bong-Jin Lee, and Icksang Han. 2020. In defence of metric learning for speaker recognition. In *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 2977–2981. ISCA.

Daniel W. Griffin and Jae S. Lim. 1983. Signal estimation from modified short-time fourier transform. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '83, Boston, Massachusetts, USA, April 14-16, 1983*, pages 804–807. IEEE.

Eren Gölge. 2020. Solving attention problems of tts models with double decoder consistency.

Po-chun Hsu, Chun-hsuan Wang, Andy T. Liu, and Hung-yi Lee. 2019. Towards robust neural vocoding for speech generation: A survey. *CoRR*, abs/1912.02461.

ITU-T Recommendation. 1999. *P.910: Subjective video quality assessment methods for multimedia applications*.

Ye Jia, Yu Zhang, Ron J. Weiss, Quan Wang, Jonathan Shen, Fei Ren, Zhifeng Chen, Patrick Nguyen, Ruoming Pang, Ignacio Lopez-Moreno, and Yonghui Wu. 2018. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 4485–4495.

Daniel Jurafsky and James H. Martin. 2009. *Speech and Language Processing (2nd Edition)*. Prentice-Hall, Inc., USA.

Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aäron van den Oord, Sander Dieleman, and Koray Kavukcuoglu. 2018. Efficient neural audio synthesis. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2415–2424. PMLR.

Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brébisson, Yoshua Bengio, and Aaron C. Courville. 2019. Melgan: Generative adversarial networks for conditional waveform synthesis. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 14881–14892.

Yeptain Leung, Jennifer Oates, and Siew Pang Chan. 2018. Voice, articulation, and prosody contribute to listener perceptions of speaker gender: A systematic review and meta-analysis. *Journal of Speech, Language and Hearing Research (Online)*, 61(2):266–297.

Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. In *The 9th ISCA Speech Synthesis Workshop, Sunnyvale, CA, USA, 13-15 September 2016*, page 125. ISCA.

Wei Ping, Kainan Peng, Andrew Gibiansky, Sercan Ömer Arik, Ajay Kannan, Sharan Narang, Jonathan Raiman, and John Miller. 2018. Deep voice 3: Scaling text-to-speech with convolutional sequence learning. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2019. Fastspeech: Fast, robust and controllable text to speech. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 3165–3174.

Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ-Skerrv Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu. 2018. Natural TTS synthesis by conditioning wavenet on MEL spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*, pages 4779–4783. IEEE.

R. J. Skerry-Ryan, Eric Battenberg, Ying Xiao, Yuxuan Wang, Daisy Stanton, Joel Shor, Ron J. Weiss, Rob Clark, and Rif A. Saurous. 2018. Towards end-to-end prosody transfer for expressive speech synthesis with tacotron. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 4700–4709. PMLR.

Antti Suni, Sofoklis Kakouros, Martti Vainio, and Juraj Simko. 2020. Prosodic prominence and boundaries in sequence-to-sequence speech synthesis. *CoRR*, abs/2006.15967.

Yuxuan Wang, R. J. Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc V. Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif A. Saurous. 2017. Tacotron: Towards end-to-end speech synthesis. In *Interspeech 2017, 18th Annual Conference of the International Speech Communication Association, Stockholm, Sweden, August 20-24, 2017*, pages 4006–4010. ISCA.

Yuxuan Wang, Daisy Stanton, Yu Zhang, R. J. Skerry-Ryan, Eric Battenberg, Joel Shor, Ying Xiao, Ye Jia, Fei Ren, and Rif A. Saurous. 2018. Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 5167–5176. PMLR.

Geng Yang, Shan Yang, Kai Liu, Peng Fang, Wei Chen, and Lei Xie. 2021. Multi-band melgan: Faster waveform generation for high-quality text-to-speech. In *IEEE Spoken Language Technology Workshop, SLT 2021, Shenzhen, China, January 19-22, 2021*, pages 492–498. IEEE.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3320–3328.

A Code base and model weights

The source code developed during this project is available at the following link: `https://github.com/vincenzo-scotti/ITAcotron_2`. Inside the repository we also provide the links to download the weights of the fine-tuned model ITAcotron 2, for Italian speech synthesis. We remind that the original source code we forked, and the weights of the speaker encoder and neural vocoder, were taken from the reference open source project developed by Mozilla[4].

Supporting Undotted Arabic with Pre-trained Language Models

Aviad Rom and **Kfir Bar**
The Data Science Institute, Reichman University, Herzliya, Israel
{aviad.rom,kfir.bar}@post.idc.ac.il

Abstract

We observe a recent behaviour on social media, in which users intentionally remove consonantal dots from Arabic letters, in order to bypass content-classification algorithms. Content classification is typically done by fine-tuning pre-trained language models, which have been recently employed by many natural-language-processing applications. In this work we study the effect of applying pre-trained Arabic language models on "undotted" Arabic texts. We suggest several ways of supporting undotted texts with pre-trained models, without additional training, and measure their performance on two Arabic natural-language-processing downstream tasks. The results are encouraging; in one of the tasks our method shows nearly perfect performance.

1 Introduction

Arabic is a highly inflected Semitic language, spoken by almost 400 million native speakers around the world. Arabic words are highly ambiguous, mostly due to the lack of short vowels, represented by diacritic vocalization marks, which are typically omitted in standard writing. Modern Standard Arabic (MSA), is the language that is used in official settings, while the dialectal variants of Arabic are used in day-to-day conversations. In addition to vocalization marks, some Arabic letters carry dots, called *i'jaam* (إِعْجَام), which are used to distinguish between consonants represented by the same orthographic form, or *rasm* (رَسْم) in Arabic. For example, the letters Tā (ت), Yā (ي), Thā (ث), Bā (ب), and Nun (ن) have exactly the same orthographic shape, excluding the number and location of the dots they carry. Without the dots, the letter remains ambiguous. Nevertheless, some dots are sometimes forgotten in handwritten scripts, forcing the reader to use the surrounding context in order to resolve such ambiguities. It becomes slightly more complicated when some of the letters turn into other Arabic letters after their dots are being removed. For example, by removing the three dots from the letter Shīn (ش) we get the letter Sīn (س), and that makes different words look the same. For example, the reader may have difficulty understanding the meaning of the word سعب (*sa'b*), which can be interpreted without the dots as "sigh", and with the dots شعب (*sha'b*) as "people". Additional examples are provided in Table 2.

Fortunately, dots are strictly used in digitized texts. However, we have noticed a recent trend of removing dots from Arabic posts on social media (Drißner, 2021)[1], where people use special keyboards and applications to naturally write without dots, mainly for bypassing automatic content-filtering algorithms to avoid having their message classified as offensive. It seems like most native Arabic speakers can still understand the meaning of the text, even if provided dotless. Table 1 shows an example for a text and its undotted version.

The use of dots for distinguishing between consonants was introduced to the Arabic language after the rise of Islam, when non native speakers started showing interest in the new religion. Until that time, the knowledge of how to pronounce undotted text was based on the reader's memory and the surrounding context (Daniels, 2014).

The use of Transformer (Vaswani et al., 2017) in natural language processing (NLP) has become fundamental to achieve state-of-the-art results in different downstream tasks, including content filtering. Since Transformer-based language models are trained with digitized texts, the vocabulary acquired from the data is represented with dots. Therefore, the undotted letters that are not part of the official Arabic language, are not recognized by the model, even if they exist in the Unicode character set (e.g., "Dotless Archaic Beh" [ٮ]).

In this work, we study the effect of removing dots from text written in Arabic, on the perfor-

[1] https://arabic-for-nerds.com/dotless-arabic/

mance of a Transformer-based language model, employed as a typical content-filtering classifier. Our results show that replacing the dotted MSA letters with their corresponding dotless versions, causes a strong adversarial effect on the performance of the language model that was fine-tuned on various downstream tasks. We describe our attempts to handle undotted Arabic, none of them require re-training the language model, and discuss their results and potential contributions.

2 Related Work

2.1 Arabic Transformer Models

Multilingual BERT, or mBERT (Devlin et al., 2019), was the first pre-trained language model to include Arabic. It covers only MSA, and usually do not perform well enough on downstream Arabic NLP tasks, due to the relatively small Arabic training data it was trained on. AraBERT (Antoun et al., 2020) and GigaBERT (Lan et al., 2020) are two language models that were trained on a much larger portion of Arabic texts, still only MSA. Both offer better performance on downstream tasks. Two recent models, MARBERT (Abdul-Mageed et al., 2021a), and CAMeLBERT (Inoue et al., 2021), include Dialectical Arabic in their training data, reaching better performance on relevant tasks. None of these models have been used with undotted Arabic, which is the main focus of our work.

2.2 Adversarial Inputs in NLP

Adversarial inputs are crafted examples to deceive neural networks at inference time. Such attacks have already been introduced and discussed by Szegedy et al. 2013 and Goodfellow et al. 2015, focusing mostly on adversarial perturbations in vision tasks. Generating adversarial inputs in NLP is considered to be more challenging than in computer vision, mostly due to the relatively large importance every word has in a given input text, comparing to the small importance a single pixel has in an input image. Nonetheless, it has been recently addressed by Jin et al. (2020), who presented an efficient way of generating adversarial textual inputs for a BERT (Devlin et al., 2019), by modifying the texts semantically based on some word statistics taken from the language model itself. They showed that while their modified texts are understandable by human readers, their BERT-based models have struggled to produce the correct output. In this work, we evaluate a more natural approach for fooling an Arabic language model, simply by converting some letters to their undotted versions, keeping the modified text understandable for human readers.

3 Handling Undotted Arabic

We begin by fine-tuning a Transformer-based Arabic language model on two downstream tasks, and evaluate their performance on undotted inputs. Following that, we develop different computational approaches for recovering the missing information that was lost with undotting, without pre-training the language model itself. We evaluate the different approaches on the same downstream tasks, and report on the results in the following section. For all our experiments we use the recent CAMeLBERT-Mix base model (Inoue et al., 2021), which was pre-trained on a mix of MSA, Classical Arabic, and Dialectical Arabic texts.

3.1 Undotting

In order to remove dots from the text, we created a mapping for all the Arabic characters available in the Unicode character set, for which we match the most resemblant undotted character. The mapping table is provided in Appendix A. Some Arabic letters have different forms, depending on whether they appear at the beginning, middle or end of a word. Therefore, we map all the forms of a relevant letter. Undotting an input text is a simple replacement of all relevant letters with their orthographic equivalents.

3.2 Supporting Undotted Arabic

As reported in the following section, fine-tuned Arabic language models do not perform well on undotted texts. Therefore, we suggest two ways to handle undotted texts. In one way, we make changes to the tokenizer of the model, and in another way we develop an algorithm for restoring the dots of the input text, which runs as a preprocessing step before submitting the text to the language model.

3.2.1 Changing the Tokenizer

Before processing the text with a pre-trained language model, it is necessary to break it into tokens using the same tokenizer that was used during the pre-training phase of the model. CAMeLBERT-Mix uses a standard BERT tokenizer, provided by Hugging Face[2], with a vocabulary of 30,000 tokens.

[2] https://github.com/huggingface/tokenizers

Original	وتعد آقاروة إحدى الجزر الكويتية التسع التي تنتشر في المياه الإقليمية الكويتية
Undotted	وبعد آقاروة إحدى الحرر الكوسه السع الى سسر ى الماه الإقلمه الكوسه
Translation	Qaruh is one of the nine Kuwaiti islands in the Kuwaiti territorial waters

Table 1: Arabic text, given with and without the dots. The text was taken from https://arabic.cnn.com/travel/article/2021/07/13/qaruh-island-kuwait.

Each token has a numeric identifier. We take two different approaches for changing the configuration of the tokenizer in order to handle undotted texts without having to pre-train the language model, nor fine-tuning it on a downstream task.

Undotting the Tokenizer Vocabulary. According to this approach, we undot the entire vocabulary of the tokenizer, thereby enabling it to seamlessly recognize undotted letters and words. Obviously, after undotting the vocabulary some of the tokens (5,852 out of the original 30,000 tokens, or 19.52%) become identical, leaving some of the token identifiers unused; therefore, the model's vocabulary get smaller. Since we suspect that working with a smaller vocabulary may be detrimental to the performance of the model on downstream tasks, we suggest another approach for modifying the tokenizer.

Extending the Tokenizer Vocabulary. Under this approach, we extend the tokenizer's vocabulary by adding the undotted version of the relevant tokens and mapping them to the same identifier of their original token. This way the tokenizer keeps the original dotted version of every token, and thus can accept both, dotted and undotted inputs. We add the undotted version of a token only if it is not already part of the vocabulary; overall, we added 17,280 undotted versions. The resulting vocabulary has about 57% token identifiers that are mapped to two token versions.

3.2.2 AReDotter: Restoring Arabic Dots

As opposed to the previous approach, here we develop an algorithm for pre-processing the input undotted text to restore its dots. The language model itself remains unmodified.

We train a sequence-to-sequence machine-translation (MT) model on the unlabeled 10M Arabic tweets dataset published with the second NADI shared task (Abdul-Mageed et al., 2021b). The tweets were posted from multiple geographies.

For creating parallel texts for training, every tweet from the original corpus was paired with its automatically generated undotted version, using the mappings provided in Appendix A. We remove from the tweets URLs, user mentions, and hashtags.

Our MT model is based on the pre-trained Arabic-to-English Marian MT (Tiedemann and Thottingal, 2020) architecture[3], which was fine-tuned for "undotted Arabic"-to-Arabic translation. We fine-tune our model on the entire parallel dataset for two epochs.

4 Experimental Results

To evaluate our proposed methods, we fine-tune CAMeLBERT on two tasks, sentence level and token level.

For the sentence-level task we use the sentiment analysis subtask of ArSarcasm-v2 (Abu Farha et al., 2021), designed as a three-labels (positive, negative, neutral) classification task. As we did with NADI, we preprocess the text to remove URLs, user mentions, and hashtags. For evaluation, we use the official evaluation objective metric, defined as macro average F1 score of both non-neutral labels.

For a token-level downstream task, we evaluate our language model on the named-entity recognition (NER) task using the ANERcorp dataset (Benajiba et al., 2007). We use the modified version of the dataset, which was recently released by Obeid et al. (2020). Following previous works on NER, we use the micro average F1 metric for evaluation.

For each task, we fine-tune CAMeLBERT on the original preprocessed training data for 10 epochs, using the official train/test split, and evaluate it on the undotted version of the test set. We use the standard Hugging Face's pipelines, AutoModelForSequenceClassification and AutoModelForTokenClassification[4] for the sentiment analysis and NER tasks, respectively. We evaluate the models under different conditions of supporting undotted

[3] Specifically, we used the Helsinki-NLP/opus-mt-ar-en model from Hugging Face.

[4] https://huggingface.co/transformers/model_doc/auto.html

Undotted	Option 1 (pronunciation, meaning)	Option 2 (pronunciation, meaning)
فحب	فيجب (*fyajib*, "must")	فتحت (*fatahat*, "opened")
بفارق	تفارق (*tafaruq*, "leave")	بفارق (*bifariq*, "difference")
بحار	نجار (*najaar*, "carpenter")	بحار (*bahaar*, "seas")
حبوب	حبوب (*hubub*, "cereal")	جنوب (*janub*, "south")

Table 2: Examples of undotted ambiguous words. We do not provide all the possible pronunciations in each row.

	ArSarcasm V2 - Sentiment	**ANERCorp**
Original Text	70.55	81.39
Undotted Text	44.86	9.16
Undotted Text + Undotted Tokenizer	64.50	72.85
Undotted Text + Extended Tokenizer	65.03	71.68
AReDotter	68.27	67.97

Table 3: Model performance on downstream tasks, using different undotted text handling approaches.

texts, as described in the previous section.

4.1 Results

The results, reported in Table 3, demonstrate the adversarial effect of processing undotted Arabic with a vanilla, unmodified CAMeLBERT model. The first row lists the results we get by working with the original texts. In the second row we provide the results of using the same model, but this time applied on the undotted version of the texts. As observed, the metrics measured for the two tasks dropped significantly on undotted texts. Unsurprisingly, the tokenizer of the vanilla language model does not recognize tokens with undotted letters, which are excluded from the modern Arabic script, and thus treating them as "unknown" tokens.

The two tokenizer-updating approaches, whose results are reported in the 3rd and 4th rows, prove to be effective for undotted texts, in both tasks. This improvement is achieved mainly due to the reduction in the number of unknown tokens the model is assigned with. Among the two, we observe that the extended tokenizer is slightly better on the sentiment analysis task, while the undotted tokenizer is better on the NER task. However, the difference in those results is insignificant.

Interestingly, AReDotter, our MT dots restoration model, which we run as a preprocessor before submitting the text to the language model, provides competitive results in both task. It is slightly better than the tokenizer-updating techniques on sentiment analysis, but slightly worse on the NER task. Naturally, a sequence-to-sequence translation model may sometimes generate some out-of-context tokens in the target sequence. We believe that NER is more sensitive to this type of mistakes than sentiment analysis task. For future work, we plan to work with a simple sequential-tagging model instead of the sequence-to-sequence MT model, to avoid generating such tokens. The results we get from AReDotter are encouraging; it provides an elegant way to support undotted text without modifying the model or the tokenizer.

5 Conclusion

Undotting has been recently adopted by social-media users in order to bypass content-filtering gateways. We studied the effect of undotting on the performance of a standard pre-trained language model. Our results show that processing undotted text with a vanilla, unmodified language model, has a detrimental effect in two downstream NLP tasks. By simply editing the tokenizer, which is used by the language model, we are able to show significant improvements over the vanilla model.

Our third approach, which does not require changing the tokenizer, is using a machine-translation model for restoring the missing dots. With this technique we show competitive results to the tokenizer-updating techniques, without having to modify the model or its tokenizer. We believe that our study provides some conclusions as for how undotted texts should be treated with modern Transformer-based language models. We recommend that at least one of our techniques will be adopted as a standard step in a common Arabic NLP pipeline.

References

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021a. ARBERT & MARBERT: Deep Bidirectional Transformers for Arabic. In *Proceedings of the ACL-IJCNLP 2021 Main Conference*. Association for Computational Linguistics.

Muhammad Abdul-Mageed, Chiyu Zhang, Abdel-Rahim Elmadany, Houda Bouamor, and Nizar Habash. 2021b. NADI 2021: The second nuanced Arabic dialect identification shared task. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 244–259, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Ibrahim Abu Farha, Wajdi Zaghouani, and Walid Magdy. 2021. Overview of the WANLP 2021 shared task on sarcasm and sentiment detection in Arabic. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 296–305, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resource Association.

Yassine Benajiba, Paolo Rosso, and José Miguel BenedíRuiz. 2007. ANERsys: An Arabic named entity recognition system based on maximum entropy. In *Computational Linguistics and Intelligent Text Processing*, pages 143–153, Berlin, Heidelberg. Springer Berlin Heidelberg.

Peter T. Daniels. 2014. *The Type and Spread of Arabic Script*, pages 25 – 39. Brill, Leiden, The Netherlands.

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.

Gerald Drißner. 2021. Social media & palestine: Dotless Arabic outsmarts algorithms.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*.

Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The interplay of variant, size, and task type in Arabic pre-trained language models. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, Kyiv, Ukraine (Online). Association for Computational Linguistics.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is BERT really robust? A strong baseline for natural language attack on text classification and entailment. 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 8018–8025. AAAI Press.

Wuwei Lan, Yang Chen, Wei Xu, and Alan Ritter. 2020. An empirical study of pre-trained transformers for Arabic information extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4727–4734, Online. Association for Computational Linguistics.

Ossama Obeid, Nasser Zalmout, Salam Khalifa, Dima Taji, Mai Oudah, Bashar Alhafni, Go Inoue, Fadhl Eryani, Alexander Erdmann, and Nizar Habash. 2020. CAMeL tools: An open source python toolkit for Arabic natural language processing. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 7022–7032, Marseille, France. European Language Resources Association.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

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.

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

A Appendix A: Undotting Table

Letter	MSA	Initial/Medial Undotted	Terminal Undotted
Ba	ب	ـبـ	ـب
Ta	ت	ـتـ	ـت
Tha	ث	ـثـ	ـث
Jim	ج	ـحـ	ـح
Kha	خ	ـحـ	ـح
Dhal	ذ	د	ـد
Zayn	ز	ر	ـر
Shin	ش	ـسـ	ـس
Dad	ض	ـصـ	ـص
Za’	ظ	ـطـ	ـط
Ghayn	غ	ـعـ	ـع
Fa	ف	ـفـ	ـف
Qaf	ق	ـفـ	ـق
Nun	ن	ـنـ	ـن
Ya	ي	ـيـ	ـى

Table 4: Character mapping used for our undotting function, specifying only the letters which are not identical to their undotted version.

Identifying and Understanding Game-Framing in Online News: BERT and Fine-Grained Linguistic Features

Hayastan Avetisyan
German Centre for Higher Education Research and Science Studies
Hannover, Germany
avetisyan@dzhw.eu

David Broneske
German Centre for Higher Education Research and Science Studies
Hannover, Germany
broneske@dzhw.eu

Abstract

News providers tend to add an entertaining and eye-catching spin to online news stories by framing politics around strategies and tactics as well as wins and losses. Thus, they shape the reader's view on a particular subject, person, or event and influence their behavior regarding, for instance, voting respectively. To address this issue, we introduce the first attempt in computational linguistics to model computational frame classification. We offer a human-labeled dataset indicating the issue-game framing in news media using a comprehensive range of linguistic features. Moreover, we present an overview of potential linguistic indicators of issue and game frames. Furthermore, we attempt to provide methods for analyzing, understanding, and flagging problems that deal with subjectivity with respect to framing by identifying and presenting the respective cues. We use BERT, a pre-trained word representation model, and fine-tune it with our dataset on the binary text classification task. It turns out that BERT-like approaches can be used to detect issue and game frames. However, studies utilizing more annotated training data should be conducted to investigate its universal effectiveness. We put forward some suggestions on the grammatical features that could be taken into consideration in the further development of similar language models.

1 Introduction

Daily we are plied with political messages from various sources. Those messages are more and more often framed around strategies, competitions, wins, and losses. Studies suggest that this framing of politics increases political distrust and cynicism (Cappella and Jamieson, 1996) negatively influences citizens' knowledge, attitude, and decisions (Aalberg et al., 2012). Thus, in one way or another, framing in political news media might shape the reader's view on a particular subject, person, or event. It might even destroy our collective trust and initiate social conflict (Pryzant et al., 2020). Therefore, the framing of political messages as a game can benefit from further research.

We find it crucial to discuss the attractiveness and popularity of the game frame in political news media coverage. Firstly, the news media tend to frame politics as a strategic game rather than to focus on political issues. Secondly, this kind of news coverage has increased over time. Thirdly, framing politics as a game increases political distrust and cynicism (Aalberg et al., 2012; Cappella and Jamieson, 1996). Moreover, it might also have a negative effect on citizens' knowledge acquisitions. One of the main reasons behind its popularity among scholars, however, might be the assumption that game framing might negatively influence democracy (Aalberg et al., 2012).

Even though the relatively few available studies have illustrated a correlation between the wording and grammar of political messages and attitudes regarding electability, the framing of political messages is still an understudied area (Tan, 2019). The profound significance of further research on this matter in terms of grammatical information being a "likely predictor of election outcomes" is emphasized.

Language can be viewed as the main instrument of politics and public opinion formation (Jahnen, 2019). Therefore, greater attention to the language of framing and the influence of linguistic details can lead to an increased awareness of political subject matters (Baumer et al., 2015; Fausey and Matlock, 2011). This could be accomplished by

researchers from computational linguistics and NLP whose extensive research on framing has already shown a positive effect (Baumer et al., 2015; Card et al., 2015; Choi and Palmer, 2012; Chong and Druckman, 2007). Additionally, Card et al. (2015) emphasize the potential contributions of computational linguists in formalizing and automating the analysis of framing. Therefore, our paper fills this gap by exploring the language of issue and game frames at multiple linguistic levels.

For this purpose, we constructed a human-labeled dataset of issue and game frames in news media annotated with a vast range of linguistic features at the following levels: syntactic, semantic, semantic-syntactic, and pragmatic. In summary, we contribute

1. a human-labeled corpus[1] of news articles, containing issue and game frames, annotated with a great range of linguistic features at the following levels: syntactic (both form and function levels), semantic, semantic-syntactic, and pragmatic,

2. an overview of potential linguistic indicators of issue and game frames, and

3. a starting point for future studies attempting to investigate issue and game frames by presenting an overview of potential linguistic indicators of the frames that should be taken into consideration.

To the best of our knowledge, this work is one of the first attempts in computational linguistics to model issue and game frames in news media.

2 Theoretical background

2.1 The concept of framing

Conceptually, framing has interdisciplinary roots in sociology, psychology, and linguistics. As the focus of our study is the issue and game frame in news media articles, we consider framing from the perspective of media and communication science (Brugman et al., 2017). Framing is the process of intentionally hiding or emphasizing facts in communication (Schäfer and O'Neill, 2016).

[1]The annotated corpus can be downloaded here.

2.1.1 Political news framing

Politicians often seek to make voters view their policies in a specific way (Chong and Druckman, 2007). They reach this aim by stressing the particular features of the policy but often leave out important facts (Ardèvol-Abreu, 2015).

2.1.2 Issue and game frame detection

There is a distinction between issue-specific and generic frames (Vreese de, 2005). Issue-specific frames are defined as those that are relevant only to particular topics or events. Whereas frames that go beyond thematic boundaries and can be determined with reference to various subject matters are called generic frames. The focus of our study, issue and game frame, belongs to the generic frames. In the following, we define the conceptual characteristics that are considered to indicate issue and game frames, respectively.

The focal points of the game frame are stories about depicting winning, along with the respective strategies, or losing elections. The game frame also comprises legislative debates or politics in general and is often associated with opinion polls and election results. It is characterized by depicting images of politicians, their tactics or strategies. Moreover, it is not uncommon for language of war or games to be used to describe the campaign. Politicians are quite often seen as persons rather than as spokespersons for specific policies. Therefore, elections are often depicted as personality contests emphasizing the performance, style, and personality of candidates (Aalberg et al., 2012; Cappella and Jamieson, 1996; Lawrence, 2000; Shehata, 2014; Jamieson, 1996).

The issue frame, on the other hand, is considered to be stories about the substance of policy problems and their possible solutions. It quite often tackles politicians' views on policy issues and as well as depictions of government programs and their impact on the public. The issue frame covers the substance of political problems, issues, and proposals, or any substantive issue (Aalberg et al., 2012; Lawrence, 2000).

Although it has been often stated that the issue frame can be contradictory to the game frame (Lawrence, 2000), the two frames may coexist in the same text and even complement

each other (Dekavalla, 2018). Different approaches to the coding process of the above-discussed frames have been employed. Two of the most prominent ones are: 1) coding the issue and game frames on a dominant frame basis, 2) investigating frames on a present-absent basis (Aalberg et al., 2012). In our study, we follow the latter approach.

2.2 Related NLP work on framing

The concept of framing and automated framing analysis is the subject of interest of a growing number of scholars.

Several NLP studies focus on public statements, congressional speeches, and news articles (Baumer et al., 2015; Card et al., 2015; Tsur et al., 2015). Other works investigate the process of identifying and measuring political ideologies, policies, and voting patterns (Johnson et al., 2017). Much NLP dwells on identifying entities or events, analyzing schemes or narrative events in terms of characters, inferring the relationships between entities, and predicting personality types from the text (Card et al., 2016). Johnson et al. (2017) focus on issue-independent framing analysis of US politicians on Twitter. They offer new Twitter-specific frames and provide weakly supervised models that extract tweets.

However, most of the research on the computational analysis of framing (Nguyen et al., 2015; Tsur et al., 2015; Baumer et al., 2015), focuses on one specific dimension or domain. Choi et al. (2012) explore the concept of hedging identifying it in the discussion of GMOs using an SVM trained on n-grams from annotated cue phrases.

Furthermore, Tsur et al. (2015) propose a new framework for automated analysis of an extensive collection of political texts demonstrating that topic ownership and framing strategies can be inferred using topic models.

Finally, Baumer et al. (2015) propose a classifier automatically identifying the language that is most related to framing. However, the study focuses on the language of framing in general, without giving special attention to any specific frames.

3 Labeling framing in news articles

In the following, we describe the process of labeling framing in our data, which comprises article selection, corpus construction, and linguistic annotation.

3.1 Article selection

Our specialized corpus has been created carefully to represent the written online media language regarding political news. The news articles in our corpus have different lengths and have been written independently. Quality newspapers were chosen over tabloids as they contain higher levels of journalistic interventionism, which is expressed through evaluations and an interpretative style (Bartholomé et al., 2018; Schmuck et al., 2017). Furthermore, we considered online articles rather than printed versions, as the online coverage has higher degrees of strategy reporting, and personal attacks are more prominently featured than in traditional media (Bartholomé et al., 2018). We chose the online versions of the New York Times[2] and Los Angeles Times[3] – the most circulated newspapers in the USA. We decided to focus solely on American newspapers to avoid linguistic issues and inconsistencies that might occur while including British newspapers. We decided not to consider issue and game frames separately since they are usually interwoven and complement each other (Dekavalla, 2018). Thus for convenience, we will refer to them as the issue-game frame from now on.

The following five topics were selected from the current events portal of Wikipedia: USA elections 2020, Donald Trump's impeachment, the Armenian genocide, Greta Thunberg, and Taiwan's elections. In total, 100 news articles were extracted, including image descriptions and titles that seemed relevant for our analysis. The initial data filtering, i.e., selecting the articles that might have the issue-game frame, was implemented during the corpus construction.

3.2 Corpus construction

While creating our corpus, we followed the three desiderata of corpus creation proposed by (Voormann and Gut, 2008): a sufficiently

[2]https://www.nytimes.com/
[3]https://www.latimes.com/

large and representative corpus, sufficient richness, and satisfactory accuracy of the annotations.

Our annotated dataset comprises 4063 statements (paragraphs), including 1519 positive and 2544 negative ones. For more information on the distibution of positive paragraphs across topics and newspapers, see Figure 1. Moreover, 6406 linguistic units displaying issue and game frames have been identified and annotated. Following Voormann and Gut (2008), we added two types of annotation: a) meta-information and b) linguistic information. Additionally, an extra annotation was added, namely the presence/absence of the issue-game frame. The meta-information includes the ID of the paragraph, source, topic, and file name.

The data processing consisted of several steps. Firstly, each article was stored in a separate text file. Secondly, the text was transported to an Excel table, where an ID was assigned to each paragraph. Being a coherent piece of writing, the paragraph was taken as a unit of analysis, and metadata variables (i.e., source, topic, and file name) were added. Once the data was processed, we examined all the paragraphs and assigned either the label "positive" or "negative", depending on the presence or absence (Aalberg et al., 2012) of any cues for issue and game frames.

Moreover, a few further articles had to be filtered out because of their type being an opinion article. The overall number of paragraphs before filtering was 4106 and 4063 after the deletion of paragraphs of opinion articles.

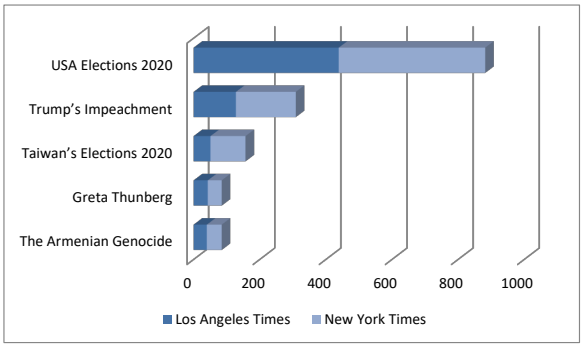


Figure 1: The distribution of the positive paragraphs in relation to the news media sources and topics.

3.3 Linguistic annotation

After the corpus construction, the linguistic features identifying the respective frames were identified and annotated. To this end, a qualitative corpus linguistic technique – a thorough inspection of the paragraphs at hand – was applied (see Figure 2). Our research approach was corpus-driven, i.e., we made minimal a priori assumptions regarding the linguistic features that should be employed for our following analysis.

Firstly, after a thorough examination of each paragraph, we extracted the phrases introducing the issue-game frame. Secondly, in the form of lemmas, the actual linguistic unit was extracted from the corresponding phrase. Thirdly, a linguistic analysis of the extracted unit was employed. To make the annotation schema as consistent as possible, we decided to consider words rather than phrases as the smallest linguistic units for our analysis. The terminology used for grammatical categories was considered from the lexical point of view.

We annotated our dataset at multiple linguistic levels to describe the linguistic units that display the issue-game frame linguistically as detailed as possible. Similar to other studies (Baumer et al., 2015; Matlock, 2012; Reah, 1998; Tan, 2019) on linguistic features identifying framing and their findings, we focused mainly on extracting grammatical information. Thus, the linguistic annotation of our corpus consists of the following features: a) Text, b) Phrase, c) Linguistic units of analysis, d) Syntax – form level, e) Syntax – form-level (feature description), f) Syntax – function level, e) Semantics (frames), g) Syntax-Semantics (negation), h) Pragmatics (discourse markers), i) Pragmatics (categories of discourse markers).

4 Experimental setup and evaluation

In the following, we present the implementation of our experiment and evaluate its performance in regards to predicted labels by briefly discussing the evaluation metrics.

4.1 Experimental settings

Our experiments were based on the implementation of BERT within Pytorch and HuggingFace. For the implementation of our binary

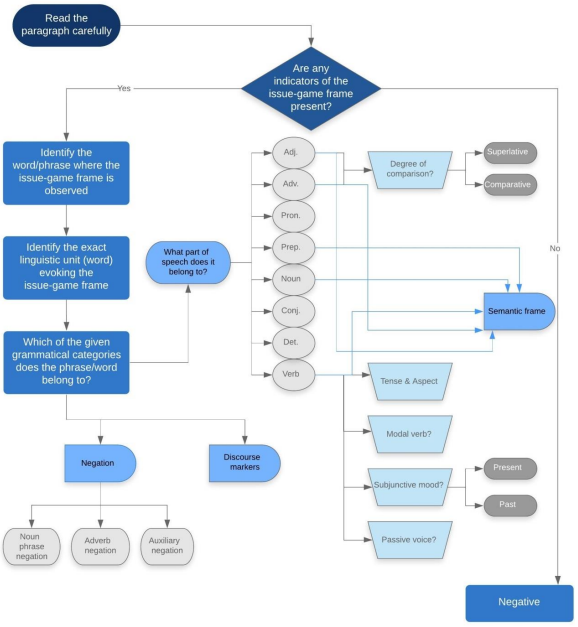


Figure 2: The flowchart of the linguistic annotation process.

classification task, we used the pre-trained BertForSequenceClassification[4]. We utilized the pre-trained model, added an untrained layer of neurons on the end, and trained the new model for our task. The training loop is based on the run_glue.py script (Wolf et al., 2020). We fine-tuned BERT on our corpus annotated with the issue-game frame. The dataset consists of two columns: – "label" and "text". The column "text" contains the paragraph, whereas "label" is a binary variable where "1" refers to "containing the issue-game frame" and "0" to "not containing the issue-game frame".

For splitting our dataset into train (70 %), validation (15 %), and test sets (15 %), we used the Scikit-Learn library, precisely the train_test_split method. Moreover, we added a random_state parameter to assure reproducibility. The hyperparameters followed those from the original BERT implementation. We set the batch size to 32, the sequence length to 300, and the base learning rate, as recommended in the original paper, to 5^{-5}. Moreover, we optimized using AdamW [5].

Since there is a class imbalance in our dataset (i.e., the majority of the sequences do not contain issue-game frames), we computed class weights for the labels in the train set and then passed those weights to the loss function to regulate the class imbalance. After multiple experiments and inspections of the training and validation sets learning values, we set the number of training epochs to 20.

[4] https://huggingface.co/transformers/model_doc/bert.html#bertforsequenceclassification

[5] An improved version of Adam (Kingma and Adam, 2017)

	Prec.	Recall	F1
0	0.79	0.77	0.78
1	0.63	0.65	0.64

Table 1: Evaluation metrics.

4.2 Evaluation

To evaluate the performance, we predicted the labels using our trained model and evaluated it against the true label. Afterwards, we reported the evaluation metrics through the classification report, including test accuracy, precision, recall, F1-score, which we show in Table 1.

Both recall and precision for class 1 are relatively high. We aimed at detecting sequences containing the issue-game frame, so misclassifying class 1 (holding the issue-game frame) samples is a more significant concern than misclassifying class 0 samples. The recall for class 1 is 0.65, which means that the model was able to classify 65 % of the paragraphs containing the frames correctly. However, the model misclassifies some of the class 0 sequences as containing the issue-game frame (precision: 0.63).

Matthews correlation coefficient – a balanced measure in classification problems – was also considered when evaluating the model. It can be used even if the classes are of very different sizes, which, in turn, is in accordance with our dataset. The total MCC score of our fine-tuned model is 0.423. This is quite an impressive outcome considering that the only hyperparameter tuning we carried out was adjusting the number of epochs from the recommended 2 or 4 to 20 epochs.

Moreover, we printed out the confusion matrix for visualizing and summarizing the performance of our fine-tuned model, i.e., to see how many sequences our model predicted correctly and incorrectly for each class. 149 out of 228 positive samples and 295 out of 382 negative samples were predicted correctly. Furthermore, 87 samples were wrongly classified as positive

(Type I Error) and 79 as negative (Type II Error).

5 Results and discussion

In the following, we introduce the findings of our study based on the manual annotation as well as its limitations. Furthermore, we analyze the performance of our fine-tuned model by interpreting the attention weights to different input elements.

5.1 Manual annotation

Our findings confirm the results of previous studies that the game frame is often characterized by war and sports language. Moreover, the results show that a significant portion of these words carries a negative connotation, which appears to be another indicator of the issue-game frame.

Furthermore, data regarding the personal pronouns *I* and *we* are in tandem with the findings of Bramley (2011); Alavidze (2017). In our data, politicians often use the personal pronoun *we* to create some sense of collectivism and "share the responsibility". The personal pronoun *I*, on the other hand, is often used by the speaker to show authority and personal responsibility along with commitment and involvement.

As for the use of passive voice, our data confirms the findings discussed by Tan (2019). We found instances of passive voice, where it is often used to either deflect blame or minimize emotional reactions. Moreover, our results show that past simple/present simple is the most common combination of tense and aspect used for the issue-game formation. As the imperfective framing evokes richer and more vivid action details in the minds of the readers than in the case of the perfective framing (Tan, 2019), the use of present simple and past simple might be employed to attract the reader's interest or to put the particular individual in a bad light.

Furthermore, our results show a connection between the language of the issue-game frame and subjectivity. Based on the framework proposed by Bednarek (2010), we assigned the evaluative parameters to the issue-framing words found in our data. Interestingly, lexical units belonging to each of the eleven parameters were identified. Additionally, based on the respective functions (Bednarek, 2010) of the parameters found in our data, we summarized the findings. We concluded that this type of evaluative vocabulary might be used to trigger positive or negative evaluations and emotions, evoke dramatization or intensification, signal, strengthen or mitigate evaluations, increase negative evaluation of news actors, or promote the negative as routine. Furthermore, it can contribute to the news values of eliteness, attribution, relevance, competition, and negativity, provide a sense of the reported discourse, comment on language activity (i.e., evaluate the style) or raise the interest of readers. Moreover, it might be employed while stating non-verifiable facts, attaching status, or lending reliability to expressions of the speaker's subjectivity.

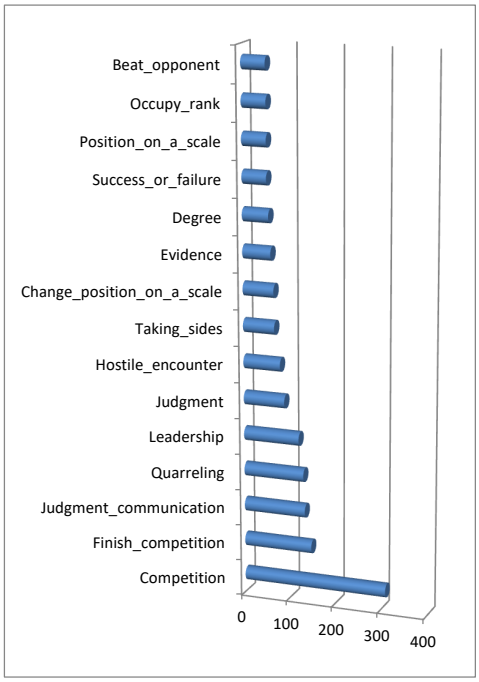


Figure 3: The number of occurrences of the 15 most frequent semantic frames.

Interestingly, our results regarding the insightful concept of semantic frames show that the five most frequent frames found in the linguistic units describing the issue-game frame (see Figure 3) corresponds to the contextual characteristics of frames discussed in Sec-

tion 2.1.2. Additionally, an interesting correlation between different topics and the dominant semantic frames should be mentioned. The Competition frame is among five most frequent frames in the articles covering the topic of elections. In comparison, the frame of Judgement was frequent in the articles regarding Greta Thunberg and Trump's impeachment. In four of the topics, the Judgment_communication frame was among the most common ones. Moreover, the Quarreling frame seems to be dominant within the articles discussing the Armenian Genocide, Trump's impeachment, and USA Elections 2020. A more detailed overview of the five topics in relation to the semantic frames is illustrated in Figure 4.

Figure 4: Most frequent semantic frames in relation to five topics.

Additionally, we found that our data corresponds to the existing linguistic cues of subjectivity, as well. The indicators of biased language proposed in the subjectivity literature (Biber and Finegan, 1989; Halliday, 2004; Hunston, 2011; Hunston and Sinclair, 2003; Thompson and Hunston, 2003; Labov, 1972) and found in our data, along with their respective examples, can be taken from the Appendix. These findings and the conclusion that the issue-game frame's language might be subjective/biased is based only on our corpus. Thus, a further investigation of this matter within a broader scope might be needed.

5.2 Limitations

Due to the limited scope of our study, we were not able to include and consider all grammatical features that might be useful for the annotation of our corpus of the issue-game frame.

Considering that manual annotation can often result in subjective results, we believe that the overall annotation can be improved with the help of a second annotator who could complete the same annotation task. Those results, afterwards, could be compared with the first annotation.

5.3 Comparison of the model's performance with our manual annotation

In order to analyze the performance of our fine-tuned model, we interpreted the attention weights assigned by the model to different input elements. Furthermore, we compared those results with the manually extracted features from our annotation to test if the model can reliably use the linguistic features defying the issue-game frame in the respective paragraphs. To implement this, we used the attention-head and the neuron views supported by the multiscale visualization tool BertViz (Vig, 2019). Due to the limited scope of our study, the analysis was carried out on a sample basis. The input sequences were carefully chosen to contain linguistic annotations at multiple levels within the same paragraph. Moreover, we aimed at involving samples from all event sets. As the visualizations work best with shorter sentences and may fail if the input text is very long (Vig, 2019), we created the respective visualizations for the chosen samples sentence by sentence.

The results of the analysis of the visualization of the attention weights of our fine-tuned model confirm the observation that "[...] attention in the Transformer correlates with syntactic constructs such as dependency relations and part-of-speech tags" (Vig, 2019). Therefore, the model is, indeed, able to identify the syntactic relationships between different words in a sentence, e.g., representative + of, have + accused. The visualization shows that attention is the highest between words within the same sentence, i.e., the model might understand that it should relate words to other words in the same sentence to understand their

context better. A significant portion of the words (e.g., accused, controversial, infighting, suffered, reclaim, speech) that were considered during the manual annotation are considered by the model, as well.

The performance of the model is similar across different event sets. However, it seems that the detection of deep linguistic features used to create issue-game frames, but not necessarily semantically or syntactically connected to the remaining elements of the sentence, might be a challenge for the model. In other words, issue and game frames quite often are not realized through dependency relations in a sentence. This might be due to the complex nature of framing and its realization at multiple linguistic levels. Furthermore, the linguistic features describing the issue-game frame at the pragmatic (discourse markers) and syntactic-semantic level (negation) might not be deeply identified by the BERT model yet.

6 Conclusion

This paper discussed the concept of issue and game frames in news media. We identified and defined them through linguistic means by manually annotating our corpus with linguistic information at multiple levels, i.e., syntactic, semantic, semantic-syntactic, and pragmatic. Based on our analysis, we presented an overview of potential linguistic indicators of issue and game frames in news media, along with respective examples from our data.

Furthermore, in accordance with the existing literature on this matter, we found cues for subjectivity and biased language in our data. Thus, showing that the language of issue and game frames in news media can, indeed, be subjective or biased.

Moreover, we evaluated the performance of the fine-tuned model and compared its results with our manual annotation. It can be concluded that BERT-like approaches can be used to detect issue and game frames. However, studies utilizing more annotated training data should be conducted to investigate its universal effectiveness. As to the question, whether BERT-like approaches understand and focus on similar linguistic cues as human annotators, it can be said that the model was able to identify some of the tokens our human annotation was based on. Nevertheless, identifying deep linguistic features at the pragmatic and syntactic-semantic levels seems challenging for the model. Thus, future development of the model should focus on these aspects, as well. However, since the comparison was conducted on a few samples, further research is needed to answer this question with more confidence.

For future work, the results of our analysis can be used in the development and automation of computational frame detection and classification. Furthermore, our study provides a starting point for future studies investigating the linguistic indicators of issue and game frames. It displays the grammatical features and levels of analysis that are crucial to the analysis of this matter. Our findings suggest that a further investigation of the following linguistic aspects and their inclusion in the annotation scheme might help to gain more insights into the process of formation of the issue-game frame: 1) semantic types of verbs or further analysis of modal verbs and the verbs they are followed by, 2) the concept of negation (types of negated verbs and nouns), 3) the linguistic theories of speech acts as well as topic and focus, 4) the concept of linguistic evidentiality, 5) thematic roles, semantic fields, and lexical fields, 6) lexico-grammatical patterns, 7) hidden metaphorical and ideological meanings. Some of the mentioned tasks might, however, be challenging in terms of their automation. Moreover, the results can be applied in further research on analyzing and understanding subjectivity in connection to framing. As a final point, we encourage leveraging our manual annotations in the process of developing models for the fine-grained issue and game framing identification.

7 Acknowledgments

This paper and the research behind it would not have been possible without the exceptional support of my supervisor, Dr. Besnik Fetahu. Furthermore, we thank the anonymous reviewers for their valuable feedback.

References

Toril Aalberg, Jesper Strömbäck, and Claes H. de Vreese. 2012. The framing of politics as

strategy and game: A review of concepts, operationalizations and key findings. *Journalism: Theory, Practice & Criticism*, 13(2):162–178.

Maia Alavidze. 2017. The use of pronouns in political discourse. *International Journal of Arts & Sciences*, 9(4):349–356.

Alberto Ardèvol-Abreu. 2015. Framing theory in communication research. origins, development and current situation in spain. *Revista Latina de Comunicación Social*, (70).

Guus Bartholomé, Sophie Lecheler, and Claes de Vreese. 2018. Towards a typology of conflict frames. *Journalism Studies*, 19(12):1689–1711.

Eric Baumer, Elisha Elovic, Ying Qin, Francesca Polletta, and Geri Gay. 2015. Testing and comparing computational approaches for identifying the language of framing in political news. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1472–1482, Denver, Colorado. Association for Computational Linguistics.

Monika Bednarek. 2010. Evaluation in the news: A methodological framework for analysing evaluative language in journalism. *Australian Journal of Communication*, 37(2):15–50.

Douglas Biber and Edward Finegan. 1989. Styles of stance in english: Lexical and grammatical marking of evidentiality and affect. *Text - Interdisciplinary Journal for the Study of Discourse*, 9(1):93–124.

Nicolette Ruth Bramley. 2011. *Pronouns of politics: the use of pronouns in the construction of 'self' and 'other' in political interviews.* Phd thesis, the Australian National University.

Britta C. Brugman, Christian Burgers, and Gerard J. Steen. 2017. Recategorizing political frames: a systematic review of metaphorical framing in experiments on political communication. *Annals of the International Communication Association*, 41(2):181–197.

Joseph N. Cappella and Kathleen Hall Jamieson. 1996. News frames, political cynicism, and media cynicism. *The Annals of the American Academy of Political and Social Science*, 546:71–84.

Dallas Card, Amber E. Boydstun, Justin H. Gross, Philip Resnik, and Noah A. Smith. 2015. The media frames corpus: Annotations of frames across issues. 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 438–444, Beijing, China. Association for Computational Linguistics.

Dallas Card, Justin Gross, Amber Boydstun, and Noah A. Smith. 2016. Analyzing framing through the casts of characters in the news. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1410–1420.

Eunsol Choi, Chenhao Tan, Lillian Lee, Cristian Danescu-Niculescu-Mizil, and Jennifer Spindel. 2012. Hedge detection as a lens on framing in the gmo debates: A position paper. *arXiv preprint arXiv:1206.1066*.

Jinho D. Choi and Martha Palmer. 2012. Fast and robust part-of-speech tagging using dynamic model selection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 363–367, Jeju Island, Korea. Association for Computational Linguistics.

Dennis Chong and James N. Druckman. 2007. Framing theory. *Annual Review of Political Science*, 10(1):103–126.

Marina Dekavalla. 2018. Issue and game frames in the news: Frame-building factors in television coverage of the 2014 scottish independence referendum. *Journalism*, 19(11):1588–1607.

Caitlin M. Fausey and Teenie Matlock. 2011. Can grammar win elections? *Political Psychology*, 32(4):563–574.

Michael Halliday. 2004. *An Introduction to Functional Grammar*, 3 edition. Edward Arnold, London.

Susan Hunston. 2011. *Corpus approaches to evaluation: Phraseology and evaluative language*, volume 13 of *Routledge advances in corpus linguistics*. Routledge, New York, NY.

Susan Hunston and John Sinclair. 2003. A local grammar of evaluation. In Susan Hunston and Geoff Thompson, editors, *Evaluation in text*, Oxford linguistics, pages 74–101. Oxford University Press, Oxford.

Verena Jahnen. 2019. Die sprache der afd und wie sie sich verändert. In Eva Walther and Simon D. Isemann, editors, *Die AfD – psychologisch betrachtet*, pages 121–138. Springer Fachmedien Wiesbaden, Wiesbaden.

Kathleen Hall Jamieson. 1996. *Packaging the presidency: A history and criticism of presidential campaign advertising*. Oxford University Press.

Kristen Johnson, Di Jin, and Dan Goldwasser. 2017. Modeling of political discourse framing on twitter. In *Proceedings of the Eleventh International AAAI Conference on Web and Social Media*, pages 556–559.

DP Kingma and Ba J Adam. 2017. A method for stochastic optimization. cornell university library. *arXiv preprint arXiv:1412.6980*.

William Labov. 1972. *Language in the inner city*. University of Pennsylvania Press, Philadelphia.

Regina G. Lawrence. 2000. Game-framing the issues: Tracking the strategy frame in public policy news. *Political Communication*, 17(2):93–114.

Teenie Matlock. 2012. Framing political messages with grammar and metaphor. *American Scientist*, 100(6):478–483.

Viet-An Nguyen, Jordan Boyd-Graber, Philip Resnik, and Kristina Miler. 2015. Tea party in the house: A hierarchical ideal point topic model and its application to republican legislators in the 112th congress. 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 1438–1448.

Reid Pryzant, Richard Diehl Martinez, Nathan Dass, Sadao Kurohashi, Dan Jurafsky, and Diyi Yang. 2020. Automatically neutralizing subjective bias in text. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(1):480–489.

Danuta Reah. 1998. *The language of newspapers*. Intertext. Routledge, London and New York.

Mike S. Schäfer and Saffron O'Neill. 2016. Frame analysis in climate change communication. In *The Oxford Research Encyclopedia, Climate Science*. Oxford University Press USA.

Desirée Schmuck, Raffael Heiss, Jörg Matthes, Sven Engesser, and Frank Esser. 2017. Antecedents of strategic game framing in political news coverage. *Journalism: Theory, Practice & Criticism*, 18(8):937–955.

Adam Shehata. 2014. Game frames, issue frames, and mobilization: Disentangling the effects of frame exposure and motivated news attention on political cynicism and engagement. *International Journal of Public Opinion Research*, 26(2):157–177.

Debbita Ai Lin Tan. 2019. Language and political psychology: Can grammar influence electability? *GEMA Online Journal of Language Studies*, 19(3):1–21.

Geoff Thompson and Susan Hunston. 2003. Evaluation: an introduction. In Susan Hunston and Geoff Thompson, editors, *Evaluation in text*, Oxford linguistics, pages 1–27. Oxford University Press, Oxford.

Oren Tsur, Dan Calacci, and David Lazer. 2015. A frame of mind: Using statistical models for detection of framing and agenda setting campaigns. 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 1629–1638, Beijing, China. Association for Computational Linguistics.

Jesse Vig. 2019. A multiscale visualization of attention in the transformer model. *arXiv preprint arXiv:1906.05714*.

Holger Voormann and Ulrike Gut. 2008. Agile corpus creation. *Corpus Linguistics and Linguistic Theory*, 4(2):235–251.

Claes H. Vreese de. 2005. News framing: Theory and typology. *Information Design Journal + Document Design*, 13(1):51–62.

Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, 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.

8 Supplemental Material

Potential indicators	Examples
Semantic frames	Competition, Finish competition, Judgment communication, Quarreling, Leadership, Judgment, Hostile encounter, Taking sides, Change position on a scale, Evidence, Degree, Success or failure, Position on a scale, Occupy rank, Beat opponent, Statement, Importance, Size, Risky situation, Likelihood, Locale, Sole instance, Attempt suasion, Emotion directed, Sufficiency, Social interaction evaluation, Similarity, Stimulus focus, Judgment of intensity, Attention, Removing, Frequency, Warning, Emotion, Hindering, Weapon, Attack, Affirm or deny, Give impression, Level of force resistance, Negation, Manipulation, Fairness evaluation, Quitting, Hedging, Causation
Adjectives	*Top, good, only, strong, likely, big, unlikely, personal, leading, significant, low, moderate, sharp, high, unfair, fair, angry, weak, critical, polarized, top-tier, tough, wrong, aggressive, first, bad, clear, risky, overwhelming, urgent, old, serious, huge, bitter, divided, divisive, dangerous, crucial, dramatic, safe, unable, striking, negative, viable, pointed, well positioned, potential, successful, responsible, unrealistic, ridiculous, vulnerable, rival, young, implicit, fierce, increasing, direct, great, disappointed, important, close, inevitable, evasive, concerned, steady, uneven, surprising, powerful, necessary, private, perceived, prominent, lousy, protracted, upset, real, small, remarkable, wary, memorable, sudden, right, terrible, robust, trusted, outrageous, ludicrous, sarcastic, volatile, shameful, possible, appropriate, electable, ambitious, different, impressive, inexperienced, complicated, large, central, deep, enormous, fraught, inaccurate, controversial, experienced, limited, harsh, charged, focused, depressing, followed, empty*
Adverbs	*Even, too, never, very, so, just, rather than, really, well, already, increasingly, ever, repeatedly, largely, almost, little, simply, consistently, effectively, privately, enough, quickly, rarely, likely, directly, clearly, extremely, always, seriously, pretty, still, especially, deeply, immediately, forcefully, successfully, hard, differently, highly, completely, barely, notably, hardly, only, unusually, particularly, poorly, narrowly*
Adjectives and adverbs in comparative and superlative degrees of comparison	
Conjunctions	*More … than, as … as, unless, less … than*
Determiners	*More, less, much, enough*

Nouns	*Race, rival, attack, leader, field, contest, victory, charge, strategy, opponent, pressure, fight, contender, contrast, argument, criticism, concern, threat, power, remark, strength, top, record, front-runner, challenge, abuse, tactic, opposition, lack, tension, lead, war, position, performance, stake, winner, success, hoax, force, evidence, fear, chance, standing, uncertainty, rebuke, lie, problem, scrutiny, shot, battle, allegation, conflict, target, point, risk, rating, stance, war chest, leadership, wrongdoing, resentment, run, battleground, clash, difference, crime, failure, anger, cover-up, impact, defeat, test, status, rise, obstruction, stumble, leverage, mistake, outsider, stain, denial, fraud, fire, anxiety, game, insult, complaint, disinformation, brat, dismissal, frustration, enemy, gap, attention, critique, defensive, defiance*
Prepositions	*Like, behind, versus, against, unlike*
Pronouns	*Own, both, more*
Verbs	*Win, fail, attack, argue, lose, defeat, beat, refuse, criticize, warn, lead, fight, call, run, deny, reject, accuse, avoid, force, dismiss, drop out, believe, challenge, push, pressure, appear, question, worry, expect, fear, view, rise, need, mock, face, note, ignore, defend, confront, demand, block, claim, underscore, struggle, oppose, decline, disagree, denounce, abuse, refer, raise, threaten, lash out, go after, lie, indicate, compare, take on, target, portray, remove, suffer, respond, see, highlight, leave, insist, damage, boast, silence, vie, play, press, surpass, trash, stake, undermine, seem, flood, loom, emphasize, battle, blame, dispute, counter, complain, play out, take aim, violate, spar, promise, trail, prevent, promote, vow, undercut, quit, steal, risk, narrow, gain, overlook, parry, erupt, perceive, frame, obstruct, object, erode, lack, increase, fade, misspeak, isolate, lament, demonstrate, anger, divide, defeat, backfire, characterize, describe, abandon, allege, chide, ask, compete, delay, concede, appeal, condemn, dim, consider, dominate, betray, contradict*
Verbs in past simple, present simple, present perfect, and present progressive	
Passive voice	
Discourse markers	Self-mention *(we, I, our, my, us, me)*, Attitude markers *(I think, I don't think, I know, it's important, it's impressive, I believe)*, Interrogative sentences, Boosters *(of course, clearly, actually, certainly, obviously, really, in fact, definitely)*, Imperative sentences, Hedges *(perhaps, probably)*, Engagement markers *(note, consider)*
Negation	*No longer, couldn't afford, didn't work hard*

Biased language	Hedges, boosters, attitude markers *(I agree, in fact, actually, perhaps, possibly)*, Comparative adjectives and adverbs *(More aggressive, the worst, the most brutal, more dangerous, more forcefully, the biggest)*, Epistemic modality expressed through modal verbs, some adverbs, and metaphorically *(can, could, have to, must, might, may, should, Certainly, clearly, likely, literally, It's possible, it's obvious, it's unlikely, I think, I believe)*, Comparators *(questions, imperatives) (Of course! But are they strong enough? So, can a woman beat Donald Trump?)*, Emphatics *(really, certainly, of course, simply)*, Expressions of negativity: morphological, grammatical, and lexical *(unwilling, unwelcome, unstable, unrealistic, inexperienced, irresponsible, hardly, never, fail, disdain, pugnacious, limited, mislead)*, Patterns beginning with it and there *(It is extremely unlikely, it's impressive, it's really unfortunate)*, Pseudo-clefts *(It is he who, who I'd love to vote for is)*, Itensifiers *(Absolutely, completely, particularly, really, dangerously, highly)*, Adverbs of degree *(Enough, little, so, too, very)*, Comparator adverbs *(just, only, at least)*

Table 2: An overview of potential linguistic indicators of the issue-game frame.

Formulating Automated Responses to Cognitive Distortions for CBT Interactions

Ignacio de Toledo Rodriguez, Giancarlo Salton, Robert Ross
School of Computing, Technological University Dublin, Ireland
ignacio.toledo.rodriguez@gmail.com
{giancarlo.salton, robert.ross}@tudublin.ie

Abstract

One of the key ideas of Cognitive Behavioural Therapy (CBT) is the ability to convert negative or distorted thoughts into more realistic alternatives. Although modern machine learning techniques can be successfully applied to a variety of Natural Language Processing tasks, including Cognitive Behavioural Therapy, the lack of a publicly available dataset makes supervised training difficult for tasks such as reforming distorted thoughts. In this research, we constructed a small CBT dataset via crowd-sourcing, and leveraged state of the art pre-trained architectures to transform cognitive distortions, producing text that is relevant and more positive than the original negative thoughts. In particular, the T5 transformer approach to multitask pre-training on a sequence-to-sequence framework, allows for higher flexibility when fine-tuning on the CBT dataset. Human evaluation of the automatically generated responses showcases results that are not far behind from the overall quality of the ground truth scores.

1 Introduction

Recent studies (GDBC, 2018) estimate that approximately 300 million people globally suffer from depression, anxiety and other mental disorders. Cognitive Behavioural Therapy (CBT) is one of the leading practices across the field of psychotherapy (David et al., 2018) and one of the most effective ways of treating mental disorders such as anxiety or depression (Hofmann et al., 2012). CBT focuses on guiding the patients through a series of steps for identifying, analysing and correcting any cognitive distortions that may contribute to their mental health issues.

Traditional in-person CBT techniques applied in counselling sessions can be prohibitive for a large portion of the population due to cost, scarcity of therapists, convenience, stigma or other social considerations. However, in recent years there has been an increase in CBT material delivered online via computers and smartphone applications. In addition, a comprehensive review of these methods shows they can have many of the benefits of face-to-face therapy (Barak et al., 2008; Andersson and Cuijpers, 2009).

Automated agents that can deliver effective treatments represent a clear next step of research for online CBT. However, one of the main challenges here is the lack of publicly available datasets that can be used for training the necessary models. In light of these challenges, this research builds on the idea of a crowd-sourced corpus to generate CBT agent development by focusing on one of the foundational ideas of a CBT exercise, namely, the rewriting of distorted thoughts. Using this dataset, we then develop sequence-to-sequence (seq2seq) models to derive agents that can at least begin to address this central thought-rewriting challenge. While this is only an individual element of a complete CBT agent, it can be seen as a vital step in the study and analysis of the typical properties of CBT. In summary, the main contributions of this study are twofold:

- The creation of a Cognitive Behavioural Therapy dataset [1] that contains key information needed to train automated agents in producing CBT-related content, contributing to the development of Natural Language Processing (NLP) research in this domain.
- The use of modern machine learning techniques that demonstrate the effectiveness of leveraging a small CBT dataset to train a model to transform distorted negative thoughts into more realistic alternatives.

[1] https://github.com/itoledorodriguez/cbt-dataset

Cognitive Distortion	Description
All-or-Nothing Thinking	You see things in black and white categories. If your performance falls short of perfect, you see yourself as a total failure.
Overgeneralization	You see a single negative event as a never-ending pattern of defeat.
Mental Filter	You pick out a single negative detail and dwell on it exclusively so that your vision of reality becomes darkened.
Disqualifying the Positive	You reject positive experiences by insisting "they don't count" for some reason or other.
Jump to Conclusions - Mind Reading	You arbitrarily conclude that someone is reacting negatively to you, and you don't bother to check this out.
Jump to Conclusions - Fortune Teller Error	You anticipate that things will turn out badly, and you feel convinced that your prediction is an already established fact.
Magnification (Catastrophizing) or Minimization	You exaggerate the importance of things (such as your goof-up or someone else's achievements), or you inappropriately shrink things until they appear tiny (your own desirable qualities or the other fellow's imperfection).
Emotional Reasoning	You assume that your negative emotions necessarily reflect the way things really are: "I feel it, therefore it must be true".
Should Statements	You try to motivate yourself with shoulds and shouldnt's, as if you had to be whipped and punished before you could be expected to do anything. The emotional consequence is guilt. When you direct should statements toward others, you feel anger, frustration, and resentment.
Labelling and Mislabelling	This is an extreme form of overgeneralization. Instead of describing your error, you attach a negative label to yourself (eg: "I'm a loser"). When someone else's behaviour rubs you the wrong way, you attach a negative label to him (eg: "He's a goddam louse"). Mislabelling involves describing an event with language that is highly coloured and emotionally loaded.
Personalization	You see yourself as the cause of some negative external event which in fact you were not primarily responsible for.

Table 1: Definitions of the Cognitive Distortions used in this research. Taken from "Feeling Good: The new Mood Therapy" by Burns, D. 1981

2 Related Work

In the field of task-oriented Dialogue Systems, the technology has vastly improved since the introduction of ELIZA (Weizenbaum, 1966). Modern architectures such as Google Duplex (Leviathan and Matias, 2018) can handle complex goal-oriented conversations without human guidance, and novel approaches to frameworks such as Wizard-of-Oz (Wen et al., 2017) allows for the creation of crowd-sourced human datasets that can be used to train end-to-end agents towards a realistic conversation flow for different scenarios.

In the CBT domain, the highly rated and free of charge Woebot application is helping users around the world to identify and challenge cognitive distortions (Fitzpatrick et al., 2017). It combines template-based rules and modern machine learning techniques to deliver results but it does not, as of the time of writing, fully allow for the flexibility of a natural conversation.

The advancement in the last decade of machine learning, and in particular deep learning techniques for NLP, has made possible the development of automated models that excel at specific language tasks by being trained end-to-end over many iterations of large datasets, without the need for pre-established rules or templates. These techniques build on the seq2seq (Sutskever et al., 2014) and encoder-decoder architectures (Cho et al., 2014) to produce results in tasks such as machine translation, text summarization or sentiment analysis. In particular, the use of attention-based architectures (Bahdanau et al., 2015) that expand on the Transformer model (Vaswani et al., 2017) are widely used in the current state of the art models for NLP tasks.

Transfer Learning, a technique that was originally applied to the fine-tuning of computer vision tasks, has been a recent focus of NLP research, especially since ULMFit (Howard and Ruder, 2018) demonstrated how the weights of a LSTM language model pre-trained on a large dataset could be fine-tuned on a smaller corpus, for both language modelling and additional NLP tasks of the target dataset. Since then, other pre-trained models mostly based on the transformer architecture such as Elmo (Peters et al., 2018), GPT-2 (Radford et al., 2019) or BERT (Devlin et al., 2019), have been producing better results in diverse text generation and classification tasks.

When considering the rewriting of distorted or negative thoughts, this exercise can be compared to a seq2seq style transfer task where the situation or context remains the same, but the negative thoughts passed as inputs to the model are converted into more positive outputs. Shen et al. (2017) successfully demonstrate the effectiveness of style transfer in non-parallel data by mapping the inputs to a style-independent content representation.

3 Key Ideas in Cognitive Behavioural Therapy

A basic CBT interaction outlines a structure where the patients attempts to examine their own thoughts

Situation	Emotions	Negative Thoughts	Cognitive Distortions	Rational Response	Outcome
I had an important meeting that didn't go very well	Anxious 70% Sad 80%	I made a fool out of myself	Labelling Mind-Reading	It's true it wasn't my best meeting, but it's a big leap to label myself a fool just because I had a bad day. Also, you can't know what the rest of the people were thinking. Even if some thought that, they'd probably forget soon enough or do you remember all of the meetings conducted by your colleagues that didn't go that well?	Anxious 30% Sad 40%

Table 2: Daily Record of Dysfunctional Thoughts (Beck, 1979)

in terms of what they perceive to be a negative event, identifying any cognitive distortions and rephrasing them. In that process, the key steps are:

- Recognizing the situation that provoked the patient into experiencing a negative emotion and the intensity of those feelings.
- Writing down the automatic thoughts that accompany such emotions.
- Identifying any negative distortions that may be present in those thoughts (Table 1 shows the list of distortions considered in this study).
- Rewriting each distorted thought, aiming for a more rational or realistic alternative.
- Evaluating the patient feelings after the CBT exercise.

The patients with more experience in CBT techniques will be able to follow these steps by themselves in what is known as a CBT diary, also represented in Table 2. This is an exercise that allows them to immediately and effectively reduce their anxiety levels. However, and especially at the beginning of therapy, it is not always possible for the patients to come up with realistic alternatives that help combat their negative emotions. For that reason, a therapist can assist on guiding the patients through the main steps in the form of a conversation with a clear objective: i.e., reducing their anxiety levels.

When building a CBT dialogue corpus, much of the data needed is publicly available in forums, books or other online content – at least in raw format. It is relatively simple to identify in public forums negative situations where people express both their feelings and the distorted thoughts that accompany them. However, in this online content, there is one piece of information usually missing: the alternative, rational thought that will help patients to combat their negative feelings. This is a key part in CBT exercises, and it is at the same time the more difficult element to source when examining public data.

4 Data Collection

As part of integrating a CBT system within a modern machine learning dialogue framework, the previous section established the key idea of being able to transform irrational or distorted cognitive patterns into more realistic thoughts that are able to alleviate the negative emotions felt by the patients.

Hence, the main focus of our data collection has been the gathering of a series of negative thoughts that are objectively distorted and the use of crowdsourcing resources to obtain realistic counter arguments. More precisely, we build a dataset that contains multiple key value pairs for a single interaction, such as situation, emotions, negative thoughts, and rational response to those thoughts. All this data except the rational response is first prepared and then provided to the users that participate in the study.

As a first step in data preparation, a series of situations, feelings and negative thoughts were collected from a variety of sources such as CBT books, forums and public content aggregators. For those examples where the cognitive distortions contained within the negative thoughts are not mentioned explicitly, those distortions have been annotated manually. Note that the purpose of this research is not the cognitive distortion classification, but rather the rewriting of negative thoughts. The cognitive distortions just provide additional context for the survey users and help them to come up with a realistic counter argument.

The survey respondents were asked to read carefully the instructions and to provide, in their own words, a realistic alternative to the negative thoughts in each of the situations presented. For

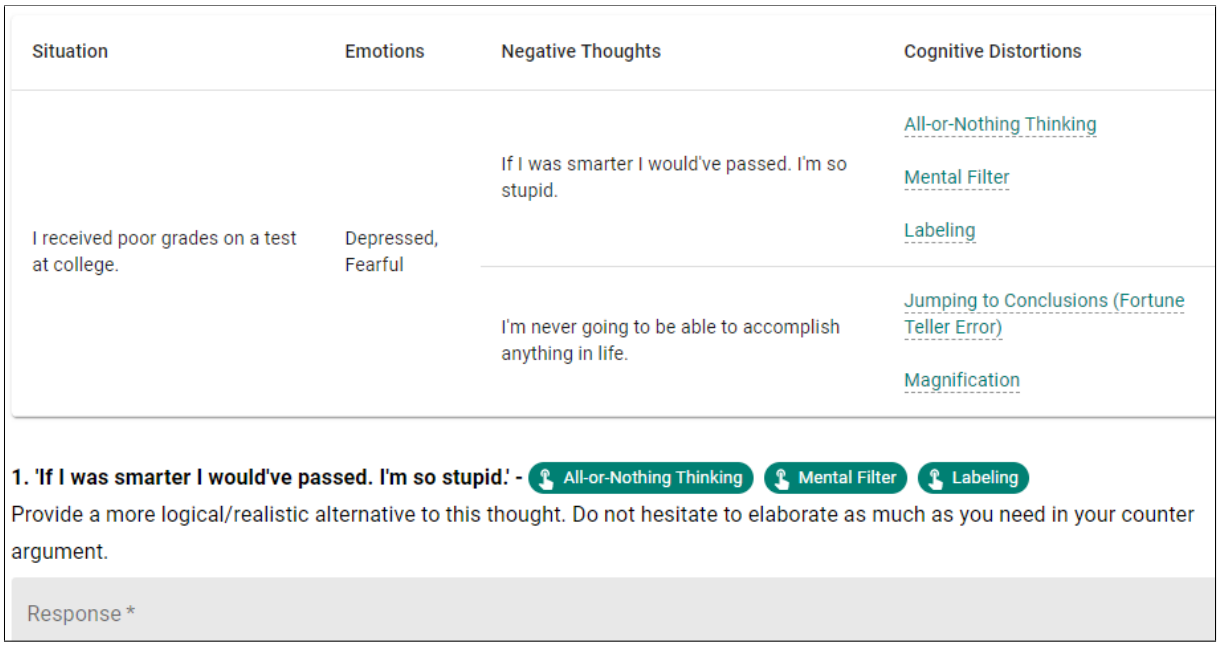


Figure 1: Example situation that contains negative distortions. Participants in the survey will write a more realistic counter-argument to each automatic thought.

	Counts
Situations	108
Type Count (%)	
Work	26.85
Romantic	22.22
Social	12.04
Friends	11.11
Family	10.19
Health	7.41
School and College	5.56
Other	2.78
Bereavement	0.92
Addiction	0.92
Negative Thoughts	200
Participants	114
Responses	442

Table 3: Number of responses gathered during the survey for the situations and negative thoughts that were prepared beforehand. Note that, for some situations, there have been multiple responses collected.

this study, the crowd-sourcing platform of choice has been Prolific [2], linked to a custom website (Fig 1) that loads two random situations for every participant, with an average of two negative thoughts per situation. Table 3 showcases the different situation types and the number of responses collected.

[2]https://www.prolific.co/

5 CBT Response Generation

While creating a new dataset is essential to our goals, the primary objective is to explore the use of modern deep learning architectures to automatically formulate appropriate responses against negative thoughts that can help to counter anxiety and depression. Overall, to do this, a number of seq2seq models that have produced good results in other NLP tasks are examined in this research.

5.1 Modelling Strategies

As a modelling strategy, we concatenate the situation description with each negative thought, forming a single sequence that serves as the input to the models, in a supervised learning approach. The target texts are those responses written by humans as per the crowdsourcing task from last section.

Due to the small dataset collected, and in order to produce significant results when trying to transform distorted thoughts into more realistic alternatives, the use of transfer learning and pre-trained language models is necessary. The responses generated with a model solely trained on the CBT dataset, regardless of the architecture used, do not achieve good results from the point of view of basic literacy or semantic coherence.

However, some of the pre-trained models used during the research, such as a simple transformer architecture, are not nuanced enough to allow for the small CBT dataset to significantly influence the

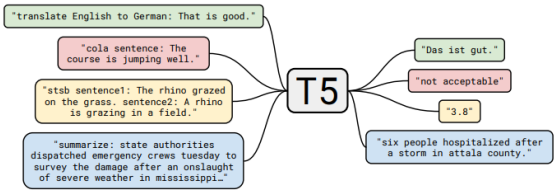


Figure 2: Google T5 Model Overview. The input text contains prefix keywords that allows for parallel training in different downstream tasks.

results produced in a seq2seq cognitive rephrasing task. In order to achieve an effective transfer learning strategy when fine-tuning for the response generation task, this study leverages one of the current state of the art architectures, namely the T5 transformer.

5.2 T5 Transformer

Recently, Google published the T5 text-to-text transformer model (Raffel et al., 2020), trained on a cleaned version of the Common Crawl corpus (C4). The published model checkpoints have been pre-trained in a diverse variety of unsupervised and supervised tasks (language modelling, word embedding, machine translation, text summarization, etc), which allows for the flexibility of fine-tuning smaller datasets in any NLP downstream tasks. During training and evaluation, the model is able to recognize prefix tokens which are added to the input text, in order to distinguish between the different tasks (Figure 2).

The architecture of the T5 transformer is very similar to that of the original transformer, keeping a stack of encoder and decoder blocks, each composed of self-attention layers and feed-forward networks. There are some modifications around layer normalization and position embedding, but what makes the pre-trained models excel at transfer learning is the multi-task approach to pre-training when applied to their large C4 dataset, and scaling up the number of parameters of the model.

While the largest model checkpoint - T5-11B, with 11 billion parameters, is able to exceed previous benchmarks in tasks such as GLUE or SQAD, for the purposes of this research and, due to processing constraints, the T5-Large model with 770 million parameters has been leveraged when fine-tuning the CBT task.

5.3 Baseline

We also make use of a non-trivial baseline model consisting of a seq2seq framework where both the encoder and decoder are composed of a BERT base architecture (Devlin et al., 2019). As one of the first transformer models fully pre-trained on a plain text corpus, mostly on the English Wikipedia and the BooksCorpus (Zhu et al., 2015), we choose BERT to contrast and showcase the advancement, in a relatively short span of time, of these type of pre-trained architectures when fine-tuning on a smaller dataset.

6 Experiments

To evaluate the quality of our automatically constructed responses we make use of both quantitative metrics and survey-based human evaluation, comparing aspects such as fluency, positive sentiment and overall quality of the text produced.

6.1 Metrics

As discussed in previous sections, rewriting negative thoughts into a more positive or realistic version can be considered a style transfer task. With this in mind, we have considered metrics that have been commonly applied to the style transfer objective. (Yang et al., 2018).

Here we have specifically made use of **Perplexity**, **BLEU** (Papineni et al., 2002) and **METEOR** (Banerjee and Lavie, 2005). These metrics are also commonly applied across a range of other NLP tasks such as machine translation and image captioning. Each of them can be thought of as providing an assessment of how much our predicted text seems to match that of the original labels.

Sentiment Analysis. We also make use of a pre-trained sentiment classifier model which we fine-tune for the CBT dataset. The classifier determines whether the alternative responses produced by the text-to-text transformers are considered positive or negative, obtaining an average accuracy for each of the models. The model we chose for this sentiment analysis has been RoBERTa (Reimers and Gurevych, 2019), pre-trained on the Yelp dataset reviews and fine-tuned on the CBT dataset, where all of the inputs or negative thoughts are considered as negative and all of the targets as positive.

The evaluation set has been used to compute the automatic metrics score. Except for perplexity, the experiments generate thirty different responses for each of the inputs in the target, averaging the

	Automatic Metrics				Human Evaluation		
	Perplexity	BLEU	METEOR	Sentiment	Rel.	Sentiment	Quality
Google T5	**18.21**	**0.016**	**0.094**	70.29%	**3.74**	**3.77**	**3.60**
BERT Seq2Seq	71.09	0.012	0.077	**90.72%**	2.74	3.30	2.56
Human	-	-	-	-	4.01	4.18	4.05

Table 4: Evaluation results for the CBT dataset, for both automatic metrics and human assessment. For reference, the table also includes the human evaluation of the dataset labels.

results to obtain the BLEU, METEOR and sentiment scores. The model also uses nucleus sampling (Holtzman et al., 2019) with a top-p value of 0.95, to allow for diversity in the responses.

6.2 Human Evaluation

In order to subjectively evaluate the responses generated by the models under study, and to contrast them against the original human labels, a number of surveys have been sent to users of the Prolific crowd-sourcing platform. The participants are restricted to those that have English as their first language.

The surveys were divided in three groups of 20 participants each. One group evaluated the original human targets from the dataset, while the other two groups received the responses from the T5 and BERT models. The survey in each group contains the same five different situations picked from the evaluation set, along with their initial negative thoughts, and the only difference is the generated responses.

The methodology followed for choosing the generated responses included in the survey was to produce three different responses for each of the situations, and pick the subjective best one from those. Appendix A includes all three responses generated for each situation by the T5 model.

The questions asked in the survey attempt to evaluate the generated responses in terms of relevance, positive sentiment and, finally, semantic quality and coherence of the text. The participants are asked to rank each of these metrics from 1 to 5, lowest to highest score.

6.3 Configurations

All experiments have been run using the Simpletransformers library [3], which leverages the more popular HuggingFace's Transformers repository [4] allowing for a fast setup and a fine-tuning of many pre-trained transformer architectures.

This research uses the T5 large model for the tests, comprising a total of 770 million parameters and 24 layers for both the encoder and decoder, along with a 16-head attention mechanism. During fine-tuning and evaluation, the max sequence length has been restricted to 64 tokens.

The baseline seq2seq model uses a BERT uncased pre-trained model with 110 million parameters, 12 layers and a 12-headed attention, for both the encoder and the decoder. The quantitative and qualitative results are better than those produced by the larger BERT model with 336 million parameters; we believe that this is likely due to the small size of the CBT dataset.

6.4 Results

Table 4 summarizes the results obtained for both the automatic and human evaluations, which also includes the results of the original dataset targets, for contrast.

The BLEU and METEOR scores are very low due to the large probability space when generating responses and the use of p-sampling to obtain more diverse and fluent results. This, coupled with the small size of the dataset in comparison with the pre-training corpus, affects the score by producing text which diverges substantially from the original labels.

The automated sentiment analysis by the fine-tuned RoBERTa model shows a higher positive sentiment for the baseline BERT model, but this doesn't reflect the subjective quality of the text produced by both models which is subjectively much better for the T5 architecture, as seen in the human evaluation results.

When the automatically generated responses are judged by participants in the survey, the BERT model falls behind significantly, specially in terms of relationship to the situation and overall quality. In fact, the scoring of the T5 in these two metrics is much closer to the original human written responses, showcasing BERT's inability to directly address the situation, often producing text with low

[3] https://github.com/ThilinaRajapakse/simpletransformers
[4] https://github.com/huggingface/transformers

semantic coherence.

7 Next Steps

One of the limitations in this research is the small size of the dataset, with just about a hundred different situations, so the obvious course of action would be to continue expanding on them by gathering new situations via crowd-sourcing. With a bigger corpus of data, along with other architectural improvements and transfer learning mechanisms, the results obtained in this study can be improved significantly.

Ultimately though, the aim of the research is to incorporate all of the key value pairs of the dataset - such as situations, emotions, negative thoughts, cognitive distortions and alternative responses - into a full fledged dialogue framework with the clear task of guiding patients through all of the steps of a CBT interaction.

8 Conclusion

This research focuses on the key CBT idea of transforming negative thoughts that contain cognitive distortions into more realistic alternatives, in order to provide automatic and therapeutic assistance to patients experiencing anxiety and depression.

Although there have been previous research within the NLP and CBT domains, especially in distortion and emotion classification (Rojas-Barahona et al., 2018), this study appears to be the first that manufactures a full CBT dataset, and attempts to apply modern machine learning architectures to automatically convert an initial negative thought into a more positive or realistic alternative.

Existing crowd-sourcing platforms represent a practical way for collecting human responses against distorted or negative thoughts and, in the future, they may also prove to be an effective source for gathering new situations.

The results obtained show how effective transfer learning can be when using state of the art transformers architectures to fine-tune a small CBT dataset. In particular, and specially when considering human evaluation, the Google T5 transformer model produces quality responses that are more realistic, while still being relevant to the situations and thoughts causing anxiety.

Future work will focus on expanding the existing CBT dataset, while trying to incorporate it into a more complete dialogue system framework.

References

Gerhard Andersson and Pim Cuijpers. 2009. Internet-based and other computerized psychological treatments for adult depression: A meta-analysis. *Cognitive behaviour therapy*, 38:196–205.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

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, Ann Arbor, Michigan. Association for Computational Linguistics.

Azy Barak, Liat Hen, Meyran Boniel-Nissim, and Na'ama Shapira. 2008. A comprehensive review and a meta-analysis of the effectiveness of internet-based psychotherapeutic interventions. *Journal of Technology in Human Services*, 26:109–160.

A.T. Beck. 1979. *Cognitive Therapy of Depression*. Guilford clinical psychology and psychotherapy series. Guilford Publications.

David D. Burns. 1981. *Feeling Good: The New Mood Therapy*. Penguin Books.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Daniel David, Ioana Cristea, and Stefan Hofmann. 2018. Why cognitive behavioral therapy is the current gold standard of psychotherapy. *Frontiers in Psychiatry*, 9:4.

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.

Kathleen Kara Fitzpatrick, Alison Darcy, and Molly 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. *JMIR Mental Health*, 4:e19.

GDBC. 2018. Global burden of disease study 2017.

Stefan Hofmann, Anu Asnaani, Imke Vonk, Alice Sawyer, and Angela Fang. 2012. The efficacy of cognitive behavioral therapy: A review of meta-analyses. *Cognitive therapy and research*, 36:427–440.

Ari Holtzman, Jan Buys, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *CoRR*, abs/1904.09751.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Yaniv Leviathan and Yossi Matias. 2018. Google duplex: An ai system for accomplishing real-world tasks over the phone.

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.

Matthew 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.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners.

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.

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.

Lina M. Rojas-Barahona, Bo-Hsiang Tseng, Yinpei Dai, Clare Mansfield, Osman Ramadan, Stefan Ultes, Michael Crawford, and Milica Gašić. 2018. Deep learning for language understanding of mental health concepts derived from cognitive behavioural therapy. In *Proceedings of the Ninth International Workshop on Health Text Mining and Information Analysis*, pages 44–54, Brussels, Belgium. Association for Computational Linguistics.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

J. Weizenbaum. 1966. Eliza - a computer program for the study of natural language communication between man and machine. *Commun. ACM*, 9:36–45.

Tsung-Hsien Wen, David Vandyke, Nikola Mrkšić, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. 2017. A network-based end-to-end trainable task-oriented dialogue system. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 438–449, Valencia, Spain. Association for Computational Linguistics.

Zichao Yang, Zhiting Hu, Chris Dyer, Eric P Xing, and Taylor Berg-Kirkpatrick. 2018. Unsupervised text style transfer using language models as discriminators. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Y. Zhu, Ryan Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 19–27.

A Appendix: Generated Responses by T5 during Evaluation

The following tables showcase the different situations presented in the human evaluation survey. For each situation, three alternative responses are generated automatically by the T5 model. The responses in bold were hand-picked to be included in the survey.

Situation	You are a psychiatrist at the University of Pennsylvania. You are attempting to revise your manuscript on depression after meeting with your editor in New York. Although your editor seemed extremely enthusiastic, you notice you are feeling nervous and inadequate due to your thoughts.
Negative Thought	"They made a terrible mistake when they chose my book! I won't be able to do a good job."
T5 Alternative Responses	**1. "I am trying to revise my book to make it better. You have to find the time to relax and not feel nervous. The better the rest of the week the better the results will be."** 2. "You have made a horrible mistake and you must find another way of expressing your opinion. You can not make all mistakes at the same time and you must be willing to improve your writing." 3. "No matter how much you understand about your client's personality, it is important that you get the answers to the questions before your editor."

Situation	You are about to give a lecture and you notice that your heart is pounding.
Negative Thought	"My mind will blank out. I'll make a fool of myself."
T5 Alternative Responses	**1. "You will do your best to not make a fool of yourself. I will have the confidence in myself to do what I need to."** 2. "No one wants to make a fool of themselves, just because the thought in their head is negative – they won't realize how many people out there really believe in them." 3. "You are not able to think through that and will let you make mistakes. So I will not be so focused to speak, if I recite the lesson I can forget."

Situation	Starting to have a panic attack on the subway.
Negative Thought	"What if I pass out on the subway?"
T5 Alternative Responses	**1. "You will not feel like you will pass out on the subway, and it is not likely to occur to you. You will never know what happens."** 2. "You can't run out of oxygen in the subway." 3. "There are tons of things that can go wrong when you're panicking on the subway, and you don't have to worry about the fact that you are going to pass out in public."

Situation	Friend walks past you in the street, without acknowledging you.
Negative Thought	"I'm a terrible person that isn't funny nor charismatic. I don't have any real friends."
T5 Alternative Responses	**1. "People should be interested in you and you aren't looking for friendship. Friendship is about being yourself e.g. making new friends is important to gaining new relationships and it is also important that friendship develops over time so you continue to grow, develop, and improve."** 2. "I should just tell someone to make friends for themselves." 3. "This is not my personal issue I have no friends and that does not change the fact of it. It is not a private matter of me deciding to make friends, or to be friends only, but to be friends with one another to share your life experiences and thoughts if they choose to."

Situation	My friend has left several messages for me about going to lunch.
Negative Thought	"I am so lazy and weak that I can't do a simple thing like calling my friend."
T5 Alternative Responses	**1. "I should call back again, it's because I am so lazy and weak that I can't do simple things. I will try something but I will make it easy for my friend to call and see if I can go."** 2. "I can contact him even when the friend can't reach me." 3. "It is not easy and I won't know if this will happen to her. Maybe the way we communicate, when it happened with a friend, isn't enough to fix the problem."

The Quality of Lexical Semantic Resources: A Survey

Hadi Khalilia
University of Trento / Italy
hadi.khalilia@unitn.it

Abed Alhakim Freihat
University of Trento / Italy
abdel.fraihat@gmail.com

Fausto Giunchiglia
University of Trento / Italy
fausto@disi.unitn.it

Abstract

With the increase of the lexical-semantic resources built over time, lexicon content quality has gained significant attention from Natural Language Processing experts such as lexicographers and linguists. Estimating lexicon quality components like synset lemmas, synset gloss, or synset relations are challenging research problems for Natural Language Processing. Several lexicon content quality approaches have been proposed over years in order to enhance the work of many applications such as machine translation, information retrieval, word sense disambiguation, data integration, and others. In this research, a survey for evaluation the quality of lexical semantic resources is presented.

1 Introduction

Lexical Semantic Resources (LSRs) are lexical databases that organize the relations between their elements (synsets) via lexical and semantic relations. The basic element of LSR is a synset. A synset is a set of lemmas (dictionary form of a word), gloss (natural language text that describes of the synset), and synset examples. A lemma within a synset has a meaning which we call a meaning (sense). Synset examples are used to understand the meaning of a lemma in a synset. For example, the following is a synset:

#1 person, individual, someone, somebody, mortal, soul: a human being; "there are too much for one person to do".

"Person, individual, someone, somebody, mortal and soul" are lemmas of the synset, "a human being" is the gloss, and "there are too much for one person to do" is the synset example (Miller et al., 1990). Lexical relations organize the relationships between senses. For example the antonym lexical relation expresses that two senses are opposite in meaning such as *love is antonym of hate*. Semantic relations organize the relationships between synsets". For example the synset (a) is a hypernym (is-a) of (b) (Miller et al., 1990; Chandrasekaran and Mago, 2021).

(a) chicken, Gallus gallus: a domestic fowl bred for flesh or eggs; believed to have been developed from the red jungle fowl.

(b) domestic fowl, fowl, poultry: a domesticated gallinaceous bird thought to be descended from the red jungle fowl.

The quality of synset components and also, the quality of its lexical semantic relations are the main factors that influence its quality that participate in increasing or deceasing a LRS quality. Therefore, synset quality measurement is important in order to evaluate the quality of LRSs.

Building LRSs such as WordNet faces many challenges. These challenges are polysemy, missing lemmas, missing senses, and missing relations. For example, one of the main problems that makes Princeton WordNet (Miller and Fellbaum, 2007; Freihat, 2014) difficult to use in natural language processing (NLP) is its highpolysemous nature due

to too many cases of redundancy, too fine grained senses, and sense enumerations. On the other hand, it has several synsets that have missing lemmas and missing relations with other synsets. Also, some lemmas in WordNet have missing senses.

In order to solve these challenges, researchers have proposed three categories of approaches, which are: the category of synset lemmas evaluation approaches, the category of synset gloss evaluation approaches, and the category of synset relations evaluation approaches.

In this survey, these categories are described by tracking the evaluation of synset quality approaches over the past years. Also, the survey focuses on recent researches that have not been covered in the previous surveys.

The paper is organized as follows. In Section 2, we discuss the lexicon quality challenges. In Section 3, we describe the current approaches for synset-quality evaluation. In Section 4, we conclude the paper and discuss future research work.

2 Lexicon Quality Challenges

Lexical-semantic resources face several challenges, they are categorized into two main categories: OVERLOAD work or UNDERLOAD. Inappropriate senses, inappropriate lemmas and inappropriate connections between synsets are needed some extra works and produce OVERLOAD components in LSRs. On the other hand, missing senses, missing lemmas and missing connections between synsets produce UNDERLOAD problem. Therefore, in the following sections we present some of the challenges that produce lexicon with low quality.

2.1 Polysemy

LSR, e.g, WordNet organizes the relation between terms and synsets through senses (term-synset pair). A term may have many meanings (one or more senses) which is called polysemous term. For example, **head** has 33 senses in WordNet which indicates that there are 33 relations between the word head and associated synsets. The ambiguity of a term that can be used (in different contexts) to express two or more different meanings is called polysemy. Due to synonymy and polysemy, the relation between terms and synsets is many-to-many relationship. Really, wrong semantic connection can be occurred in WordNet. A misconstruction that results in wrong assignment of a synset to a term is called *Sense enumeration* (Freihat et al., 2015).

In WordNet, a compound-noun which contains two-parts (modifier and modified) causes polysemy this is called *compound-noun polysemy*. It corresponds to "the polysemy cases, in which the modified noun or the modifier is synonymous to its corresponding noun compound and belongs to more than one synset". WordNet contains a substantial amount of this type of ploysemy such as: *center* and *medical center* in WordNet (Kim and Baldwin, 2013).

Also in WordNet, a special case is founded when there are related some senses (synsets) with a specific polysemous term and not connected with it. For example, a hierarchical relation between the meanings of a polysemous term (Freihat et al., 2013b). "In case of abstract meanings, we say that a *meaning A* is a more general meaning of a *meaning B*. We say also that the *meaning B* is a more specific meaning of the *meaning A*" which is called *specialization polysemy*. In this case, synset connections require reorganizing the semantic structure (using semantic relations) to cover and reflect the (**implicit**) hierarchical relation between all such senses.

So, the big challenge in WordNet is polysemy, because it may produce OVERLOAD connections (overload of a number of term-synset pairs). For example wrong assignments of a synset to terms in sense enumeration add overload relations in WordNet which decrease the synset quality implicitly.

2.2 Missing Senses, Lemmas and Relations

Despite "the highpolysemous nature of wordNet, there are substantial amount of *missing senses* (term-synset pairs) in WordNet" based on Ciaramita and Johnson's work that cause UNDERLOAD of term synsets problem which is the opposite of the overload of term synsets. For example, new added words in languages cause missing senses (synsets) for some terms in lexical resources (e.g, WordNet). Such as *Crypto Mining* sense is missing from the synsets of *mining* term in WordNet and only two synsets are founded in WordNet for it (Ciaramita and Johnson, 2003).

Also, WordNet contains synsets with *missing lemmas* as shown in (Verdezoto and Vieu, 2011). For example, "the term *brocket* denotes two synsets in WordNet, the lemmas of the two synsets are incomplete. This is due to the following: the terms *red brocket* and *Mazama americana* which are syn-

onyms of the lemmas in (b) are missing. The two synsets do not even include the term *brocket deer*. *(a) brocket: small South American deer with unbranched antlers. (b) brocket: male red deer in its second year"*

WordNet relations are "useful to organize the relations between the synsets, while substantial amount of relationships between the synsets remain **implicit** or sometimes **missing** as in the case synset glosses relations. For example, the relation between *correctness* and *conformity* is *implicit*. The relation between *fact or truth* and *social expectations* in the following two meanings of the term correctness is *missing*. A human being may understand that *correctness* is a hyponym of *conformity* and *fact or truth* is a hyponym of *social expectations*, but this is extremely difficult or impossible for a machine because *conformity* is neither the hypernym of (a) nor (b). The relation between *fact or truth* and *social expectations* is *missing* because *social expectations is not defined* in WordNet which makes the two synsets are **incorrect** (Freihat et al., 2013a).

Missing senses, missing terms or missing Relations may cause UNDERLOAD problem whether UNDERLOAD in connections or UNDERLOAD in synset itself. Therfore, to enhance synset quality, you have to solve the two main problems: OVERLOAD and UNDERLOAD which are caused by **polysemy** and **missing**, respectively.

3 Quality Evaluation Methods

Lexicon quality estimation methods evaluate the quality of semantic network that a lexical-semantic resource should have. This work depends on the calculation of the synset (acts as a node in the semantic network) correctness and completeness, and also, depends on the connectivity degree of the synset with other synsets in the semantic network. In this section, we introduce and discuss the methods of the lexicon quality evaluation which contains both manual and automatic evaluation methods for the synset quality dimensions (synset correctness, synset completeness and its connectivity) and further classify the evaluation methods into three categories, including synset terms/lemmas evaluation approaches, synset gloss analytical methods, and synset relations with other synsets measures.

3.1 Synset Lemmas Evaluation Methods

Based on the underlying principle of how the synset lemmas are assessed, synset lemmas evaluation methods can be further categorized as Lemmas Validation Methods, and Lemmas Clustering Methods.

3.1.1 Lemmas Validation Methods

The most famous method for lemmas validation is the work of Ramanand in (Nadig et al., 2008). They presented **Validate Synset** algorithm, its principle depends on "**dictionary definitions** to verify that the words present in a synset are indeed synonymous or NOT". This is due to the availability of synsets in which some members "do not belong". To accomplish their work they discussed the following research questions: "is a given WordNet complete, how to select one lexico-semantic network over another, and are WordNet synsets INCOMPLETE (may be many words have been **omitted** from the synset) and are WordNet synsets CORRECT (the words in a synset indeed synonyms of each other and the combination of words should indicate the required sense)". To answer the questions they try to validate the available synsets which are the foundations of a WordNet. "A WordNet synset is constructed by putting together a set of synonyms that together define a particular sense uniquely. This sense is indicated for human readability by a gloss". To evaluate the **quality of a synset**, they begin by looking for validating the synonyms that the synset has them. They follow these subtasks in the synset validation: are the words in a synset indeed synonyms of each other? Are there any words which have been omitted from the synset? And does the combination of words indicate the required sense? In their work, they focus on the quality of content embedded in the synsets; this is by attempting to verify a given a set of words/lemmas if they were synonyms and thus correctly belong to that synset or not synonyms based on the following **two principles**: "*if two words are synonyms, it is necessary that they must share one common meaning out of all the meanings they could possess. And a condition could be showing that the words replace each other in a context without loss of meaning*" (Nadig et al., 2008).

A simple block diagram for a synset synonym validation using the system is shown in Figure 1. As we notice from the block diagram, the **input** to the system is: "a WordNet synset which provides the following information: the synonymous words

in the synset, the hypernym(s) of the synset and other linked nodes, gloss, example usages". The **output** consists of "a verdict on each word as to whether it fits in the synset, i.e. whether it qualifies to be the synonym of other words in the synset, and hence, whether it expresses the sense represented by the synset". They used the following **hypothesis**: "*if a word is present in a synset, there is a dictionary definition for it which refers to its hypernym or to its synonyms from the synset*" (Nadig et al., 2008).

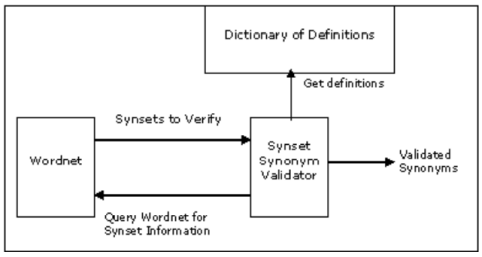


Figure 1: Block Diagram for Synset Synonym Validation.

However, dictionary definitions include useful clues for validating and verifying synonymy. The results show that: the algorithm is simple to implement and depends on the nature (the depth and the quality) of the used dictionary. Many words in WordNet are not validated, around 0.18 of total words in WordNet and 0.09 of total WordNet synsets that couldn't be validated. Also, the algorithm cannot detect omissions from a synset. To overcome this shortcoming of the algorithm, they proposed that expanding the validation to the synset gloss, and synset relations; using more dictionaries in validation; running the algorithm to other language WordNets, and applying the algorithm on other parts of speech in English. The same team proposed in (Ramanand and Bhattacharyya, 2007) an automatic method for **the synset synonyms and the hypernyms validation** based on new rules: 8 rules in synonym validation and 3 rules for hypernyms validation which is the first attempt of automatic evaluations for synsets in WordNet. They focus on the *synsets* because they are the foundational elements of wordnets and focus on the *hypernymy* hierarchy this is due to its importance in semantic linkages with other synsets. The quality of the synset and its hypernymy ensure the correctness, the completeness and the usability of the resource. They evaluate the quality of a wordnet by "examining the validity of its constituent synonyms and its hypernym-hyponym pairs". The authors defined the synonymy validation as "*the inspection of the words in the synset indeed synonyms of each other or NOT*", and they use the following observation: "If a word **w** is present in a synset along with other words w_1, w_2, ..., w_k, then there is a dictionary definition of w which refers to one or more of w_1, w_2, ..., w_k and/or to the words in the hypernymy of the synset" which was the hypothesis in the (Nadig et al., 2008) work. In the **synonymy validation algorithm**, the authors apply 8 rules in order which are the basic steps of the algorithm. Also, omissions from synsets aren't considered. Examples of these are synsets such as: *Taylor, Zachary Taylor, President Taylor*: no definition for the last multiword. Thus the multiword synonyms do share partial words. To validate such multi-words without dictionary entries, they check for the presence of partial words in their synonyms". They run the algorithm on the *noun synsets* (39840 from the available 81426) of PWN, the inputs of the algorithm are synsets with more than one lemma, by running the validator which uses the online dictionary service *Dictionary.com* in validation, the results show that the percentage of the synsets where all words were validated is (0.701), Pushpak algorithm is simple and acts as a backbone for the synset validation models, also, the applied rules such as: Rule1, Rule2 and Rule7 are the most impact among synonym validation rules, on the other hand Rule4, Rule5 and Rule6 are the lowest. They conclude that many of the words present in PWN aren't validated and those with rare meanings and usages. "The wordnet contains synsets that have outlier words and/or missing words". The limiting factors are "the availability of dictionaries and tools like stemmers for those languages". They plan to summarize the quality of the synsets into a single number. The results could then be correlated with *human evaluation*, finally converging to a score that captures the human view of the wordnet. "The presented algorithm is available only for Princeton WordNet. However, the approach could broadly apply to other language wordnets and other knowledge bases as well. And the algorithm has been executed on noun synsets; they can also be run on synsets from other parts of speech". Also, in the same area and due to the wide-spread usage of lexical semantic resources, the lexicon quality evaluation became more and more important to tell us how well the applica-

tions and operations based on these resources perform, for example, the authors in (Giunchiglia et al., 2017) describe a general approach to improve the quality of the lexical semantic resources by proposing an algorithm to classify the ambiguity words (based on their senses) in the lexical semantic resources to three classifications for a: polyseme, homonym or unclassified. Also, they present "a set of formal quantitative measures of resource incompleteness". And apply their work and analysis on "a large scale resource, called the Universal Knowledge Core (UKC)". The authors define "two types of incompleteness, i.e., *language incompleteness* and *concept incompleteness*". Language Incompleteness (in a lexical resource): a set of synsets/words/concepts is not lexicalized in a lexical resource (e.g UKC) by a specific language. A model (language incompleteness measurement) that can be used to measure the count (how much) of omitted synsets/words/concepts in the language is described in (Giunchiglia et al., 2017).
The notion of "concept incompleteness can be thought of as the dual of language incompleteness. If the language incompleteness measures how much of the UKC a language does not cover, the concept incompleteness measures how much a single concept is covered across a selected set of languages. Concept incompleteness: is the complement to 1 of its coverage". A concept incompleteness model that can be used to measure the concept incompleteness is described in (Giunchiglia et al., 2017). Also in the same research, **lexical ambiguity** is described (it is happened when one word in a language denotes to more than one concept) and they computed the number of ambiguity instances in UKC, e.g., polysemy or homonymy. As an application example they applied the proposed algorithm to "checks whether any two concepts denoted by a single word are polysemes of homonyms or NOT on the UKC concepts". They run the algorithm which consists of 4 steps, and the results showed that, "the UKC contains 2,802,811 ambiguity instances across its pool of 335 languages, these instances were automatically evaluated by the algorithm which, generated 0.32 polysemes among all the ambiguity instances and 0.22 homonyms across all languages". They concluded that when the language coverage increases then the average ambiguity coverage decreases, and vice versa. Also, "increasing the minimal required number of ambiguity instances consistently increases the percentage of polysemes (up to the 0.74), decreases the percentage of homonyms (down to the 0.11) as well as the percentage of unclassified instances (down to around the 0.15)". Giunchiglia's group presented the language incorrectness evaluation method in UKC in (Giunchiglia et al., 2018), the authors proposed that "the languages in the UKC are far from being complete, i.e., from containing all the words and synsets used in the everyday spoken or written interactions. And far from being correct, i.e., from containing only correct senses, namely, only correct associations from words and concepts to synsets". These limiting factors impact the lexical resource quality. **Language Incorrectness** is the number of psycholinguistic mistakes in a language in a lexical resource per the number of total of concepts in that language in the same resource. They proposed a model to measure the language Incorrectness in (Giunchiglia et al., 2018). Furthermore, this work solves the problem of synset incomplete through presenting a model that transforms the semantic relations nodes from synsets to concepts. This is based the fact that is some words have multiple meanings, and each word is codified as a synset, consisting of a (possibly incomplete) set of synonymous words. The proposed approach describes the UKC design as three-layers: words, synsets and concepts. "Word layer, stores what we call the universal lexicon, the synset layer, stores the world languages, and the concept layer, stores the world (mental) model(s), as represented by the CC". This work makes an improvement in the UKC that influences on its quality; this due the work that becomes a language independent and handles the problem of each synset is associated with one and only one language.

3.1.2 Lemmas Clustering Methods

Lemmas clustering methods retrieve and make clusters for collected synonym lemmas from lexical semantic resources and dictionaries based on the lexical semantic network, for example, the authors in (Lam et al., 2014) proposed an approach that built a new Wordnet from several lexical resources and Wordnets using machine translation (MT- is the core operation for their approach). The presented algorithms use three approaches to generate (translate synsets) synset candidates for each synset in a target language T, the approaches as follow. "*1) the direct translation (DR) approach:* this approach directly translates synsets in PWN to T. *2) Approach using intermediate Wordnets (IW):* for each synset,

they extract its corresponding synsets from intermediate Wordnets. Then, the extracted synsets, which are in different languages, are translated to T using MT to generate synset candidates. Synset candidates are evaluated using IW method as shown in Figure 2, also the ranking of synset candidates depends on the ranking equation in (Lam et al., 2014). *3) Approach using intermediate Wordnets and a dictionary (IWND):* in this approach, one bilingual dictionary is added to the intermediate languages to T. They translate synsets extracted from intermediate Wordnets to English, then translate them to the target language using English-Target language dictionary. For each sysnset, they have many translation candidates. A translation candidate with a higher rank is more likely to become word belonging to the corresponding synset of the new Wordnet in the target language". To improve the quality of the Wordnet synsets; feedback and comments from communities (mother-tongue) can be used.

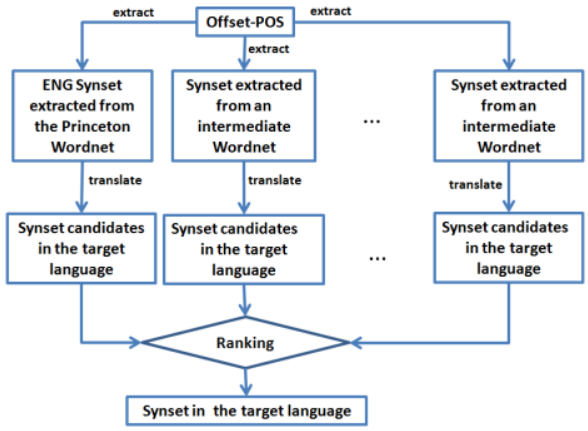


Figure 2: The Intermediate Wordnets Method for Synset Ranking.

Various synset construction (lemmas clustering) methods were proposed combining the **commutative** methods. (Fierdaus et al., 2020) proposed a novel approach for Automatic Indonesian WordNet Development (automatic synset creation). In previous researches, manual methods were used in the establishment of synsets such as the cluster ring approach. The approach in (Fierdaus et al., 2020) creates synsets from the Indonesian Thesaurus. The input to the system is a set of words which is taken from Thesaurus Bahasa Indonesia (*the used thesaurus is in pdf format which was published in 2008.*), and then the system works on the set of words (input) to find out the semantic similarities between words through successive steps. The initial input is a word, after that the commutative method is used for the **synset extraction**. Then through the **pre-processing stage**, the system removes excessive characters in the synset that produced from previous stage. After that, the synset that produced from pre-processing will be clustered and combined using **Agglomerative Hierachical** Clustering algorithm. All these steps are applied for the **Automatic** WordNet Development. Also, the resulted synsets are evaluated using the F-measure method which involves the calculation of precision (P) recall (R) and the evaluation using the gold standard as in (Fierdaus et al., 2020). Commutative method focuses on a commutative relation between the synonyms/lemmas. If a commutative relation is available between the lemmas then the synset will be Valid. Synonym relations should be commutative which means that "*if a word* k_1 *has a synonym* k_2*, then* k_2 *also must be a synonym of* k_1". Finding a synset that has a valid value is done using a matrix table. Synset extraction is carried out in several steps of the algorithm as follows (Ananda et al., 2018): searching for a sense of the entry word, searching for synonyms on every sense from the previous step, searching for "the chosen word" in the sense that being sought, identify the prospective synset to be sought: by looking for candidates for the synset that can be generated from each item from the words in the dataset, determine whether every word in the prospective synset has a commutative relationship, elimination of candidate synset which is a subset of the other synset and take the remainder of the elimination synset candidate. **Clustering process** is important for making the extracted synset better. They applied Agglomerative Hierarchical Clustering. It is a bottom-up approach that grouped data based on distance value and "the clustering process will be stopped after it reached a condition decided by threshold value". And in (Fierdaus et al., 2020), the authors selected 80 words (as the **test** data) that taken randomly from the thesaurus. And then the system processed the selected word by the system and will produce one or more synset using Agglomerative Hierarchical Clustering method. And also, the authors used the **gold standard** "it finds out how much the correlation between the score issued by the system and the relevance of the words being tested" which is the result of validating synonym sets performed by lexical experts (lexicographers). However, in the validation process F-Measure value for the research

approach was 0.84. It is expected to apply and measure the performance of Agglomerate Hierarchical Clustering and another clustering method on a bigger data scale in the development of synset for Indonesian WordNet. Finally, another synset construction method is **fuzzy synsets** extraction, where fuzzy synsets are a special type of synsets which is discovered from textual definitions. For example, the authors in (Oliveira and Gomes, 2011) present a fuzzy synsets extraction method based on "the term senses are not discrete" fact. Fuzzy synsets are extracted from three (Portuguese) dictionaries **automatically**, which are: Dicionario Aberto and the Portuguese Wiktionary as public domain dictionaries and PAPEL 2.0 as a public domain lexical network. They proposed the following steps in the approach: they specify general textual patterns for extracting synonymy-pairs. Compute the similarity value between terms in synonymy-pairs. Using the similarity value, the clusters are created and they are called fuzzy clusters (fuzzy synsets). Then they build a graph using these fuzzy clusters. Lastly, the built graph participates in creating a fuzzy thesaurus in Portuguese language. The method of creating of fuzzy synsets is based on two stages: First, use a dictionary to extract synonymy graph (where they revealed that the number of collected synsets from Portuguese dictionaries in order to create Portuguese WordNet was larger than the number of synsets in Portuguese thesauri which are discovered manually), and then use the created graph to cluster the words/terms in synsets. Synpairs (two nouns are connected as synonyms) can be extracted from the definitions in dictionaries. Also, they use the clustering-algorithmm which is shown in (Oliveira and Gomes, 2011) to make fuzzy synsets clusters which are discovered from the synonymy graph [G = (N, E), where —N— are the number nodes in G and —E— are the number of edges in E]. The main steps of the algorithm are: 1) Empty sparse matrix creation. 2) Fill the cells of the Empty sparse matrix with the similarity ratio between of the words in the adjacency vectors. 3) Normalize the cell values in the sparse matrix. 4) extract fuzzy clusters. 5) If two clusters have the same elements then they will be merged in a bigger cluster. The *input* of the algorithm is synonymy graph G and the outputs are the resulting synsets in Portuguese. The authors used the manual evaluation for the created synsets (Padawik thesaurus). The average of corrected synonyms pairs in Padawik is 0.75 and for the synsets is higher than 0.73. In order to get more improvement, they will focus on each fuzzy synset by specifying individual cut-points, and also to work on new relations between words not only the similarity (synonymous words).

3.2 Synset Gloss Evaluation Methods

Measuring lexical semantic relatedness for a synset or a concept generally requires certain background information about the synset. Such information is often described in the synset **gloss**, which includes a different number of examples. The authors in (Zhang et al., 2011) introduced a new model to measure the semantic relatedness. The model exploits the WordNet gloss and semantic relations as features in building concept vectors. Also, they use other features in the designed model: "**wn-synant** merges WordNet synonyms and antonyms. **wn-hypoer** merges WordNet hypernyms and hyponyms, and **wn-assc** merges WordNet meronyms, holonyms and related, which are features corresponding to associative relations". This work participates in the improvement of the quality of WordNet and Wikipedia operations.

Hayashi and his team used a gloss as an indicator in semantic relatedness is their work in (Hayashi, 2016) paper which measures the strength of the evocation relation between the lexicalized concepts. The authors in (Hayashi, 2016) defined the evocation as "a directed yet weighted semantic relationship between lexicalized concepts". Evocation relations are "potentially useful in several semantic NLP tasks, such as the measurement of textual similarity/relatedness and the lexical chaining in discourse, the prediction of the evocation relation between a pair of concepts remains more difficult than measuring conventional similarities (synonymy, as well as hyponymy/hypernymy) or relatednesses (including antonymy, meronymy/holonymy)" as in (Cramer, 2008). The work in (Hayashi, 2016) made good improvements on evocation relations by applying a novel approach in to prediction of the strength and direction of the evocation relations. For example, PWN dataset includes (39,309) synset pairs. If we compare the work of Y. Hayashi with the results of (Ma, 2013), Y. Hayashi considered "evocation as a semantic relationship between lexicalized concepts, rather than a relation between words", which were considered in (Ma, 2013). Also, the authors in (Maziarz and Rudnicka, 2020) worked on the possibility of the WordNet construc-

tion based on "a distance measure which performs better than other knowledge-based features in evocation relations" (Hayashi, 2016). They used the Dijkstra's algorithm to "measure the distance between nodes (words/synsets) in WordNet structure using a new method for evocation strength recognition based with four types of relations: "**wn**: pure WordNet relations (directed WordNet edges), **g**: gloss relations (directed gloss relation instances), **polyWN**: the set of all pairs of polysemous lemma senses taken from WordNet (bidirectional relations between different senses of the same polysemous lemma) and **polySC**: the set of all pairs of polysemous lemma senses co-occurring in SemCor corpus" as described in (Chklovski and Mihalcea, 2002). "Dijkstra's distance measuring algorithm was applied on the four structures (one structure for each relation type) to get the minimum points between lexical concept pairs. Then 3-similarity measures are used in each time in order to obtain the best predictions of evocation strength in all cases" (Maziarz and Rudnicka, 2020). Marek Maziarz and his team presented a novel approach for *evocation relation measurement which based on the combination of three types of relations: "gloss relations, pairs of polysemous lemma senses and instances derived from the SemCor corpus, and using the proposed inverse Dijkstra's distance* for improving lexical WordNet structure for the needs of evocation recognition". Like the categorization of methods in the preceding subsection, the next group of methods that we present attempt to explain the importance of the synset gloss in the synset quality evaluation by incorporating an additional examples in the gloss.

3.2.1 Synset Gloss Properties

In this section, we discuss the features and properties of the synset gloss, and how the instructions in (Jarrar, 2006) coverage the gloss properties. In addition, we will define some of the problems that can be solved with help of the synset gloss. Synset gloss writing has several rules and instructions; each synset developer has to apply them during a synset creation for example 6 instructions are explained in the paper of (Jarrar, 2006). In this study, the notion of gloss for ontology engineering purposes and the significance of glosses have been introduced. Gloss is "a useful mechanism for understanding concepts individually without needing to browse and reason on the position of concepts". For example, the work in (Jarrar, 2006) introduced the notion of gloss for concepts/terms in lexical resources by suggesting a list of instructions for writing a gloss. These **instructions** are the following:
1. "It should start with the principal/super type of the concept being defined. For example, Search engine: A computer program that ..., University: An institution of ...".
2. "It should be written in the form of propositions, offering the reader inferential knowledge that helps him to construct the image of the concept". For example, instead of defining Search engine as "A computer program for searching the internet" one can say "A computer program that enables users to search and retrieve documents or data from a database or from a computer network..."
3. "It should focus on distinguishing characteristics and intrinsic properties that differentiate the concept from other concepts (it is the most important)".
4. "The use of supportive examples is strongly encouraged".
5. "A gloss should not contradict the formal axioms" and vice versa.
6. "It should be sufficient, clear, and easy to understand".

The supportive examples in glosses are important due to "clarify true cases (commonly known as false), or false cases (commonly known as true); and to illustrate and strengthen distinguishing properties". They will implement the proposed algorithm in lexical resources like WordNet. And they plan to investigate "how much the process of validating glosses can be (semi-) automated". The synset gloss is the explanation of the sysnset that can cause correct or wrong assignment in synset-term pairs. Sense enumeration in WordNet is "one of the main reasons that results in wrong assigning of a synset to a term". The authors in (Freihat et al., 2015) proposed a novel approach to "discover and solve the problem of sense enumerations in compound noun polysemy in WordNet". Compound noun polysemy in WordNet is classified into three types such as: "metonymy polysemy cases where the modified noun belongs to two synsets, one of these synsets is base meaning and the other is derived meaning. The specialization polysemy cases where the modified noun belongs to two synsets, one of these synsets is a more general meaning of the other or both synsets are more specific meanings of a third synset. And Sense enumeration

means a misconstruction that results in wrong assignment of a synset to a term, i.e., assignment the noun modifier or the modified noun as a synonym of the compound noun itself". They reduced "the number of sense enumerations in WordNet without affecting its efficiency as a lexical resource". This research improves the lexicon quality by **removing irrelevant semantic relations** between synsets (Freihat et al., 2015).

3.2.2 Synset Gloss Validation

Synset gloss validation methods are computationally simple but they need much effort when someone works on the gloss validation manually, and the synset resources as lexical databases and dictionaries act as a strong backbone for the gloss extraction models, for example, Purnama and his group presented a supervised learning based approach which is an automatic gloss extractor for Indonesian synsets (Purnama et al., 2015). The main sources and datasets used are web documents containing the gloss of the synsets. The proposed approach includes three main phases which are: **preprocessing**, **features extraction**, and **classification phase**.

Preprocessing phase includes several sub tasks such as they fetch a collection of web documents using a search engine, raw text extraction and clean-up, and they extract sentences from gloss candidates. Also, in *the features extraction* phase, seven features are extracted for each gloss: "the number of characters in a sentence, the number of words in a sentence, the position of a sentence in a paragraph, the frequency of a sentence in the document collection, the number of important words in a sentence, the number of nouns in the sentence and the number of gloss sentences from the same word". In the final phase- *Classification*, the supervised learning approach depends on these features to accept or reject the candidate which is a gloss in a test. In the classification operation, they used two models which are: Backpropagation feedforward neural networks (BPFFNN) and decision tree DT models. BPFFNN is a multilayer architecture with seven input nodes; these nodes represent the extracted features (attributes) that extracted in the second phase. But the output node is used for deciding which one of two classes (ACCEPT or REJECT) through the gloss prediction operation. The nodes are shown in the BPFFNN architecture in Figure 3. On the induction using DT, they consider all of the features as continuous value with positive integer and the internal branch nodes in DT are binary splits, –each node has a label (value) $\leq$n and $>$n.

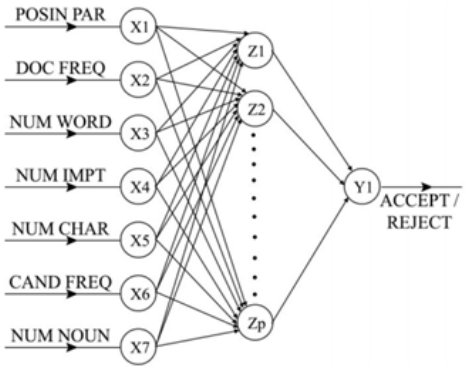


Figure 3: BPFFNN architecture for Gloss Candidate Classification.

In this research, the system was successful in collecting 6,520 Indonesian synset glosses, and the accuracy of using the decision tree and BPFFNN is then calculated, and the accuracy average is 0.74 and 0.75, respectively. This work represents an improvement in a gloss sentence candidate validation. The authors recommend applying the method of the acquisition of gloss natural languages in the world other than Indonesian.

3.3 Synset Relations Evaluation Methods

In this section, we discuss two types of synset relations: Implicit relations and Special relations. In addition, we will present the sub-types of relations, several examples and how they can coverage the synset-connectivity with other synsets.

3.3.1 Implicit Relations

The methods of synset relatedness based on implicit relations were emerging to measure the lexicon quality such as the work of Bhattacharyya and his team in the research (Nadig et al., 2008). The authors proposed an approach for **hypernymy validation**, this approach receives two synsets as input and states whether they have a hypernym-hyponym relationship between them or NOT". Also, this work is the first attempt for hypernymy validation. The approach consists of three steps (Nadig et al., 2008): *First step*: "prefix forms as an indicator of hypernymy: this is using the following rule: If one term of a synset X is a proper suffix of a term in a synset Y, X is a hypernym of Y" *Second step*: "using web search to validate hypernymy: this is using the following hypothesis: If two words in the form of a Hearst pattern show a sufficient number

of search results on querying, the words can be validated as coming from a hypernym-hyponym synset pair". *Third step*: "using coordinate terms to validate hypernymy: this is using the following hypothesis: If two terms are established to be coordinate terms, a hypernym of one can be stated to be the hypernym of the other". The hypernymy validation was tested on "the set of all direct hypernyms for noun synsets in the PWN". There are a total of 79297 hypernym-hyponym pairs constituting this set. A synset is validated if it gives non-zero search (using Microsoft Live search) results for any 2 of the 9-patterns-Hearst patterns (Hearst, 1992) tested in the algorithm. And "the utilization of coordinate terms is achieved by using Wikipedia as corpus". In all, the authors were able to validate 0.71 of noun hypernymy relation pairs in the Princeton WordNet using their algorithm. The authors concluded in (Nadig et al., 2008) that "many of the synsets present in PWN contain semantic relations may be **inappropriately** set up or may be **missing** altogether".

Also, another example, in (Freihat et al., 2013a) worked on the extraction of the explicit relations from implicit ones in order to enhance WordNet. They added "new explicit hierarchical and associative relations between the synsets which reorganized the semantic structure of the polysemous terms in wordNet". The authors transform "the **implicit** relations between the polysemous terms at **lexical** level to **explicit** relations at the **semantic** level between synsets". However, their approach deals with all polysemy types at all ontological levels of WordNet such as Metonymy, Specialization polysemy, Metaphors and Homographs polysemy. They identified the relations: is-homograph,has-aspect and is-metaphor as extracted semantic relations between sysnsets. In addition, specific relations for a specialization polysemy are extracted. The explicit relations at the semantic level are: "*Homographs*: there is no relation between the senses of a homograph term. They use the relation is-homograph to denote that two synsets of a polysemous term are homographs. For example, this relation holds between the synsets saki asalcoholic drink and saki as a monkey. *Metonymy*: in metonymy cases, there is always a base meaning of the term and other derived meanings that express different aspects of the base meaning. For example, the term chicken has the base meaning a domestic fowl bred for flesh or eggs and a derived meaning the flesh of a chicken used for food. To denote the relation between the senses of a metonymy term, they use the relation has-aspect, where this relation holds between the base meaning of a term and the derived meanings of that term. *Metaphors*: in metaphoric cases, they use the relation is-metaphor to denote the metaphoric relation between the metaphoric meaning and literal meaning of a metaphoric term. For example, this relation is used to denote that cool as great coolness and composure under strain is metaphoric meaning of the literal meaning cool as the quality of being at a refreshingly low temperature". Also, in some cases (e.g, in *Specialization polysemy*), the authors suggested to add a new (missing) parent; they established a new (missing) is-a relation and affix a number of synsets to one synset. This work improved the WordNet quality by "transforming the implicit relations between the polysemous senses at lexical level into explicit semantic relations", and they used the manual evaluation to measure the quality of the approach. The approach was applied on all polysemous nouns. So, they recommended applying the algorithm to handle "verbs, adjectives and adverbs".

Finally, a new research in implicit relations is the paper of T.Dimitrova and her group (Dimitrova and Stefanova, 2019). They have added semantic relations between *nouns* in WordNet that are *indirectly* linked via verbs and adjectives. This assigns new semantic properties to nouns in WordNet. The work reveals **hidden** (indirect) semantic relations between nouns (a noun–noun pair) by using information that is already available from the inter-POS derivative and (morpho) semantic relations between noun – verb, and noun – adjective synsets. "Most relations between synsets connect words of the same part-of-speech (POS), such as : noun synsets are linked via hypernymy / hyponymy (superordinate) relation, and meronymy (part-whole) relation, verb synsets are arranged into hierarchies via hypernymy / hyponymy relation, adjectives are organized in terms of antonymy and similarity, and relational adjectives (pertainyms) are linked to the nouns they are derived from, and adverbs are linked to each other via similarity and antonymy relations". But the authors work on the following *two main categories* for hidden semantic network extraction (Dimitrova and Stefanova, 2019):

1. *Noun – noun relations through verbs*: noun

synsets that are derivationally related to a verb synset and linked through semantic relations that are inherited from the (morpho)semantic relations between noun and verb synsets. The authors worked on 10 categories of the relations, as follow: Instrument Relation, Actor Relation, Causator Relation, Agent Relation, Theme Relation, Result Relation, Location Relation, Uses Relation, Property Relation, and Time Relation
2. *Noun – noun relations through adjectives*: both sides of the relations in this category are nouns and connected through adjectives. 4 types are selected for the category relations, as follow: Property Relation, Part-of Relation, Related Relationand Result Relation.

Dimitrova's work participated in the increasing of the SYNSET CORRECTNESS ratio. They identified the semantic relations between nouns in WordNet that are indirectly linked via derivative relations through verbs and adjectives. Also, the formulated relations will not only increase the interrelatedness and density of WordNet relations but would allow us to assign new semantic properties to nouns, these properties will explicitly assist the synset to interconnect with the appropriate synsets (senses) that also, improve the SYNSET CORRECTNESS (Dimitrova and Stefanova, 2019).

3.3.2 Special Relations

Special types of synset relations are discussed in this section, these relations added more semantic properties on the synset lattice in lexicons, such as the work of Hayashi that was discussed in Section 3.2, they proposed "a supervised learning approach to predict the strength (by regression) and to determine the directionality (by classification) of the evocation relation that might hold between a pair of lexicalized concepts PWN evocation dataset" (Hayashi, 2016). The authors used neural network (NN) model for classifying evocation relations into **FOUR** categories which are: "outbound", "inbound", "bidirectional" and "no-evocation". And the forest regression model for the prediction of evocation strength is presented. The features of evocation relations are: *Similarity/relatedness features*: 4 similarity/relatedness features are utilized; two of them are synset-based such as "wupSim computes Wu-Palmer similarity which gives the depth of node s from the root" whereas others are word-based such as "ldaSim feature provides the cosine similarity between the word vectors". *Lexical resource features*: these features have been captured some asymmetric aspects of evocation relationships such as lexNW that finds "the difference in graph-theoretic influence of the source/target concepts in the underlying PWN lexical-semantic network". And *Semantic relational vectors*: in this feature category, they depended on the rule of (Mikolov et al., 2013). "all pairs of words sharing a particular relation are related by the same constant (vector)" to implement the features of the evocation relation. This paper proposed "a supervised learning approach to predict the strength and to determine the directionality of the evocation relation between lexicalized concepts"; which directly impacts the synset connectivity through improving the strength and directionality measurements. The best case in their experiments was the combination of the proposed features "Similarity/relatedness features, Lexical resource features and Semantic relational vectors" which outperformed the individual baselines (Hayashi, 2016). In addition, the authors of the paper (Maziarz and Rudnicka, 2020) focused on a special type of evocation relation which is polysemy, in order to recognize evocation strength. Strong polysemy links participate in constructing a high-quality lexical resource. The framework consisted of three **steps**. *First*: they studied the topologies (3-topologies) of the network of polysemy (graphs of senses). All relations of these topologies are polysemy. Spearman's correlation is used for evaluating the similarity measure in the 3-topologies in order to choose the best topology for lexical resource construction. *Second*: the evocation strength is measured based on the selected topology in step 1 and using the Neural Network and Random Forest models. Also, the authors presented a novel approach which is based on Dijkstra's algorithm to calculate distances between lexical concepts in WordNet structure, and using three types of relations: "A complete polysemy graph (WN-g-co): for a given lemma- linked all senses together". SemCor-based polysemy graph (WN-g-sc): an incomplete graph built by extracting polysemy links from SemCor. It makes groups for such sense pairs that co-occur in the corpus, giving poor completeness but probably good precision". And "the chaining graph (WN-g-ch) tries to connect senses based on contemporary semantic relations between senses of all polysemous words/lemmas that are the closest as in the WordNet graph using the nearest-neighbor chaining algorithm". The **chaining topology** is the best one among the three

listed topologies for lexical resource construction. Therefore; the polysemy network of the lexical resource structure is constructed using "chaining procedure executed on individual word senses of polysemous lemmas". In the measuring of the evocation strength, the work (Maziarz and Rudnicka, 2020) used the inverse of Dijkstra's Distance. For each synset in the evocation set, they "calculated the $dist_{Dijkstra}$ measure and its inverse to find the evocation strength" using semantic relational vectors and the lexical resource features (Hayashi, 2016). *Third*, they applied Neural Network-NN and Random Forest-RF models to measure evocation strength on the chaining topology with the selected features. Good accuracy in the measurements of the evocation strength is resulted from applying both NN and RF models. Therefore, the authors recommended to utilizing the results in the applications of the polysemy such as Word Sense Disambiguation and Information Retrieval.

Conclusion

Three categories of approaches that influence synset quality in lexical semantic resources which are used in NLP applications were discussed. These are: synset lemmas evaluation category, synset gloss evaluation category, and synset relations evaluation category.

The challenges of synset quality were also discussed, these challenges might cause an OVERLOAD or UNDERLOAD the number of LSR components. They also negatively affect lexicon quality.

These approaches were related explicitly or implicitly with synset quality. Despite of each approach gave a good solution, it couldn't solve all problems/challenges of synset quality. They presented partial solutions that handled with one or two challenges at most. Each approach was a complement to each other as shown in Table 1. It shows a tabulation of these approaches according to synset quality dimensions that are influenced by the challenges. A comprehensive definition for synset quality and an approach that evaluated synset quality with all categories weren't studied in previous researches. An approach that combines all these partial solutions to reach a comprehensive evaluation of LSR quality is recommended.

Section	Completeness	Correctness	Connectivity
3.1.1	✓	✓	
3.1.2	✓	✓	
3.2.1		✓	✓
3.2.2		✓	
3.3.1			✓
3.3.2			✓

Table 1: The coverage of discussed methods

References

I Putu Prima Ananda, Moch Arif Bijaksana, and Ibnu Asror. 2018. Pembangunan synsets untuk wordnet bahasa indonesia dengan metode komutatif. *eProceedings of Engineering*, 5(3).

Dhivya Chandrasekaran and Vijay Mago. 2021. Evolution of semantic similarity—a survey. *ACM Computing Surveys (CSUR)*, 54(2):1–37.

Timothy Chklovski and Rada Mihalcea. 2002. Building a sense tagged corpus with open mind word expert. In *Proceedings of the ACL-02 workshop on Word sense disambiguation: recent successes and future directions*, pages 116–122.

Massimiliano Ciaramita and Mark Johnson. 2003. Supersense tagging of unknown nouns in wordnet. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pages 168–175.

Irene Cramer. 2008. How well do semantic relatedness measures perform? a meta-study. In *Semantics in Text Processing. STEP 2008 Conference Proceedings*, pages 59–70.

Tsvetana Dimitrova and Valentina Stefanova. 2019. On hidden semantic relations between nouns in wordnet. In *Proceedings of the 10th Global Wordnet Conference*, pages 54–63.

Valentino Rossi Fierdaus, Moch Arif Bijaksana, and Widi Astuti. 2020. Building synonym set for indonesian wordnet using commutative method and hierarchical clustering. *JURNAL MEDIA INFORMATIKA BUDIDARMA*, 4(3):778–784.

Abed Alhakim Freihat. 2014. *An organizational approach to the polysemy problem in wordnet*. Ph.D. thesis, University of Trento.

Abed Alhakim Freihat, Fausto Giunchiglia, and Biswanath Dutta. 2013a. Regular polysemy in wordnet and pattern based approach. *International Journal On Advances in Intelligent Systems*, 6.

ABED ALHAKIM Freihat, FAUSTO Giunchiglia, and BISWANATH Dutta. 2013b. Solving specialization polysemy in wordnet. *International Journal of Computational Linguistics and Applications*, 4(1):29.

Abed Alhkaim Freihat, Biswanath Dutta, and Fausto Giunchiglia. 2015. Compound noun polysemy and sense enumeration in wordnet. In *Proceedings of the 7th International Conference on Information, Process, and Knowledge Management (eKNOW)*, pages 166–171.

Fausto Giunchiglia, Khuyagbaatar Batsuren, and Gabor Bella. 2017. Understanding and exploiting language diversity. In *IJCAI*, pages 4009–4017.

Fausto Giunchiglia, Khuyagbaatar Batsuren, and Abed Alhakim Freihat. 2018. One world–seven thousand languages. In *Proceedings 19th International Conference on Computational Linguistics and Intelligent Text Processing, CiCling2018, 18-24 March 2018*.

Yoshihiko Hayashi. 2016. Predicting the evocation relation between lexicalized concepts. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1657–1668.

Marti A Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Coling 1992 volume 2: The 15th international conference on computational linguistics*.

Mustafa Jarrar. 2006. Position paper: towards the notion of gloss, and the adoption of linguistic resources in formal ontology engineering. In *Proceedings of the 15th international conference on World Wide Web*, pages 497–503.

Su Nam Kim and Timothy Baldwin. 2013. Word sense and semantic relations in noun compounds. *ACM Transactions on Speech and Language Processing (TSLP)*, 10(3):1–17.

Khang Nhut Lam, Feras Al Tarouti, and Jugal Kalita. 2014. Automatically constructing wordnet synsets. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 106–111.

Xiaojuan Ma. 2013. Evocation: analyzing and propagating a semantic link based on free word association. *Language resources and evaluation*, 47(3):819–837.

Marek Maziarz and Ewa Rudnicka. 2020. Expanding wordnet with gloss and polysemy links for evocation strength recognition. *Cognitive Studies— Études cognitives*, (20).

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

George A Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J Miller. 1990. Introduction to wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–244.

George A Miller and Christiane Fellbaum. 2007. Wordnet then and now. *Language Resources and Evaluation*, 41(2):209–214.

Raghuvar Nadig, J Ramanand, and Pushpak Bhattacharyya. 2008. Automatic evaluation of wordnet synonyms and hypernyms. In *Proceedings of ICON-2008: 6th International Conference on Natural Language Processing*, volume 831. Citeseer.

Hugo Gonçalo Oliveira and Paulo Gomes. 2011. Automatic discovery of fuzzy synsets from dictionary definitions. In *Twenty-Second International Joint Conference on Artificial Intelligence*.

I Purnama, Mochamad Hariadi, et al. 2015. Supervised learning indonesian gloss acquisition. *IAENG International Journal of Computer Science*, 42(4).

J Ramanand and Pushpak Bhattacharyya. 2007. Towards automatic evaluation of wordnet synsets. *GWC 2008*, page 360.

Nervo Verdezoto and Laure Vieu. 2011. Towards semi-automatic methods for improving wordnet. In *Proceedings of the Ninth International Conference on Computational Semantics (IWCS 2011)*.

Ziqi Zhang, Anna Lisa Gentile, and Fabio Ciravegna. 2011. Harnessing different knowledge sources to measure semantic relatedness under a uniform model. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 991–1002.

An interpretable person-job fitting approach based on classification and ranking

Mohamed Amine Menacer[1], Fatma Ben Hamda[2], Ghada Mighri[2]

Sabeur Ben Hamidene[2] and Maxime Cariou[1]
[1]AYMAX Consulting, Tour Black Pearl, 14 Rue du Général Audran, 92400 Courbevoie, France
[2]AYMAX Consulting, Immeuble Youssef Towers-Avenue de Dinars, 1053 Tunis, Tunisia
{mohamedamine.menacer,fatma.hamda,ghada.mighri
sabeur.ben.hmidene,maxime.cariou}@aymax.fr

Abstract

While the development of online recruitment platforms has allowed companies to post their job offers and for job seekers to submit their resumes simply and efficiently, the recruitment process remains time and resources consuming. Accordingly, models are needed to support the automatic matching between candidate resumes and job offers, which is called person-job fit issue. Recent works have focused on modeling the matching between resumes and job requirements through deep learning techniques. However, due to the complex internal transformations that will subject to the input, these models suffer from the interpretability problem. Yet, in real deployment, it is necessary to explain why a candidate is accepted or rejected for a given job offer. To this end, we propose a hybrid approach that takes benefit from deep learning techniques while making the matching results human-readable. This was achieved by extracting several features from the resume and job description and use them to perform classification and ranking. The obtained results on French resumes dataset show an accuracy of 93.7% in the case of classification and 70% of accepted resumes were ranked on the top 5 candidates, and this in the case where the problem is processed as a ranking issue.

1 Introduction

Over the last few years, the number of job posts and resumes submitted to online recruitment platforms is evolving and growing at a rapid rate. In 2020, more than 40M people used LinkedIn to search for their dream job each week, which leads to an average of 3 people hired every minute on the same platform[1]. While these online platforms make the process of jobs posting and resumes submitting easy, the resumes analysis and the candidates selection still time and energy consuming. For example, at AYMAX consulting, a recruiter needs more than 19 days to process unsolicited applications.

The candidate selection is a process that consists of dividing all applicants to a job offer into two categories: those who we want to interview and those who are not accepted. A good candidates' selection process is the one, which saves the time and helps to find the suitable candidate to the job requirements. It should be as fast as possible because spending too much time looking for the best profile can drive talent away and we need therefore to start the process all over again. This process is based on a good balance between time optimization and quality. This could be achieved by designing effective models that perform job-resume matching automatically, which is called person-job fit issue.

Several approaches were proposed to deal with the person-job fit issue and this from several perspectives: the issue can be modeled as job recommendation issue (Patel and Vishwakarma, 2020), skills measuring (Gugnani and Misra, 2020), candidates matching (Jiang et al., 2020) and candidates ranking (Zaroor et al., 2017). In our case, we relied on this latter approach to deal with the person-job fit issue.

Candidate ranking consists on according a score to each submitted resume in such a way that the appropriate candidate will have the highest score. One way to handle with this problem is by modeling it as a classification or a regression issue where machine learning approaches are used. These approaches require a large amount of labeled data to achieve good results, which are not available because resumes and job posts data are sensitive and they are not shared publicly. An alternative to this approach is based on deep learning techniques. The idea is to learn representations (namely semantic features) from the free text of the job description

[1]Source.

and resume, then apply cosine similarity to the extracted representations to calculate the matching score. The semantic representation could be extracted by using shallow feed forward neural networks such as word2vec (Mikolov et al., 2013) and doc2vec (Le and Mikolov, 2014), or by using deeper and advanced neural network such as Convolutional Neural Network (CNN) (Lecun et al., 1998), Recurrent Neural Network (RNN) (Elman, 1990) and attention based model (Vaswani et al., 2017; Devlin et al., 2018). While training these models does not required labeled data, they suffer from the interpretation problem due to the complex internal transformations. However, in real deployment, it is necessary to explain why a candidate is accepted or rejected for a given job post.

To address the issues associated with the previously highlighted techniques, we propose in this work a hybrid approach that takes benefits from deep learning techniques to learn semantic representations from the job description and resumes, and to learn other features that improve the model performance. We summarize the contributions of our work as follows:

- Our proposed model performs on raw French data (often PDF files), we do not use at any time a form to get structured data. Thus, we propose a new approach to extract key contents form unstructured data.
- Besides learning representation from the free text in the resume and job description, our method extract several human-readable features that help to explain the matching results.
- Propose several techniques to fuse all the extracted features to get a final matching score for each resume.

2 Related work

A good person-job fit model must capture the semantic of the resume and the job description. This can be achieved via deep learning based models. Nowadays, significant improvements are observed in text representation using deep learning techniques (Devlin et al., 2018; Mikolov et al., 2013; Pennington et al., 2014). Such representations were used in several works to compute the similarity of the query and candidate documents. Authors in (Qin et al., 2018) proposed a word-level semantic representation for both job requirements and job seekers' experiences based on RNN. It was enhanced with four hierarchical ability-aware attention strategies that measure the different importance of job requirements for semantic representation, as well as measuring the different contribution of each job experience to a specific ability requirement. All these representations were combined and fed into a binary classification network. A similar work was done by (Zhu et al., 2018) where the authors used CNN to project both job postings and candidates resumes into a shared latent representation. The cosine similarity was applied against these representations to estimate the final matching score. These approaches perform on the free text in resumes and job descriptions; authors in (Jiang et al., 2020) proposed, in addition to the latent representation extracted from the raw text, to learn other representations from entities extracted from the whole resume and job description. In addition, they exploited the past actions to infer the preference or intention of candidates and recruiters.

The learning-to-rank models were also used to handle the person-job fit issue (Faliagka et al., 2012; Le et al., 2019). The idea is to combine predefined features (extracted from resumes and job descriptions) for ranking by means of supervised learning algorithms. Currently, ranking is transformed into a pairwise classification or regression problem where for each candidate we predict classes or scores depicting how well the candidate is in accordance with the job requirement. Training these models required labelled data of the form (resume-job features, score), which are unfortunately not available in most cases. This limits the use of these models to handle with the person-job fit issue.

In our work, we take advantage from the two approaches: the semantic based matching approach and the ranking based approach. The proposed model learn several representations from the free text in the resume and job description and from other entities extracted from the whole text. These representations are human-readable, which allows us to explain and to interpret the matching results. For each representation, scores are calculated to generate other feature that are combined according to several machine learning techniques to estimate a final score showing the relevance degree of the candidate.

3 Proposed approach

We divide the resume-job fit issue into three subtasks as it is shown in Figure 1:

Documents analysis. A resume or a job description could be a .PDF, a .DOCX or a .TXT file. In this stage, we extract the text in a structured format from these documents.

Features extraction. Extract semantic and human-readable information from resumes and job descriptions.

Score calculation. Use the extracted features to calculate scores and use them to perform classification, ranking or matching tasks.

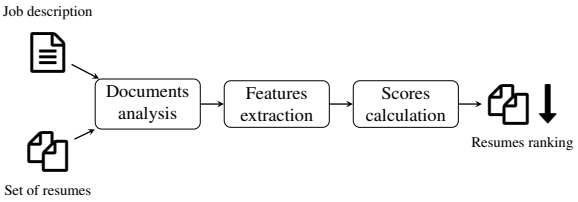


Figure 1: The architecture of proposed model.

3.1 Document analysis

While several tools were proposed to extract text from .PDF or .DOCX files[2], they failed to process files with complex structure. In fact, resumes layout is entirely at the discretion of the author, they can include list format (Figure 2a), double column format (Figure 2b) or combined format (Figure 2c) as shown in Figure 2.

(a) List (b) Column (c) Combined

Figure 2: Most used layouts in resumes.

The extracted text from resumes that include list format is spatially consistent, that means that the logical flow in the original document will match the physical flow extracted from the text. However, in the case where the resume includes double column or combine format, the logical flow will not be the same as the physical flow (starting from left (first column) to right (second column)). To handle with this issue, we built heuristic rules based on the spacing of the text on the page. In fact, for each text zone, we stored its x position and used it afterward to calculate gaps between the different zone texts. In the case where the gap is greater than a fixed threshold, we suppose then that the two zones belong to two different columns. This allows us to change the naive physical flow returned by the conversion tools to approximate the true logical flow of the document. An example of extracted text with and without our proposed heuristic rules is given in Figure 3.

Once the text is extracted from the original document, we still need to cluster it in order to detect the different sections mentioned in the resume (for example: experience, education, etc.). This is very useful to extract the experience years or the education level of the candidate. For this, we used metadata like font size, style, etc. to detect different headings from the original document and consider them as the existing sections. This is justified by the fact that heading properties remain the same in the whole resume. As the extracted text is logically ordered, we used these headings to divide the text into different sections.

By combining the consistent structure extracted from the original document and the available metadata, we were able to extract the full information contained within the resume while keeping the structure intact.

3.2 Features extraction

In the aim to make the matching results human explained, seven features were extracted from the resume and job description:

3.2.1 Count features

The count features aim to provide a simple way to tokenize both a resume and a job description, to build a vocabulary of known words and to encode combined documents (resume and job description) using that vocabulary. In order to enrich the representation of these features for our task and capture contextual information, they were calculated by using n-grams where $n \in \{1, 2, 3\}$.

The resulting features at this stage are multidimensional vectors that encode the frequency of each n-gram in the resume and the job description. We calculated afterward the cosine similarity between these vectors to measure how similar the resume and the job description are according to the

[2] Pdfminer, PyMuPDF and python-DOCX.

(a) A part of the original resume

```
Ingénieur systèmes(Linux/DevOps)
Etant administratrice systèmes
réseaux ; je me considère comme
acteur responsable de bon
fonctionnent de système
d'information , de la continuité
et la fiabilité de l'information:
- Vise à jouer des rôles
d'administrateur
réseau/système/sécurité/DevOps
Certifications et formations

2019 RHCSA  Red Hat Certified
System Administrator
.
.
.
```

(b) w/o heuristic rules

```
Ingénieur systèmes(Linux/DevOps)
Etant administratrice systèmes
réseaux ; je me considère comme
acteur responsable de bon
fonctionnent de système
d'information , de la continuité
et la fiabilité de l'information:
- Vise à jouer des rôles
  d'administrateur
  réseau/système/sécurité/DevOps
Diplômes et Formations
Diplôme national d'Ingénieur en
Sciences Appliquées et
Technologies / Université
Tunis El Manar - Institut
Supérieur d'Informatique (ISI)
Ariana, Tunisie
.
.
.
```

(c) With heuristic rules

Figure 3: Example of extracted text with and without heuristic rules. Red and blurred texts in the sub-figure 3b do not respect the logical flow of the original document. Personal information were blurred.

n-gram frequency.

3.2.2 TF-IDF features

The count features are very basic; for a token that has a large count (for example stop words), its corresponding value will not be very meaningful in the encoded vector. TF-IDF (Term Frequency-Inverse Document) features resolves this issue by integrating the inverse document term. This latter aims to downscale n-grams ($n \in \{1, 2, 3\}$) that appear a lot across documents. By using TF-IDF features, we were able to generate new multidimensional vectors that encode the resume and the job description. As it was done with count features, we calculated the cosine similarity between these vectors to obtain a new TF-IDF based feature.

3.2.3 Semantic features

The count and TF-IDF features are based on the token frequency in the document; they do not capture at any time the semantic relation between document tokens. Recently, deep learning based techniques were used for text representation where context and semantic of each token are captured efficiently. Examples for these: BERT (Devlin et al., 2018), attention based models (Vaswani et al., 2017), GPT (Brown et al., 2020) and ELMO (Peters et al., 2018). The advantage of these approaches is that their training is based on unlabeled textual data, but they require a large amount of data to success their training; for example, the initial BERT model was trained on more than 3 billion words. While pretrained models are available for free access and can be used in our case to generate semantic features, they are not adapted to our task where several technical terms could be found in the resume and job description. In order to prevent training a new model from scratch because only small amount of data are available in our case, we opted for the doc2vec technique (Le and Mikolov, 2014).

Doc2vec is based on a shallow neural network, which is trained to encode a document into a multidimensional vector in such a way that documents that share the same context will be closed in the multidimensional space. To train this model, we collected job descriptions from pole emploi website by using keywords like: *informatique (computer science)*, *développeur (developer)*, *data scientist*, etc. We extended this corpus with a collection of former candidate resumes at AYMAX consulting company. Statistics about the training corpus are given in Table 1.

Corpus	#Documents	#Words	#Unique words
Job descriptions	2242	645k	11k
Resumes	110	68k	4k

Table 1: Statistics about the doc2vec training corpora.

The resulted doc2vec model project the resume and the job description into 50-dimensional space. As the previous extracted features, we used cosine similarity to measure the semantic similarity between documents.

3.2.4 Skills feature

The human who is matching between job offer and candidates must consider the required skills and the candidate's skills, that is why it is necessary that the proposed model make explicit matching between the required and the possessed skills. This

process starts with a skill extraction stage. Skills extraction from raw texts is challenging; if we take as an example Python and Java, they could be an animal and an island in Indonesia, respectively, or two programming languages. One way to handle with this issue is to use a Named Entity Recognition (NER) model, which is trained to extract skill entities from an input text. We used a pre-trained model[3] that is able to identify 2K technical skills. We extended this model to recognize more than 200 French skills.

Once skills are extracted from the resume and job description, the matching skills feature is calculated as shown in the Equation 1.

$$skillsFeature = \frac{|s_j \cap s_r|}{|s_j|} \quad (1)$$

where s_j is the set of skills extracted from the job description while s_r is the one extracted from the resume and $|x|$ function means the cardinality of the set x.

3.2.5 Experience years feature

Another important feature that could be used to enhance the matching between the resume and job description is the number of experience years. A candidate with the highest experience years is more likely to get the job than another with less experience years. Our approach to calculate experience years is to extract from the experience section all dates by using a rule based approach and calculate afterwards the total number of years. This latter was considered as experience year's feature.

3.2.6 Spelling errors feature

A candidate resume with several spelling errors could reflect that the candidate is not a fastidious person. For this reason, we decided to calculate the percentage of misspelling words in the resume by using a list of 320k French words extended with more than 22K technical words collected from LinkedIn plate-form. This latter stage is justified by the fact that resumes include several technical terms that will be considered as misspelled words in the case where only the French list of words is used. The percentage of misspelled words is calculated as shown in Equation 1. The obtained value was used as a spelling errors feature.

$$spellingErrorsFeature = \frac{|l_{French} \cap l_{resume}|}{|l_{resume}|} \quad (2)$$

where l_{French} is the list of French words, l_{resume} is a list of resume words and $|x|$ function means the cardinality of the set x.

3.2.7 Document layout feature

The resume is the first contact with the employer; that is why it matters how the candidate structures his resume and what information should include. The document layout feature aims to detect whether the resume is well-structured and included the minimum of required information. A good resume should include: 1. contact information (phone number or email address), 2. education section, 3. experience section, 4. skills section. And it should respect the following constraints: 1. the non-presence of personal information (marital status for example), 2. a number of pages less than 3 pages, 3. and the use of short sentences (less than 50 words).

Once the previous constraints were checked, the document layout feature is calculated according the Equation 3

$$documentLayoutFeature = \sum_{i=1}^{7} f(C_i) \quad (3)$$

with C_i are the constraints that the resume should include and $f(x)$ is defined as follow:

$$f(x) = \begin{cases} 1 & \text{if the resume respect the constraint x} \\ 0 & \text{otherwise} \end{cases}$$

The obtained value was used as document layout feature.

3.3 Score calculation

The features extracted and presented in previous sections were combined to calculate a final matching score depicting how well the candidate is in accordance with the job requirements. This issue was interpreted as:

Weighted score. The final score in this case was calculated as a weighted sum as shown in the Equation 4.

$$score = \sum_{i=1}^{7} w_i feature_i \quad (4)$$

[3] Source

where w_i are the weights associated to each $feature_i$. These weights could be fixed manually according to the recruiter's preference or estimated to maximize the precision on a validation corpus.

Machine learning based score. In this case, the extracted features were combined to perform a candidate classification. In fact, by using a labelled corpus (see below in section 4.1), we trained several machine learning based models while using as input our extracted features to classify resumes into two classes: accepted and rejected.

4 Experiment results

The evaluation of our proposed approach is based on the precision and f1-score calculation. For this, one need a labelled data where for each pair $< job, resume >$ a label as either 0 (not match) or 1 (match) should be provided.

4.1 Data sources

To the best of our knowledge, there is no standard large open source corpus for the person-job fit task, and even less so for French. Therefore, this paper uses an internal small dataset containing a set of 2054 candidate resumes that were applied for 121 jobs. The labelled corpus was split into three parts: 80% for the training (Train), 10% for the validation (Dev) and 10% for the test (Test). Figure 4 illustrates the total number of accepted/rejected candidates in this corpus.

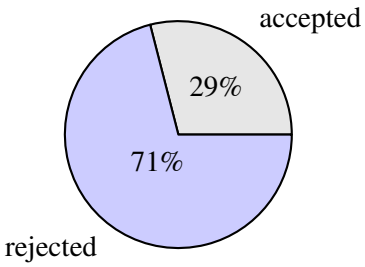


Figure 4: The percentage of accepted and rejected candidate in our corpus.

As the Figure 4 shows, our dataset is imbalanced. As a result, when we make the final decision for each candidate (accepted or rejected) by using machine learning approaches, the learning algorithm will make the decision rule biased towards the majority class. To deal with this issue, we: 1. upsample the minority class (Up), 2. downsample the majority class (Down), 3. and generate synthetic training examples by using Synthetic Minority Oversampling Technique (SMOTE) (Bowyer et al., 2011).

The number of samples after sampling the original Train dataset are given in Table 2.

Sampling	Orig	Up	Down	SMOTE
#Samples	1643	2554	732	2298

Table 2: Number of entries in the training dataset (Train) before and after applying the sampling techniques.

4.2 Results and discussion

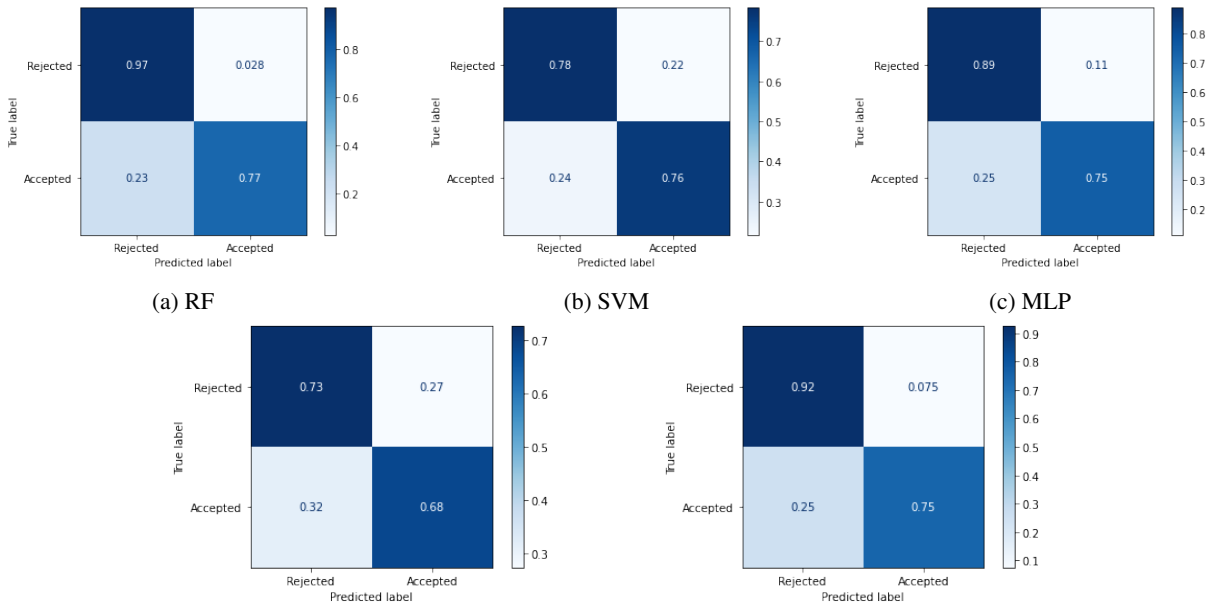
The final score of each candidate is calculated either by combining our proposed score features via a weighted sum (ranking issue) or via machine learning models (classification issue).

4.2.1 Weighted score

In the case where candidates are ranked according to the weighted features sum, the model evaluation requires a reference dataset including a set of resumes and job descriptions with scores, which are unfortunately not available. For this, the evaluation is carried out in this case by calculating the precision of the model to rank the accepted candidates among the top 5 and 10 candidates. We found that among the accepted candidates, 70% were classed on top 5 candidates according to our calculated relevant score and 88% were classed on top 10 candidates. This shows that ranking candidates by using the weighted score allows us to identify the potential candidates with a good accuracy. It should be noted that the features weights are estimated by maximizing the precision on the Dev corpus. We found that assigning a high weights to the features extracted from jointly the resume and the job description (count, TF-IDF, semantic and skills features) gives better results.

4.2.2 Machine learning based score

In the aim to evaluate the model on a large scale, we classified the candidate resumes into two classes (accepted or rejected). This was achieved by training several machine learning models while taking as input the different feature's scores. The trained models are: random forest (RF) (Ho, 1995), support vector machine (SVM) (Cortes and Vapnik, 1995), logistic regression (LR), K-Nearest Neigh-



(a) RF (b) SVM (c) MLP

(d) LR (e) KNN

Figure 5: Confusion matrices showing the classification precision for each class by using the different machine learning models.

bors (KNN) (Altman, 1992) and Multi-layer perceptron (MLP) (Popescu et al., 2009).

As the dataset is imbalanced, we carried out a comparative analysis between the sampling techniques: upsample the minority class, downsample the majority class and SMOTE. The obtained results (see Figure 6) shows that training the different models by upsamling the minority class gives better results in terms of f1-score. In should be noted that the evaluation was carried out on the Dev part by using cross validation on 10 folds.

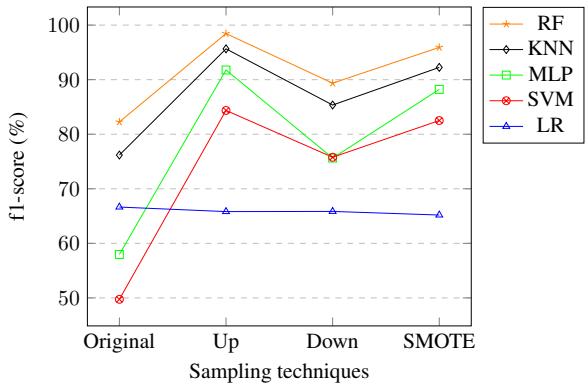


Figure 6: F1-score evolution with respect to the sampling techniques.

In the following experiments, all the models were trained by upsampling the minority class. The obtained results on the test part are illustrated in Table 3.

Model	Precision	F1-score
RF	93.7	84.3
KNN	88.6	74.6
MLP	85.6	70.1
SVM	77.9	60.6
LR	71.8	52.1

Table 3: Resumes classification results.

Results of Table 3 show that the best classification model is the one based on random forest algorithm with a precision of 93.7% and a f1-score of 84.3%. Since our dataset is imbalanced, it is more convenient to consider the f1-score rather than the precision. The gaps between the precision and the f1-score in all models is justified by the nature of our dataset. Class imbalance influences the learning algorithms during training by making the decision rule biased towards the majority class (rejected candidates). In the aim to highlight this point, Figure 5 sets forth the confusion matrices for each model.

It is clear that the model is biased towards predicting rejected candidates; the precision of these latter is more accurate than the accepted candidates class. However, the obtained precision on the accepted candidates remains acceptable for all models. In fact, the confusion matrix of the random forest based model shows that of all the candidates who are accepted, our algorithm identifies 77% of

them accurately, which is not insignificant. This confirms that the extracted features from resumes and job description are informative enough, and the accuracy results should be improved by using a large and balanced dataset. In the same vein, we found that the most important features allowing us to obtain a precision of 93.7% with the random forest based model are those extracted from jointly the resume and job description, as it is shown in Figure 7. This confirms the obtained results in the previous section, where the features were weighted for the final score estimation and candidate ranking.

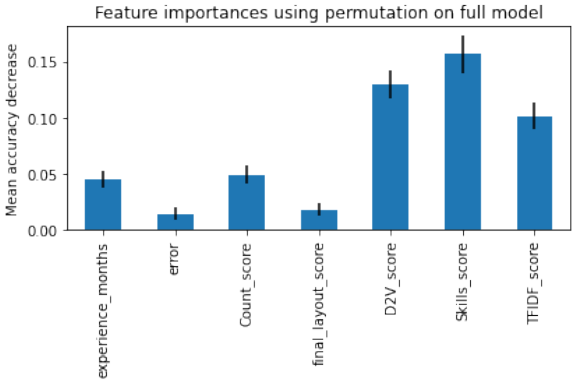


Figure 7: Features importances from Random Forest based model.

4.3 Prototype implementation

The proposed model was fully implemented as a web application where the recruiters have the possibility to select a job description and a set of candidate resumes locally or from the online recruitment platform APEC as it is shown in Figure 8.

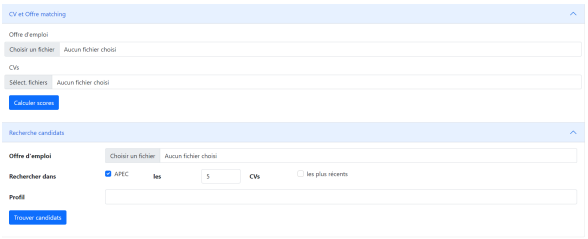


Figure 8: Recruiter interface for the candidate resumes selection.

Once the recruiter selects the job description and a collection of resumes, and upon his request, the system estimates applicant's relevance scores and ranks them accordingly. This is achieved by calling our model via an API. The system provides more detailed information about the candidate (contact information, skills, resume layout information, etc.) as it is shown in Figure 9.

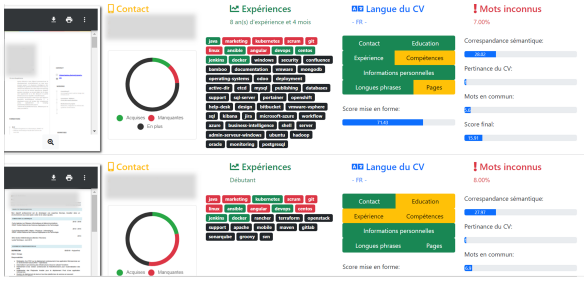


Figure 9: Candidate ranking results

5 Conclusion

In this work, we proposed a novel and an interpretable approach to deal with the person-job fit. It leverages on human-readable features extracted from the resume and job description that aim to make the matching results human explained. The proposed approach deals with raw data (.PDF or .DOCX files) with a complex structure. We proposed a heuristic rules based approach to extract structured information from such documents and used them to perform features extraction. Some features were extracted jointly from resumes and job descriptions (count, TF-IDF, semantic and skills features) and others were extracted only from resumes (experience years, spelling errors and document layout features). The extracted representations were fed to several machine learning models to perform resumes classification and ranking. The obtained results on an internal dataset showed a good precision and f1-score of 93.7% and 84.3% respectively. Unfortunately, the lack of models and French resumes data has not enabled us to compare our model with others. However, the obtained results demonstrated that the extracted features are informative enough to handle with the person-job fit issue. We found also that features extracted from jointly the resume and job description are more informative than the ones extracted exclusively from the resume. The proposed model was fully implemented as web application where resumes are ranked on the base of the obtained scores and an overview about each candidate was also displayed, which will save a lot of time and a lot of effort in the recruitment process.

References

N. S. Altman. 1992. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185.

Kevin W. Bowyer, Nitesh V. Chawla, Lawrence O. Hall, and W. Philip Kegelmeyer. 2011. SMOTE: synthetic minority over-sampling technique. *CoRR*, abs/1106.1813.

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.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

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.

Jeffrey L. Elman. 1990. Finding structure in time. *Cognitive Science*, 14(2):179–211.

Evanthia Faliagka, Kostas Ramantas, Athanasios Tsakalidis, and Giannis Tzimas. 2012. Application of machine learning algorithms to an online recruitment system.

Akshay Gugnani and Hemant Misra. 2020. Implicit skills extraction using document embedding and its use in job recommendation.

Tin Kam Ho. 1995. Random decision forests. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 1, pages 278–282 vol.1.

Junshu Jiang, Songyun Ye, Wei Wang, Jingran Xu, and Xiaosheng Luo. 2020. Learning effective representations for person-job fit by feature fusion. *CoRR*, abs/2006.07017.

Quoc V. Le and Tomás Mikolov. 2014. Distributed representations of sentences and documents. *CoRR*, abs/1405.4053.

Ran Le, Wenpeng hu, Yang Song, Tao Zhang, Dongyan Zhao, and Rui Yan. 2019. Towards effective and interpretable person-job fitting. pages 1883–1892.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Ranjana Patel and Santosh K. Vishwakarma. 2020. An efficient approach for job recommendation system based on collaborative filtering. In *ICT Systems and Sustainability*, pages 169–176, Singapore. Springer Singapore.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543.

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*.

Marius-Constantin Popescu, Valentina E. Balas, Liliana Perescu-Popescu, and Nikos Mastorakis. 2009. Multilayer perceptron and neural networks. *WSEAS Trans. Cir. and Sys.*, 8(7):579–588.

Chuan Qin, Hengshu Zhu, Tong Xu, Chen Zhu, Liang Jiang, Enhong Chen, and Hui Xiong. 2018. Enhancing person-job fit for talent recruitment: An ability-aware neural network approach. *CoRR*, abs/1812.08947.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

Abeer Zaroor, Mohammed Maree, and Muath Sabha. 2017. A hybrid approach to conceptual classification and ranking of resumes and their corresponding job posts.

Chen Zhu, Hengshu Zhu, Hui Xiong, Chao Ma, Fang Xie, Pengliang Ding, and Pan Li. 2018. Person-job fit: Adapting the right talent for the right job with joint representation learning. *CoRR*, abs/1810.04040.

Beam Search with Bidirectional Strategies for Neural Response Generation

Pierre Colombo[†⋆], **Chouchang (Jack) Yang**[*], **Giovanna Varni**[⋆], **Chloé Clavel**[⋆]
[⋆]Télécom ParisTech, Université Paris Saclay
† IBM GBS France
[*] University of Washington
pierre.colombo@ibm.com
ccjack@uw.edu
giovanna.varni@telecom-paris.fr
chloe.clavel@telecom-paris.fr

Abstract

Sequence-to-sequence neural networks have been widely used in language-based applications as they have flexible capabilities to learn various language models. However, when seeking for the optimal language response through trained neural networks, current existing approaches such as beam-search decoder strategies are still not able reaching to promising performances. Instead of developing various decoder strategies based on a "regular sentence order" neural network (a trained model by outputting sentences from left-to-right order), we leveraged "reverse" order as additional language model (a trained model by outputting sentences from right-to-left order) which can provide different perspectives for the path finding problems. In this paper, we propose bidirectional strategies in searching paths by combining two networks (left-to-right and right-to-left language models) making a bidirectional beam search possible. Besides, our solution allows us using any similarity measure in our sentence selection criterion. Our approaches demonstrate better performance compared to the unidirectional beam search strategy.

1 Introduction

Seq2seq models have shown state-of-the-art performance in tasks such as machine translation (Sutskever et al., 2014), neural conversation (Li et al., 2016b), image captioning (Venugopalan et al., 2015), and text summarization (Nallapati et al., 2016), exhibiting human-level performance (Johnson et al., 2017). Despite some drawbacks (e.g huge parallel corpora are needed to train seq2seq models, expert knowledge required to set the hyperparameters), seq2seq models are becoming increasingly popular and are now deployed in real world applications (McCann et al., 2019). During inference, a trained seq2seq model aims to find the best sentence given a source sentence. Since searching for all the possible paths is not practical but also computationally expensive, existing work relies on beam search based algorithms to solve this issue (Lipton, 2015). Current solutions have limited performances due to three major constraints: (1) beam search sentence selection is based on likelihood regardless of the evaluation metric, (2) during the generation process, only left to right dependencies (right to left are ignored) are considered by the model, (3) seq2seq strongly foster safe sentences (Li et al., 2016a): during generation, the influence of the input decreases while words are generated (Zhang et al., 2018) meaning that the end of the sequence is less likely to be input relevant. Those limitations constraint beam search performances: on Switchboard Corpus with a Beam Size of 50 optimal re-ranking would yield to an improvement of 128% (see Supplementary for full table). For the evaluation metric we follow Li et al. (2016a); Colombo et al. (2019) adopt the BLEU-4 score to compare the algorithms.

In this work, we introduce two novel algorithms based on beam search (*VBS*) with different ranking procedure: (1) *BidiS* a generalisation of the work of Wen et al. (2015); Mimura et al. (2018) that uses a "reverse" decoder to re-score the produced sentence penaliszing sentences whose end is less likely given the input, (2) *BidiA* an algorithm, that looks at the closest pair by incorporating the similarity measure between two beams of hypothesis; one with sentences generated in the regular order, the other with sentences generated in the reverse order. Our results show that leveraging the reverse order can boost beam search performance leading to higher BLEU-4 score and more diverse responses compared to *VBS*. Complexity analysis further shows that our proposed algorithms have dramatically reduced computational cost compared to the traditional approaches.

2 Models

We notice that limitation (2) and (3) previously introduced can be solved by introducing bidirectionally in the beam search. Indeed, training a seq2seq to output the sentence in the reverse order can model right to left dependencies and reduces the path length between the input and the end of the sentence (the shorter the paths are, the easier it is to model dependencies) making the end of the sentence more dependent of the input and producing more diverse sequences and model right to left dependencies as well.

2.1 Preliminaries

Vanilla Beam Search (*VBS*) We denote B as the beam size, T as the maximum sentence length and V as the vocabulary size. A RNN encoder takes an input sequence $X = (x_1, ..., x_T)$, to learn the language model word by word during the training phase. When a language task is given, the decoder travels through paths at each time and keeps the B most likely sequences. At each step, *VBS* considers at most $V \times B$ hypothesis. The sequence likelihood is measured by using the score function in (Wu et al., 2016)

$$s(Y, X) = \frac{\log P(Y|X)}{lp(Y)} \quad (1)$$

where X is the source, Y is the current target, and $lp(Y) = \frac{(5+|Y|)^\alpha}{(5+1)^\alpha}$ is the length normalization factor. We select $\alpha = 0.6$, which produce a higher BLEU score as illustrated in (Wu et al., 2016). The beam search is stopped when exactly B finished candidates have been found (Luong et al., 2015). In the worst case, the algorithm will run for a maximum of T steps.

Regular and Reverse model training For the bidirectional beam search we train two different networks over the same dataset. The first seq2seq network called "regular" is trained to predict the sentence in the regular order. The second network called "reverse" is trained to predict the sentence in the reversed order. For example, if the regular network is trained with the pair "What do you like ?" / "I like cats !", the reversed is trained with the pair "What do you like ?" / "! cats like I". During decoding, the reverse model estimates right-to-left dependencies while the regular model estimates left-to-right dependencies. [1] Two different settings have been explored: (1) training two independent seq2seq, (2) sharing the encoder of the two seq2seq and training using the following loss $\mathcal{L}$ where:

$$\mathcal{L} = \alpha\mathcal{L}_{Rg} + (1 - \alpha) * \mathcal{L}_{Re} \quad (2)$$

$\mathcal{L}_{Rg}$ is the Cross Entropy loss computed with the regular decoder and $\mathcal{L}_{Re}$ is the Cross Entropy loss computed with the reverse decoder. Since both approaches exhibit comparable performances, we choose to share the encoder to minimise the number of parameters in our model.

2.2 Beam Search with Bidirectional Scoring (*BidiS*)

A Beam search generates word by word from left to right: the token generated at time step t only depending on past token, but would not affected by the future tokens. Inspired by the work of (Li et al., 2016a), we propose a Beam Search with Bidirectional Scoring (*BidiS*), which scores the B best candidates generated by the regular seq2seq model as follows:

$$s(Y_T, X) = \frac{\log P(Y_T^+|X)}{lp(Y_T)} + \lambda \times \frac{\log P(Y_T^-|X)}{lp(Y_T)} \quad (3)$$

where Y_T^+ and Y_T^- represents the final sequence in the regular order and reversed order respectively. Moreover, $P(Y_T^+|X)$ is computed by using the regular model while $P(Y_T^-|X)$ is computed by using the reverse model. λ[2] is optimized in the validation. The intuition here is as follows: after generation of B sentences from the regular seq2seq, the reverse model computes $P(Y_T^-|X)$, and assigns higher probabilities to sequences presenting a more likely right-to-left structure and more likely ending given the input. Since B best lists produced by our models are grammatically correct, the final selected options are well-formed and present the best combination of both directions.

2.3 Beam Search with bidirectional agreement (*BidiA*)

The previous algorithm has two weaknesses. Firstly it introduces a hyperparameter λ. Secondly, the reverse model is only used to re-score the sentences

[1] From graph topology viewpoints, the decoder procedure from the right side is very different from the left sides. During exploring graph from left to right the regular seq2seq faces a huge very likely first token while the reverse seq2seq has a very restraint choice (mainly punctuation). More details are included in Supplementary.

[2] Optimisation process shows that λ compensate the difference of scale between $\frac{\log P(Y_T^+|X)}{lp(Y_T)}$ and $\frac{\log P(Y_T^-|X)}{lp(Y_T)}$

generated by the regular model, meaning that potentially good sentences generated by the reverse model are not considered. We solve these two problems by proposing a Beam Search with bidirectional agreement (*BidiA*); a hyperparameter free algorithm that uses $\frac{B}{2}$ best sequences according to the reverse seq2seq model. Formally if S and $S\prime$ are the sets containing $\frac{B}{2}$ sequence generated by the regular and reverse model respectively, we output Y_{n_0} such that:

$$(Y_{n_0}, Y_{r_0}) = \underset{Y_n \in S, Y_r \in S\prime}{\arg\min}\ 1 - sim(Y_n, Y_r) \quad (4)$$

where sim represents any similarity measure between two sentences [3]. For our experiment, we propose two different choices: (1) an adaptation of the BLEU score, $BLEU_T$ where the corpus length is set to T to foster longer responses formally:

$$BLEU_T = BP_T \times exp(\sum_{n=1}^{N} w_n log(p_n)) \quad (5)$$

where the brevity penalty is set to $BP_T = \exp(1 - \frac{T}{c})$[4], p_n is the geometric average of the modified n-gram precisions, using n-grams up to length N and w_n are positive weights summing to one, (2) an adaptation of the Word Mover's Distance (WMD_T) (Kusner et al., 2015) (stopwords are removed and final score is multiplied by BP_T) that captures the relationship between words, by computing the "transportation" from one phrase to another conveyed by each word [5].

3 Results

3.1 Corpora and Metrics

Corpora: We evaluate our algorithms on two spoken datasets (specific phenomena appear when working with spoken language (Dinkar et al., 2020) compared to written text). (1) The Switchboard Dialogue Act Corpus (SwDA)a telephone speech corpus (Stolcke et al., 1998), consisting of about 2.400 two-sided telephone conversation. (2) The Cornell Movie Corpus (Danescu-Niculescu-Mizil and Lee, 2011) which contains around 10K movie characters and around 220K dialogues.

Metrics: To evaluate the performance and language response quality for each decoder strategy, we use two classical different metrics at the sentence level. (1) A BLEU-4 score (Papineni et al., 2002) is computed on the unigrams, bigrams, trigrams and four-grams; and then micro-averaged. (2) A Diversity score: distinct-n (Li et al., 2016a) is defined as the number of distinct n-grams divided by the total number of generated words. Indeed, in neural response generation, we want to avoid generate generic responses such as "I don't know", "Yes", "No" and foster meaningful responses.

3.2 Response Quality

Figure 1 shows our proposed system results in BLEU-4 score metric. We see that our proposed methods (*BidiS* and *BidiA*) achieve better performances than *VBS* showing that bidirectionality boosts performances. *BidiA*$_{BLEU_T}$ achieves the best result overall yielding to a relative improvement of 9% on Cornell and 5% on SWA. From Figure 1, we see that for two different metric sim *BidiA* leads to better results than both other algorithm. Improvement of *BidiS* over the baseline *VBS* shows that the optimisation of λ on the validation set leads to good generalisation on the test set. *BidiA*$_{BLEU_T}$ is slightly better than *BidiA*$_{WMD_T}$ which is likely to be related to the choice of evaluation metric. From Figure 1 we can see that the BLEU-4 score of *VBS* stop increasing when $B > 10$. *BidiS* and *BidiA* keep improving the quality of the sequence while more hypothesis are proposed. This suggests that our bidirectional beam search is more efficient at extracting best sentence as the number of hypothesis increases. From Figure 1 we can see that *VBS*, *BidiA* and *BidiA*$_{WMD_T}$ present a drop in the performance for a number of hypothesis of 20 and 40: when performing the beam search for 20 hypothesis we observe that the seq2seq is very confident about sentences that lead to lower BLEU-4 score. Those sentences are not considered when $B < 20$ and better sentences are extracted when the beam size increases. *BidiA*$_{BLEU_T}$ does not present this drop of performance this is due to the metric choice (based on overlaps) that selected different sentences from *BidiA*$_{WMD_T}$.

3.3 Rank Analysis

In this section we compare the index returned by *BidiA* and *Best Hypothesis* as shown in Figure 2. Figure 2 illustrates one of the limitation of likelihood based ranking when an off-shell metric is used for evaluation: the very most likely sentences

[3] sim does not need to be differentiable.

[4] Brevity penalty introduces diversity and foster longer sentences.

[5] Implementation details are given in supplementary

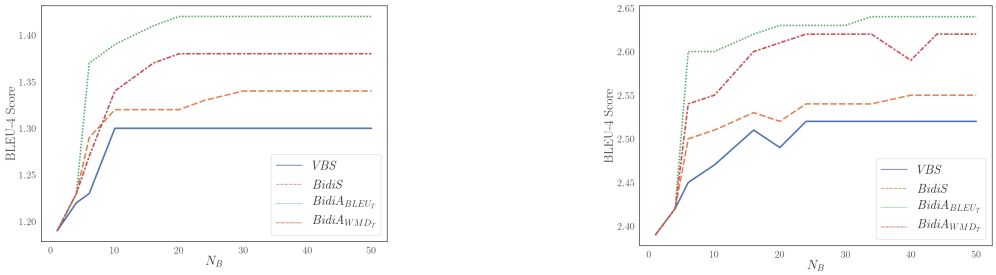


Figure 1: **BLEU-4 Scores** for the proposed algorithms on two different datasets: Cornell (left) and SWA (right). N_B is the beam size for *VBS* and *BidiS* and 2 times the beam size for $BidiA_{WMD_T}$ and $BidiA_{BLEU_T}$

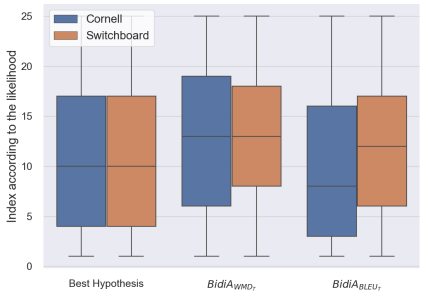


Figure 2: **Index of the response**. Index is the position of the sentence in the beam returned by *VBS*: the most likely sequence is ranked 1, the less likely is ranked 25. *Best Hypothesis* is the sentence (hypothesis) in the beam that yields to the highest BLEU-4.

are not the one with the highest BLEU-4. Interestingly, index distribution of *Best Hypothesis* is very similar for both Cornell and Switchboard, whereas for $BidiA_{BLEU}$ and $BidiA_{WMD_T}$ it varies. $BidiA_{BLEU}$ that has a better BLEU-4 (see Figure 1) than $BidiA_{WMD_T}$ has a index distribution more similar to *Best Hypothesis* than $BidiA_{WMD_T}$.

3.4 Diversity of the responses

Table 1 has shown the performance in diversity metrics. Overall, *BidiA* has the best performance among the other strategies (improvement up to 8% over the baseline for Cornell). By looking for an agreement between the reverse seq2seq and the regular one *BidiA* is able to extract sequences that are less likely according to the *VBS*, but more diverse. In all case, we see that bidirectionally helps to have more diverse sentences. Since the influence of the input decreases during the generation bidirectional beam search will output sentences that have both meaningful beginning and ending with respect to the input.

3.5 Complexity Analysis

In practical application it is important to evaluate the algorithm complexity when a limited amount of resources are available. Table 2 shows that *BidiA* is computationally cheaper than *VBS* and that *BidiS* has the same complexity as *VBS*.

	distinct-n			
	Cornell		Switchboard	
Model	n=1	n=2	n=1	n=2
VBS	0.051	0.250	0.042	0.231
BidiS	0.051	0.257	0.046	0.240
$BidiA_{BLEU_T}$	**0.056**	0.261	**0.050**	0.240
$BidiA_{WMD_T}$	0.054	**0.270**	0.048	**0.241**

Table 1: **Diversity Scores** we report the diversity score (distinct-n) for $N_B = 50$.

Algorithm	Complexity
$\mathcal{C}_{VBS}$	$T \times \mathcal{O}(BV \times \log(BV))$
$\mathcal{C}_{BidiS}$	$T \times \mathcal{O}(BV \times \log(BV))$
$\mathcal{C}_{BidiA_{WMD_T}}$	$2T \times \mathcal{O}(\frac{B}{2}V \times log(\frac{BV}{2}))$
$\mathcal{C}_{BidiA_{BLEU_T}}$	$2T \times \mathcal{O}(\frac{B}{2}V \times log(\frac{BV}{2}))$

Table 2: **Complexity of the different algorithms.** V is the size of the dictionary, B is the beam size, T is the maximum sentence length.

4 Conclusions

In this paper we show that bidirectional beam search strategies can be leverage to boost the performance of beam search. We have introduced two novel re-ranking criterions that select sentences with more diverse sentences and higher BLEU-4 and reduce computational complexity. Future work includes testing our novel bidirectional strategies with other pretrained models such as the one introduced in (Jalalzai et al., 2020; Chapuis et al., 2021, 2020; Witon et al., 2018), with other types of data (*e.g* multimodal (Garcia et al., 2019; Colombo et al., 2021a)), on differents tasks (*e.g* style transfer (Colombo et al., 2021b)) as well as exploring other stoping criterions (Colombo et al., 2021c).

References

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.

Emile Chapuis, Pierre Colombo, Matthieu Labeau, and Chloe Clave. 2021. Code-switched inspired losses for generic spoken dialog representations. *arXiv preprint arXiv:2108.12465*.

Emile Chapuis, Pierre Colombo, Matteo Manica, Matthieu Labeau, and Chloe Clavel. 2020. Hierarchical pre-training for sequence labelling in spoken dialog. *arXiv preprint arXiv:2009.11152*.

Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*.

Pierre Colombo, Emile Chapuis, Matthieu Labeau, and Chloe Clavel. 2021a. Improving multimodal fusion via mutual dependency maximisation. *arXiv preprint arXiv:2109.00922*.

Pierre Colombo, Chloe Clavel, and Pablo Piantanida. 2021b. A novel estimator of mutual information for learning to disentangle textual representations. *arXiv preprint arXiv:2105.02685*.

Pierre Colombo, Guillaume Staerman, Chloe Clavel, and Pablo Piantanida. 2021c. Automatic text evaluation through the lens of wasserstein barycenters. *arXiv preprint arXiv:2108.12463*.

Pierre Colombo, Wojciech Witon, Ashutosh Modi, James Kennedy, and Mubbasir Kapadia. 2019. Affect-driven dialog generation.

Cristian Danescu-Niculescu-Mizil and Lillian Lee. 2011. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics, ACL 2011*.

Tanvi Dinkar, Pierre Colombo, Matthieu Labeau, and Chloé Clavel. 2020. The importance of fillers for text representations of speech transcripts. *arXiv preprint arXiv:2009.11340*.

Alexandre Garcia, Pierre Colombo, Slim Essid, Florence d'Alché Buc, and Chloé Clavel. 2019. From the token to the review: A hierarchical multimodal approach to opinion mining. *arXiv preprint arXiv:1908.11216*.

Hamid Jalalzai, Pierre Colombo, Chloé Clavel, Eric Gaussier, Giovanna Varni, Emmanuel Vignon, and Anne Sabourin. 2020. Heavy-tailed representations, text polarity classification & data augmentation. *arXiv preprint arXiv:2003.11593*.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

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

Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *International Conference on Machine Learning*, pages 957–966.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature*, 521(7553):436.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and William B. Dolan. 2016a. A diversity-promoting objective function for neural conversation models. In *HLT-NAACL*.

Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. 2016b. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 994–1003. Association for Computational Linguistics.

Zachary Chase Lipton. 2015. A critical review of recurrent neural networks for sequence learning. *CoRR*, abs/1506.00019.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2019. The natural language decathlon: Multitask learning as question answering.

Masato Mimura, Shinsuke Sakai, and Tatsuya Kawahara. 2018. Forward-backward attention decoder. In *INTERSPEECH*.

Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, Çaglar Gülçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *CoNLL*, pages 280–290. ACL.

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 on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Andreas Stolcke, Elizabeth Shriberg, Rebecca Bates, Noah Coccaro, Daniel Jurafsky, Rachel Martin, Marie Meteer, Klaus Ries, Paul Taylor, Carol Van Ess-Dykema, et al. 1998. Dialog act modeling for conversational speech. In *AAAI Spring Symposium on Applying Machine Learning to Discourse Processing*, pages 98–105.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 3104–3112, Cambridge, MA, USA. MIT Press.

Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. 2015. Sequence to sequence - video to text. In *The IEEE International Conference on Computer Vision (ICCV)*.

Tsung-Hsien Wen, Milica Gašić, Dongho Kim, Nikola Mrkšić, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Stochastic Language Generation in Dialogue using Recurrent Neural Networks with Convolutional Sentence Reranking. In *Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*. Association for Computational Linguistics.

Wojciech Witon, Pierre Colombo, Ashutosh Modi, and Mubbasir Kapadia. 2018. Disney at iest 2018: Predicting emotions using an ensemble. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 248–253.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Gregory S. Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Ruqing Zhang, Jiafeng Guo, Yixing Fan, Yanyan Lan, Jun Xu, and Xueqi Cheng. 2018. Learning to control the specificity in neural response generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1108–1117.

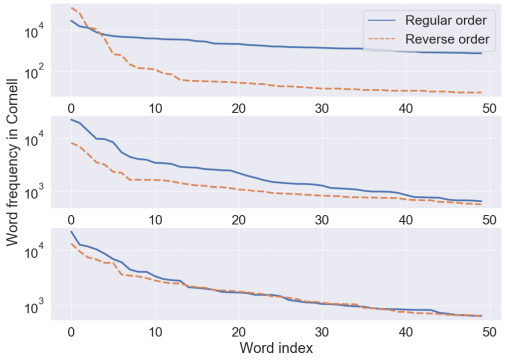


Figure 3: **Word distribution on Cornell for different word position in the sentence.** This figure shows the frequency of appearance of the 50 most common word in the sentence extracted from Cornell. An example of the regular order sentence is "The cat is red.", the associated reverse order is: ". red is cat The". Top plot shows the frequency of words appearing in position 1, the second plot for words appearing in position 2, the third plot for words appearing in position 3.

		BLEU-4			
	Beam Size	1	6	10	50
Corn.	*VBS*	1.19	1.23	1.30	1.30
	Best Hypoth.	1.19	1.40	1.60	2.56
SWA	*VBS*	2.39	2.45	2.47	2.52
	Best Hypoth.	2.39	3.40	4.30	5.77

Table 3: **BLEU-4 Scores on Cornell (Corn.) and Switchboard (SWA)**: *VBS* stands for the standard beam search (see section 2) *Best Hypoth.* is the hypothesis in the beam that leads to the highest BLEU-4. In our work performances of *Best Hypoth.* can be seen as an upper bound of the performances of *VBS*.

5 Supplementary Material

5.1 Corpus analysis: importance of reverse model

In this section, we discuss the importance of using a reverse model. Figure 3 shows the word distribution on Cornell for 50 most common words at each position. From top plot we observe that the regular seq2seq faces a lot of likely choices: all the fifty most common words appear more than 10^3 times in position 1. The reverse seq2seq faces less than 5 very likely choices in position 1 that appear more than 10^3 time. The reverse seq2seq is then less likely to transmit a mistake at time step 2.

5.2 Ideal reranking

In Table 3 we report the BLEU-4 achieved by *VBS* and *Best Hypothesis*: the best hypothesis alive in the beam. It illustrates that the limitations of the likelihood criterion and shows that making a change in the final reranking and sentence selection criterion can yield to higher BLEU-4.

5.3 Implementation of similarity measure (sim)

In this section we describes the implementation of each similarity sim used in section 3. We have introduce a brevity penalty for two main reasons:

- preliminary experiments have shown that the regular seq2seq tends to generate short sentences due to the data distribution.
- if no brevity penalty is introduced and both neural networks generate "I don't know" the selected sentence will be "I don't know" since the similarity measure will be 1. With a brevity penalty, similarity metric can select a less generic choice.

5.3.1 $BLEU_T$

$BLEU_T$ has been implemented by using the nltk librairy`https://www.nltk.org/`. In Equation 9 we set $sim = 1 - BLEU_T$.

5.3.2 WM_T

WM_T uses the wm-relax librairy (`https://github.com/src-/wmd-relax`), embeddings used are coming from FastText librairy (Bojanowski et al., 2017). At the first step stopwords according nltk list are removed, Word Mover Distance is computed and multiplied by BP_T previously defined. Formally, in Equation 9 we set $sim = WM_T$.

5.4 Architecture details

We evaluate our proposed algorithms by using off-the-shelf seq2seq models. For the encoder, we use two-layer bidirectional GRU (Chung et al., 2014) (256 hidden layers). For the decoder, we use a one-layer uni-directional GRU (512 hidden layers) with attention (Luong et al., 2015). The embedding layer is initialized with fastText pre-trained word vectors (on Wikipedia 2017, the UMBC web-based corpus and the statmt.org news dataset) and the size is 300 (Bojanowski et al., 2017). We use the ADAM optimizer (Kingma and Ba, 2014) with a

learning rate of 0.001, which is updated by using a scheduler with a patience of 100 epochs and a decrease rate of 0.5. The gradient norm is clipped to 5.0, weight decay is set to $1e^{-5}$, and dropout (LeCun et al., 2015) is set to 0.1. The models have been implemented with pytorch, they have been trained on 97%, validated on 1%, and tested on 2% of the data respectively. Since our purpose is to show that bidirectionality can boost beam search we set $\alpha = \frac{1}{2}$ in Equation 2.1.

5.5 Proofs of Complexity analysis

5.5.1 VBS complexity

For *VBS*, at each time step BV, hypotheses are re-ranked and the B most likely are kept. The final average complexity is:

$$\mathcal{C}_{VBS} = T \times \mathcal{O}(BV \times \log(BV)) \tag{6}$$

5.5.2 BidiS complexity

In the case of *BidiS*, the algorithm generates B sequences using *VBS*, and then for generating sequence Y_T it computes $\frac{\log P(Y_T^- | X)}{lp(Y_T)}$ with complexity $\mathcal{O}(T)$. The final step includes a sorting of complexity $\mathcal{O}(B \log(B))$.[6] *BidiS* complexity is:

$$\mathcal{C}_{BidiS} = T \times \mathcal{O}(BV \times \log(BV)) \tag{7}$$

5.5.3 BidiA complexity

Word Mover's Distance criterion: According to (Kusner et al., 2015) the computational cost of the Word Mover's Distance computation is $\mathcal{O}(p^3 \times \log(p))$, where p denotes the number of unique words in the documents. In our case the distance is computed between two sequences of length at most T, hence $p \leq 2T$. *BidiA*$_{WMD_T}$ complexity with Word Mover's Distance as selection criterion is given by the following formula:

$$\begin{aligned}\mathcal{C}_{BidiA_{WMD_T}} =& \underbrace{2T \times \mathcal{O}(\frac{B}{2}V \times log(\frac{BV}{2}))}_{\text{two } VBS \text{ with with beam size } \frac{B}{2}} \\ &+ \frac{B^2}{8} \times \underbrace{\mathcal{O}(8T^3 \times \log(2T))}_{\text{pairwise WMD}} \\ &+ \underbrace{\mathcal{O}(T)}_{\text{complexity of } BP_T}\end{aligned} \tag{8}$$

In general $T^3 \leq BV$ and $T \lll BV$, in Equation 9 the second term is small compared to the first term, hence $\mathcal{C}_{BidiA_{WMD_T}} \approx T \times \mathcal{O}(BV \times log(\frac{BV}{2}))$. Even thought V dominates the complexity of the algorithm, still *BidiA*$_{WMD_T}$ is more efficient than *VBS*. [7]

BLEU criterion: the computational cost of the $BLEU_T$ score is polynomial in T. *BidiA*$_{BLEU_T}$ complexity with BLEU score as the selection criterion is given by the following formula:

$$\mathcal{C}_{BidiA_{BLEU_T}} = 2T \times \mathcal{O}(\frac{B}{2}V \times log(\frac{BV}{2})) \tag{9}$$

[6] $\mathcal{O}(B \log(B))$ and $\mathcal{O}(T)$ have much less order compared to $\mathcal{O}(BV \times \log(BV))$ so they can be neglected here.

[7] For example if $T = 30$, $B = 30$, $V = 35$k we see that $\mathcal{C}_{VBS} = 1.4 \times \mathcal{C}_{BidiA}$.

A^{3}C: Arabic Anaphora Annotated Corpus

Abdelhalim Hafedh Dahou, **Mohamed Abdelmoazz** and **Mohamed Amine Cheragui**
Mathematics and Computer Science Department
Ahmed Draia University
Adrar, Algeria
dahou.halim1995@gmail.com, m.abdelmoazz@yahoo.com, m_cheragui@univ-adrar.edu.dz

Abstract

In this paper, we describe the different steps taken to build our annotated corpus which aims to treat a known linguistic phenomenon in Arabic texts called Anaphora. The objective behind the creation of this corpus[1] is to fill the lack of resources concerning the resolution anaphora (especially pronominal and verbal) in the Modern Standard Arabic language and this is by creating a newly annotated corpus that we have called A^{3}C which contains the anaphoric relations. To satisfy this objective, we created A^{3}T, an anaphoric annotating tool that uses linguistic and statistical rules to automatically detect anaphors and their referents. After that, we resort to human specialists to verify and correct our A^{3}T annotation's errors for the corpus's credibility. This study discusses novel features that can aid in determining the best reference, as well as the problem of the lack of resources for verbal anaphora.

1 Introduction

A corpus is considered today as a fundamental piece in natural language processing, due to the role that it plays in both the resolution and the testing phases. The building of annotated corpus in terms of number and size has known a real ascension in the last decades, in particular since the appearance of statistical and machine learning approaches (Beseiso and Al-Alwani, 2016), allowing, from textual resources, the development of resolution models for different linguistic phenomena such as anaphora.

Anaphora is typically defined as references to items mentioned earlier in a discourse or "pointing back " reference as described by (Mitkov, 99). In addition, the process of determining the referent of an anaphora and establishing the relationship between them is known as anaphora resolution. Anaphora still a very challenging linguistic phenomenon, where its identification and resolution can increase the performance of several NLP applications, such as: sentiment analysis (Cambria, 2016), question-answer systems (El-Said Nada et al., 2018), machine translation (Madhura and Satish, 2019), text summarization (Antunes et al., 2018), information extraction (Matysiak, 2007), language generation and dialog systems (Vinay et al., 2019).

Our motivation behind this work is to enhance anaphora resolution in Arabic text by building an anaphoric annotated corpus that can contribute to future works that tackle anaphora in the Arabic language.

This paper is structured in 6 sections. Section 2, describe the anaphoric typology in Arabic language. Section 3, gives an overview of existing anaphoric corpora (case of Arabic). Section 4, presents the challenges we face in Arabic anaphora resolution. Section 5, outlines the different phases of building of our A^{3}C corpus. Section 6, some observations noted during the building process of our corpus. The last section gives a conclusion and future work.

2 Varieties of Anaphora in Arabic text

What makes the anaphora resolution mechanism complex in natural language processing in general and in Arabic, in particular, is the fact that it can

[1] The Corpus is available for the community in : https://dahouabdelhalim.github.io/Anaphora-Corpus/

manifest in different forms (linguistic categories: lexical and grammatical), but also requires knowledge at different levels, as well as an "understanding" of the context. There are many varieties of anaphora in the Arabic text, we will only mention the most frequent ones.

2.1 Verbal anaphora

Verbal anaphora is used to describe or represent various movements or actions by using the verb (did - فعل -) and the different conjugation variants to minimize writing and avoid repetition (Trabelsi et al., 2016; Hamouda, 2014).

Translation: "Students who are about to graduate must complete their administrative file, if they do not, they will not receive their certificates."

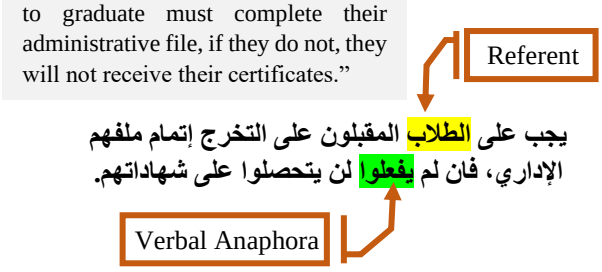


Figure 1: Example of verbal anaphora.

2.2 Lexical anaphora

Lexical anaphora occurs when the referent is designated by definite descriptions representing the same concept (the anaphora), or concepts that are semantically close (Hammami, 2009). Usually, this form of anaphora adds more information to the sentence and increases cohesion, and can take several forms (synonym, generalization / hypernymy, or specialization /hyponymy) (Seddik and Farghaly, 2014).

Translation: "Ibn Sina was born in Bukhara (in present-day Uzbekistan), one of the most famous writings of the scientist *The Canon of Medicine*".

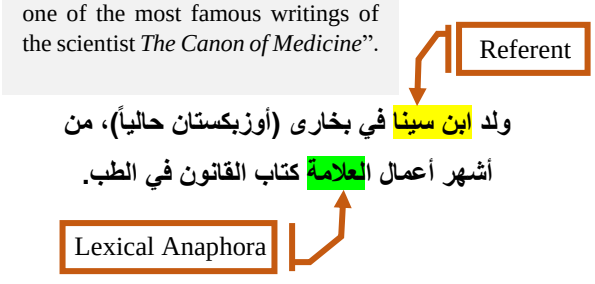


Figure 2: Example of lexical anaphora.

2.3 Comparative anaphora

This type of anaphora is manifested by the introduction of lexical modifiers (e.g., أخر / other, أخرى / one, وحدة) or comparative adjectives (أكبر من / greater than , أحسن من / better than) (Hammami, 2009). This variety of anaphora indicates a relation like: such as set-complement, similarity and comparison between the anaphora and the referent (Mahmoud Seddik and Farghaly, 2014).

Translation: "Departure of Vessels transporting more than 50,000 tons of agricultural nutrients (urea) and various other from the industrial port of AL Jubail in the Kingdom to the Port Sudan in Khartoum."

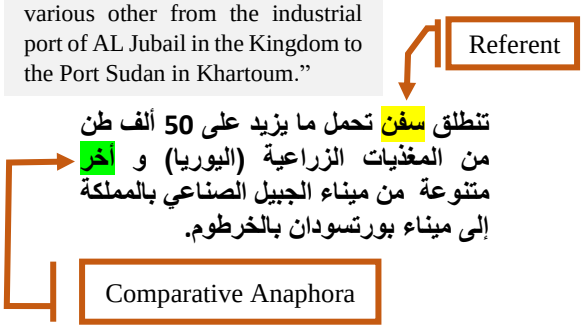


Figure 3: Example of comparative anaphora.

2.4 Pronominal anaphora

Based on statistical studies done by (Hammami, 2009) it shows that the pronominal anaphora is the most frequent variant in Arabic texts. Pronouns form a special class of an"aphora because of their empty semantic structures; they have a meaning independent of its referents and usually refer to names or noun phrases (Beseiso and Al-Alwani, 2016). However, not all pronouns are anaphoric.

Pronominal Anaphors can be divided into three categories, each category can be subdivided into subcategories according to several parameters, such as gender, number, etc.

3rd personal pronouns (ضمائر الغائب): In the Arabic, not all personal pronouns are anaphoric, so the 1st person (انا و نحن) and 2nd person (أنت انتما .., etc.) pronouns are not (they specify the communication partners and their meaning goes back to their specific uses), except the 3rd person pronouns which have this characteristic. These pronouns can be subdivided into two categories: disjoint pronoun (Example: هي / she ،هو/ he) and joint pronoun (Example: ـه، ا، ن) (El-Said Nada et al., 2018):

Translation: "Soumaia has sewn her sister's wedding dress and she is very excited".

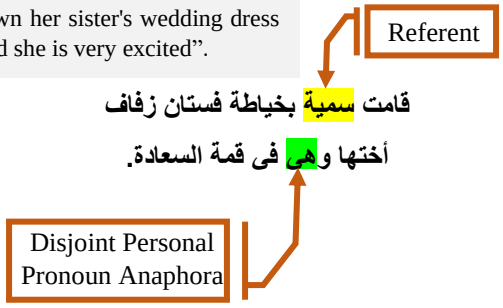


Figure 4: Example of disjoint personal pronoun anaphora.

In some cases the pronouns "ـه" and "ـها" are not anaphoric since they are not interpreted as related to an expression (referent). In this case we will call them pleonastic pronouns.

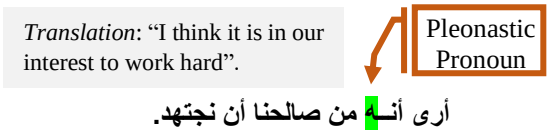


Figure 5: Example of pleonastic pronoun.

Relative pronouns (الأسماء الموصولة): Relative pronouns in Arabic have the characteristic of being always anaphoric, in addition they have only one possible referent (Trabelsi, 2016) and refer to the immediate nominal phrase mentioned before (Bouzid et al., 2014) which they agree in gender and number.

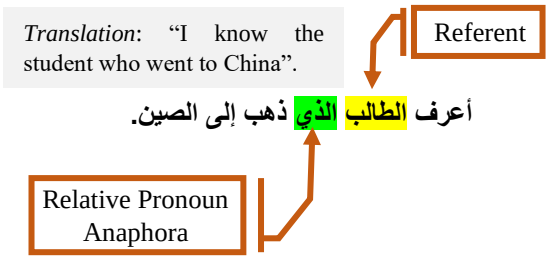


Figure 6: Example of relative anaphora.

The use of relative pronouns is possible if the referent denotes a process or situation, and here the anaphora denotes some of these lexical meanings. They refer to persons, places or things that are close or distant, the table below illustrates this type of pronouns.

Demonstrative pronouns (أسماء الإشارة): They are linguistic elements that accompany a designation gesture in order to coordinate the attention of the interlocutors when they are speaking (Jarbou, 2018). Generally, demonstrative pronouns are cataphoric and in some cases they can be anaphoric and even deixis (Bouzid et al., 2014). Demonstratives agree in person, gender and number with their referent. In addition, there are pronouns, which are considered demonstratives, and which designate time and place (Example: هذا / this, هنا / here).

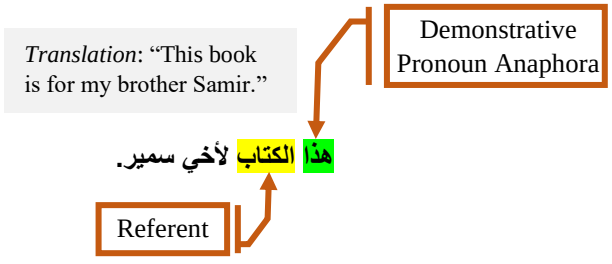


Figure 7: Example of demonstrative anaphora.

3 Related Work

For Arabic language, a considerable effort has been made concerning the anaphoric phenomenon during the last two decades, which is reflected by several studies aiming in their majority to solve the problem of the pronominal anaphora. The objective of this section is to present an overview of works dedicated to building annotated corpus (anaphora identification and referent determination).

Corpus	Size	Anaphoric Resolution Category
AnATAr (Hammami, 2009)	18895 words 2722 pairs of anaphor /referent.	Pronominal anaphora
(Hadder, 2000)	200 Sentences	Zero Anaphora
Holy Qur'an Corpus (Farghaly and Fahmy, 2015)	127,795 words 24,653 personal pronouns	Pronominal Anaphora
QAC (Sharaf and Atwell, 2012)	128,000 words 24,679 Pronouns	Pronominal Anaphora

Table 1 : Existing corpora concerning the Arabic anaphora.

4 Ambiguities and anaphoric resolution

The aim of this section is to present the main factors, which affect anaphoric resolution.

4.1 Ambiguities and lack of diacritics

Without diacritics marks, an Arabic text is extremely unclear (morphologically and grammatically). According to (Debili and Achour, 1998), 74% of Arabic words might potentially take several lexical diacritization, making it difficult to determine if the anaphoric phenomenon or referent is the case.

Word	Word + Diacritics	Translation
كـتـب	كَـتَـبَ	he wrote
	كُـتُـبٌ	books
	كُـتِـبَ	Written
	كُـتِّـبَ	was caused to write
	كَـتَّـبْ	To make someone to write

Table 2: Example of ambiguities due to the lack of diacritics.

4.2 Agglutination phenomenon

The Arabic script is characterized by the agglutination phenomena, which is explained by the fact of combining numbers of words in just one. Compared to French or English, an Arabic word can sometimes correspond to a full sentence (Bouzida and Zribi, 2020).

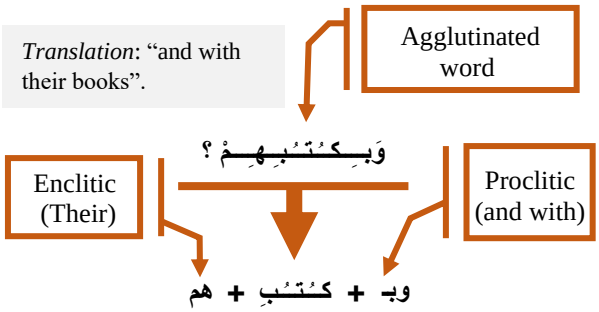


Figure 8: Example of agglutination.

4.3 Syntactic flexibility (Words free order)

Arabic is a nearly free-order language. This order causes artificial syntactic ambiguities, since the grammar should provide all the possible combination rules for reversing the order of words in the sentence. For anaphora resolution, this type of flexibility is a problem for referent localization (Beseiso and Al-Alwani, 2016; Fotiadou et al., 2020).

Sentences	English Translation	Order
قرأ محمد الكتاب	Mohamed Read the book	VSO
محمد قرأ الكتاب	Mohamed, he read the book	SVO
الكتاب محمد قرأه[2]	The book Mohamed read it	OSV
قرأه[3] الكتاب محمد	The Book was reading by Mohamed	VOS

Table 3: Words free order in Arabic sentences.

4.4 Ambiguity of the referent

This difficulty occurs when the referent is ambiguous (due to the presence of two or more referents for the same anaphora). In this case, external knowledge of the context is necessary to identify the correct referent (Brunner et al., 2002).

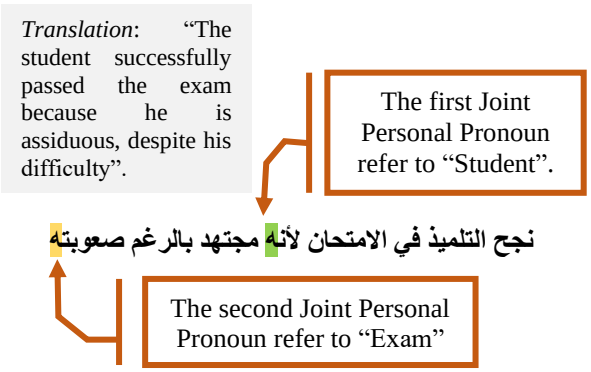


Figure 9: Example of ambiguity of the referent.

4.5 Hidden referent

This case occurs when the anaphora refers to something, which is not present in the sentence or text. The Qur'anic text is an example where this phenomenon persists (Seddik and Farghaly, 2011), so in the example below the pronominal anaphora (هو / he) refers to (الله /Allah) which is not present in the "Aya". The human through his knowledge and reasoning system can easily make the connection between the pronominal anaphora (هو / he) and (الله /Allah). However, for anaphoric resolution systems the task is complicated.

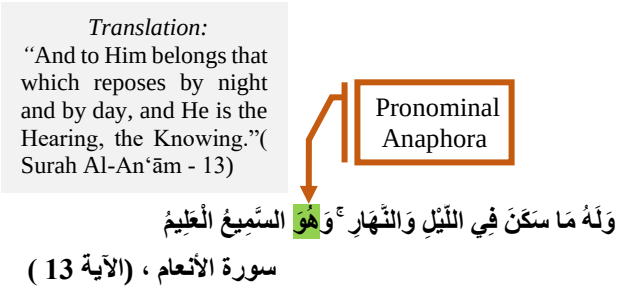


Figure 10: Example of hidden referent.

5 Building the A3 C

As mentioned above, the main objective is to provide an annotated resource that can be used in the automatic Arabic anaphora resolving systems. We decided to create an operational tool with a friendly interface that would help computer scientists and linguists to develop such resources.

In this section, we'll go over the steps involved in building our corpus A^{3}C and annotating it with our A^{3}T system. We thought about breaking down the creation of our work environment into three

[2] Joint Personal Pronoun « ه » are anaphoric.

[3] Joint Personal Pronoun « ه » are cataphoric

(03) phases: data collection, anaphora resolution system, corpus annotation and verification. Each phase consists of essential modules that take place to accomplish the phase's purpose.

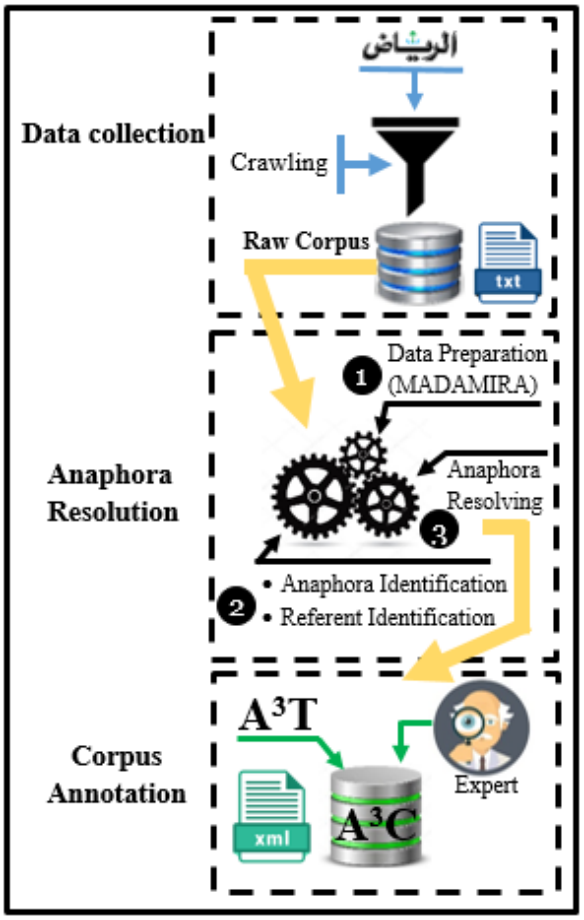


Figure 11: General architecture of building A³C.

5.1 Data collection

Our purpose is to build a corpus of texts from different fields to cover two types of anaphora, pronominal and verbal anaphora. The texts in our corpus are taken from the **Alriyadh newspaper**[4], a daily Arabic newspaper, and they are divided into five categories: culture, sports, politics, economy, and miscellaneous. The choice of those categories is made after an analysis of different categories of texts in terms of the number and diversity of anaphora types. On the other hand, the choice of this newspaper is due to the volume of information, good structure of articles and diversity of categories. To attend to this objective, we developed a crawler system that takes as an input the URL of the category page and the limited number of articles, then returns as an output a cleaned text file in (.txt) format.

[4] https://www.alriyadh.com/

5.2 Co-reference Resolution

We all know how effort and time consuming it is to manually resolve anaphora and annotate a text corpus. As a result, we created the A3T (Arabic Anaphora Annotating Program), a tool that manages resolution and annotation in an automatic way, while also providing a user-friendly interface to modify the results. The resolution process was divided into two sub-modules:

Data Preparation: To help us address the anaphora problem, the text corpus must go through three processes. The first step is to break each text file into sentences using a sentence splitter mechanism based on the punctuations. Secondly, organizing these sentences in a specific input structure to prepare them for the POS and morphological analysis (Figure 11). Finally, determine which grammatical category a given word belongs to and other morphological features such as gender, number, state, voice. The MADAMIDA tool was chosen for our purposes because of its 95.9% precision and high-quality word-level disambiguation as mentioned in (Pasha et al., 2014). The word-level disambiguation functionality will help us in the identification of the attached pronouns.

Translation: "A pilot attacks a building with his plane."

```
< preprocessed SEN 1 ="طيار يقتحم مبنى بطائرته" >
  <word id="1" word=" طيار ">
    <morph_feature_set pos="noun" vox="na"
      mod="na" gen="m" num="s" stt="c" />
  <word id="2" word=" يقتحم ">
    <morph_feature_set pos="verb" vox="a" mod="u"
      gen="m" num="s" stt="na"/>
  <word id="3" word="مبنى">
    <morph_feature_set pos="noun" vox="na"
      mod="na" gen="m" num="s" stt="i"/>
  <word id="4" word="بطائرته" >
    <morph_feature_set pos="noun" vox="na"
      mod="na" gen="f" num="s" stt="c" />
      <tok id="0" form0="ب+" form1="PREP+" />
      <tok id="1" form0="طائرَة" form1="NN" />
      <tok id="2" form0="+ه" form1 = " + POSS
      PRON 3MS" />
</preprocessed>
```

Figure 12: POS by MADAMIRA.

Anaphora Resolution System: A³T allows the expert to select text to automatically detect and resolve anaphora. Once selected, the following three steps are applied to detect and resolve the problem :

- Anaphora identification: Anaphora is identified by referring to their grammatical code, which is based on the MADAMIRA tag set. The output here is a list of all anaphora in the text with additional information like Id, Name, Gender, Number, and Sentence number. For the pronominal anaphora, the process differs from one type to another, for example, the POS tagging for pronominal attached anaphora doesn't have a tag for gender, number, and person because the output is in the attached form, we should apply a split mechanism to place each of them in their proper tag as illustrated in (Figure 13). On the other hand, for verbal anaphora identification, we combine all of the elements used for pronominal anaphora identification, such as gender, number, and so on, with a new feature that will aid in the resolution which is the voice feature (active or passive form). Tables 4 and 5 illustrate the distribution of the various types of anaphora in our corpus after applying this process.

Category	Pro. Ana[5]	POSS[6]	DEM[7]	REL[8]
Economy	18580	40.16%	36.25%	23.59%
Education	28540	47.44%	32.5%	20.06%
Politics	13210	48.73%	27.04%	24.29%
Sport	10953	51.75%	28.5%	19.75%
Miscellany	15069	43.78%	32.21%	25.01%

Table 4: Statistics about the A^{3}C corpus (A).

Category	Verbal Anaphora
Economy	1455
Education	1294
Politics	924
Sport	1370
Miscellany	1932

Table 5: Statistics about the A^{3}C corpus (B).

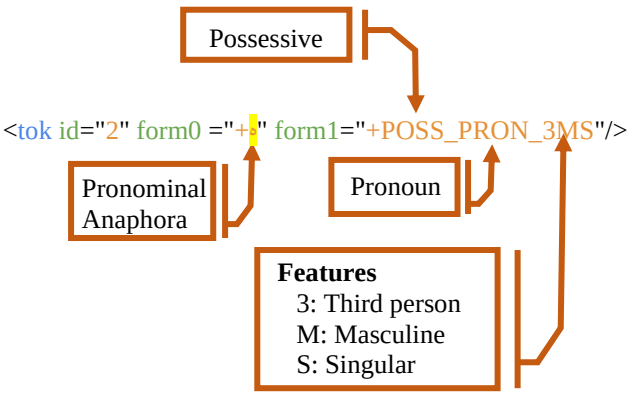


Figure 13: Example of pronominal anaphora identification

- Identification of referent candidates: Referents are chosen based on their POS (nouns, NPs and proper noun) and a specific search scope is adjusted based on some tests and previous research (Mitkov, 99). The search scope is still not fixed in the case of anaphora, but based on analysis, a high number of references occurs on the two previous sentences. In our case, we took two sentences before and as a special case for the demonstrative anaphora, we took the same number after. For the case of verbal anaphora, in the active form, we took two sentences after the verb and for the passive or unknown form; we took two sentences before the verb. The selection considers all of a candidate's features, including gender, number, voice, definiteness, and sentence number.

- Anaphora resolving: The goal is to choose the most appropriate referents from among the most likely candidates for each anaphora. We used morphological filters to remove unsuitable candidates by comparing gender, number, and existing sentence (search scope). To find the suitable referent, we used a collection of preferential factors that favor certain candidates over others, as shown in Table 6. Each rule has a score that is fixed after a series of experiments that took into account previous work (Abolohom and Omar, 2017). Each candidate was given a score for each rule, and the one with the highest overall score was recommended as

[5] Pro. Ana: Pronominal Anaphora

[6] POSS: Possessive
[7] DEM: Demonstrative
[8] REL: Relative

the best referent. We chose the one that came closest to overcoming the score similarity.

Linguistic rules	Description
Description	A score of 1 is given if an NP is definite and of 0 if not.
Recency	A score of 1 is assigned to the recency (nearest one) NP to the anaphora and 0 if not.
Referential Distance	A score of 2 is assigned to NPs in the previous sentence or two sentences and further than those are given 0.
First Noun Phrases	A score of 1 is issued to the first NP of each sentence and 0 if not.
NPs in the title	A score of 1 is issued to the existing NP in title and 0 if not
Grammatical function	Scores of 1 are given to an NP that has the same morpho-syntactic features as the anaphora and 0 if not.
Frequency of NP in text	A score of 2 is assigned to the most frequent NP in text and 0 if not.

Table 6: The linguistic preferences and their respective Scores.

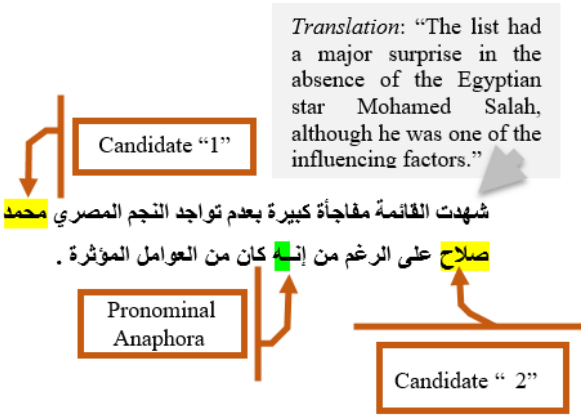


Figure 14: Score similarity (example in pronominal case).

Algorithm: Anaphora Resolution

```
1: input: Anaphora list A, Candidate list C, score function S A, C)
2: initialize score = 0, results = []
3: for t = 1 to N do
4:     for b = 1 to Z do
5:       if Cb [ID] < At [ID] do          ▶ compare if the candidate before or
                                            after the anaphora
6:           score = S (At, Cb)           ▶ S function will calculate every rule
                                            and join the score
7:           results.ADD (At, Cb, score) ▶ store the pairs with their final score
8:       end if
9:       else
10:          break
11:      end else
12:    end for
13: end for
14: results = results.GroupBy(At[ID]).max(score))  ▶ take the pairs with
                                                     maximum score
                                                     for each anaphor
15: output results list with best referent for each anaphor
```

Figure 15: Anaphora resolution heuristics.

5.3 Corpus annotation and verification

This phase aims to annotate the text document using the obtained information from the previous phase, which is a list of pairs of anaphora and their appropriate referent, along with features like Gender, Number, Type, and POS. We used our tool A^{3}T to make the annotation process simpler and fast.

The tool offers a user-friendly interface to linguistic experts, allowing them to check and, if possible, change the connections between anaphora and its referent, resulting in a reliable corpus that can be used in other studies.

More specifically, the interface displays the annotated text in the center, while all of the couples anaphora/candidates are displayed on the right, with the system's chosen couple.

In this case, all the expert has to do is check whether the anaphora tag's number of referent matches the correct one, if not, he may adjust the number of referent to the correct one from the other suggested couples or create a new one if the system doesn't find out the correct antecedent.

In the final part, the tool will add automatically the following tags for the referent and the anaphora: the first will be marked with the <Referent> tag. The remaining elements (anaphora) will be marked with <Anaphor >. We also include the features listed above in each referent and anaphora tag. Finally, the A^{3}T will generate an XML file that contains the text with anaphoric relationship tags as shown in Figure 15.

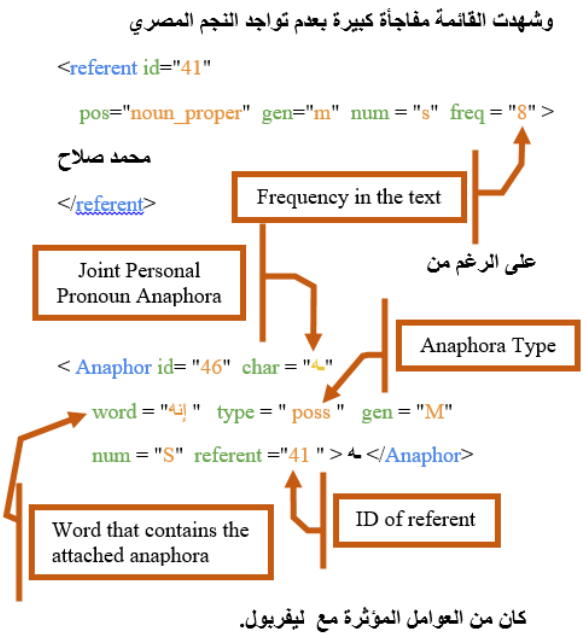


Figure 16: Score similarity (example in pronominal case)

6 Results and Discussion

In our A^{3}T system's testing, we used the "AnATAr" corpora for the evaluation for both anaphora phenomena. We used the standard accuracy metric to calculate the efficiency of the A^{3}T and Table 7 presents the results obtained. We were unaware of any prior works on verbal anaphora, so we tested our work by taking the help of a linguistics specialist and utilizing the same corpora.

Anaphora Type	Corpus	Accuracy achieved
Pronominal anaphora	**AnATAr**	**83.19%**
Verbal anaphora	**AnATAr**	**57.23%**

Table 7 : The result of anaphora resolution system A^{3}T on AnATAr corpora

After analyzing the output of our system, particularly for the verbal anaphora, we found some factors that have influenced our findings. The first factor is the word disambiguation tool limitation that can't in some cases specify the correct meaning of a word such as "سموه" which can act as verb (name; designate) and noun (Highness; grace) or "اخاه" (brother, fraternize). The second factor is the search scope, which could also lead to the best referent being excluded from the list of referents due to being out of scope. In the automatic resolution, the tool rid the references that span multiple sentences but we correct this issue in the expert verification part. The third factor is that the MADAMIRA tool can't recognize composed words like " جمهورية مصر العربية" (Arab Republic of Egypt) or even compound proper names that always occur together like "محمد صلاح" (Mohamed Salah). Finally, in some situations, the voice feature causes a faulty judgment when deciding if the better referent occurs before or after the verb anaphora.

7 Conclusion

Anaphora plays an important role in understanding text and making it coherent. At the same time, it is still a challenging task in the Arabic language due to the complexity of language, the low number of tools, and the lack of linguistic resources. Our present work will make a contribution in the field of linguistic resources for anaphora in the Arabic language and that by providing an annotated corpus that takes into consideration the pronominal and the verbal type. In terms of reducing effort and time consuming during the phase of resolution and annotating, we created A^{3}T, a tool that uses linguistic concepts to identify this phenomenon. With the help of the expert, we are sure that the A^{3}C will be very useful to use in terms of developing intelligence tools that tackle the Arabic anaphora. For the perspectives, our vision will concentrate on the amelioration of the verbal resolution mechanism by using state-of-the-art tools and methods in computational linguistics and at the same time increase the size of the A^{3}C corpus.

References

Abolohom Abdullatif and Omar Nazlia. 2017. A Computational Model for Resolving Arabic Anaphora using Linguistic Criteria. *Indian Journal of Science and Technology*. Volume 10. Issue 3.

Antunes Jamilson , Dueire Lins Rafael , Lima Rinaldo , Oliveira Hilário , Riss Marcelo , Simske Steven. 2018. Automatic cohesive summarization with pronominal anaphora resolution. *Computer Speech & Language*. Volume 52, page (s) 141–164.

Baker Kathryn, Brunner Annelen, Mitamura Teruko, Nyberg Eric, Svoboda Dave and Torrejon, Enrique. 2002. Pronominal Anaphora Resolution in the KANTOO Multilingual Machine Translation System. *Language Technologies Institute, Carnegie Mellon University*.

Beseiso Majdi and Al-Alwani Abdulkareem, 2016. A Coreference Resolution Approach using Morphological Features in Arabic. *International*

Journal of Advanced Computer Science and Applications. Volume 7. Issue 10.

Bouzida Saoussen Mathlouthi and Zribi Chiraz Ben Othmane. 2020. A Generic approach for Pronominal Anaphora and Zero Anaphora resolution in Arabic language. *In proceeding of the 24th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems*.

Cambria Erik. 2016. Affective computing and sentiment analysis. *IEEE Intelligent Systems*. Volume 31(2). Page (s) 102–107, 2016.

Debili Fathi and Hadhémi Achour. 1998. Voyellation automatique de l'arabe. *In Proceedings of the Workshop on Computational Approaches to Semitic Languages.* Page (s). 42–49.

El-Said Nada Aya Nabil Mostafa, Saad Sameh and Al-Ansary Abou El-Magd. 2018. A Syntactic based Approach to Anaphora Resolution in Arabic. *In proceeding of The Eighteenth Conference on Language Engineering (ESOLEC'18)*.

Fotiadou Georgia, Muñoz Ana Pérez and Tsimpli Ianthi Mari. 2020. Anaphora resolution and word order across adulthood: Ageing effects on online listening comprehension. *Glossa: a journal of general linguistics.* Volume 5. Issue 1. Page (s) 1–29.

Haddar Kais.2000. Caractérisation Formelle des Ellipses de la Langue Arabe et Processus de Recouvrement de la Langue Arabe. PhD thesis, University of Tunis.

Hammami Souha, Belguith Lamia, and Ben Hamadou Abdelmajid. 2009. Arabic Anaphora Resolution: Corpora Annotation with Coreferential Links. *The International Arab Journal of Information Technology.* Volume 6, No. 5.

Hamouda Wafya. 2014. Anaphora Resolution for Arabic Machine Translation: A Case Study of Nafs. *Ph.D. dissertation, University of Newcastle Upon Tyne*.

Jarbou Samer Omar. 2018. Time frame as a determinant of accessibility of anaphoric demonstratives in Classical Arabic. *Topics in Linguistics.* Volume 19. Issue 2. Page (s) 57-71.

Madhura Phadke, and Satish Devane. 2019. Pronoun Resolution Task for Multilingual Machine Translation. *In Proceeding of the 5th International Conference on Next Generation Computing Technologies (NGCT-2019)*.

Mathlouthi Bouzid Saoussen, Zribi Chiraz Ben Othmane and Trabelsi Fériel Ben Fraj. 2017. How to combine salience factors for Arabic Pronoun Anaphora Resolution. *In the proceeding of the 4th ACS/IEEE International Conference on Computer Systems and Applications*.

Matysiak Ireneusz. 2007. Information Extraction Systems and Nominal Anaphora Analysis Needs. *In Proceedings of the International Multiconference on Computer Science and Information Technology*. Page (s) 183–192.

Mitkov Ruslan 1999. Anaphora resolution: The state of the art. *In Paper based on the COLING'98/ACL'98 Tutorial on Anaphora Resolution, (University of Wolverhampton)*.

Pasha Arfath , Al-Badrashiny Mohamed , Diab Mona , El Kholy Ahmed , Eskander Ramy , Habash Nizar , Pooleery Manoj , Rambow Owen , Roth Ryan. 2014. Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of Arabic. *In the Proceeding of LREC.* Page (s) 1094–1101.

Seddik Khadiga Mahmoud and Farghaly Ali. 2014. Anaphora Resolution. *Chapter of book, Theory and Applications of Natural Language Processing, Springer.*

Seddik Khadiga Mahmoud and Farghaly.Ali. 2011. Arabic Anaphora Resolution Using Holy Qur'an Text As Corpus. *In proceeding of Arabic Language Technology International Conference (ALTIC).*

Seddik Khadiga, Farghaly Ali and Fahmy Aly Aly. 2015. Arabic Anaphora Resolution: Corpus of the Holy Qur'an Annotated with Anaphoric Information. *International Journal of Computer Applications*. Volume 124. Issue 15.

Sharaf Abdul baquee.,and Atwell Eric. 2012. QurAna. Corpus of the Quran annotated with Pronominal Anaphora. *In the Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12).*

Sukthanker Rhea, Poria Soujanya, Cambria Erik and Thirunavukarasu Ramkumar. 2020. Anaphora and Coreference Resolution: A Review. *Information Fusion*. Volume 59.

Trabelsi Fériel Ben Fraj, Zribi Chiraz Ben Othmane and Mathlouthi Saoussen. 2016. Arabic Anaphora Resolution Using Markov Decision Process. *In the Proceeding 17th International Conference on Intelligent Text Processing and Computational Linguistics*.

Trabelsi fériel ben fraj. 2016. A Novel Approach Based on Reinforcement Learning for Anaphora Resolution. *In the proceeding of the 28th International Business Information Management conference.*

Vinay Annam, Nikhil Koditala and Radhika Mamidi. 2019. Anaphora Resolution in Dialogue Systems for South Asian Languages. *arXiv:1911.09994*.

User Generated Content and Engagement Analysis in Social Media case of Algerian Brands

Aicha Chorana
Laboratoire d'informatique et Mathématiques
Université Amar Telidji Laghouat, Algérie
a.chorana@lagh-univ.dz

Hadda Cherroun
Laboratoire d'informatique et Mathématiques
Université Amar Telidji Laghouat, Algérie
hadda_cherroun@lagh-univ.dz

Abstract

Nowadays, online social media hugely influences individuals' daily lives, companies, institutions, and governments. Analyzing the online social content related to the productivity of any company becomes crucial to manage and supervise its activities and future trends. We investigate the quality of social signals and content related to Algerian products and services to enhance their exploitation and deployment. Our investigation relies on the statistical analysis of social signals and the textual analysis of User-Generated Contents (Posts and Comments). The current work has been done on a sample of more than 50 brands gathering products and services on Facebook with 10K posts and their related comments totaling around 100K.

We measure Users/Brand Engagement Rates (ER) considering reactions and content. We adopted a statistical analysis for the reaction-based measurement. We leveraged an LDA-based Topic Modeling Approach for content-based measurement. Our findings emphasize the significance of the existing social signals and user-generated content in the Algerian context.

1 Introduction

Several companies harness the potential of Online Social Networks (OSN). OSN present an effective communication channel between the company and its customers (Anubha and Shome, 2021; Santoso et al., 2020; Voorveld, 2019). Indeed, these social networks, tremendously, scale up the network effect of standard marketing techniques such as Word-Of-Mouth. Thereby, the emergence of Social Media Marketing(SMM). Indeed, SMM has become an independent field of marketing for which many opportunities have been recognized: i) raising public awareness about companies, ii) product development through community involvement, by analyzing User-Generated Content(UGC) and gathering experience for the future steps (Richter et al., 2011).

The analysis of UGC in social networks has fundamentally reshaped marketing strategies. Users have unlimited freedom to express their opinions through different interactions (e.g. reviews, like, rating...) on web resources. This rich source of social information can be analyzed and exploited to serve several applications in various contexts. In particular, opinion mining and sentiment analysis techniques that have the ability to reveal users' behavior or reaction regarding an item or event. This knowledge represents the bedrock to build an effective content-based recommender system (Zatout et al., 2019).

Users/Brand owners' Engagement analysis and measurement in Arab-world companies seem to be falling behind and show somewhat shy usage. This paper investigates the existence and magnitude of social Media Marketing and explores the nature of both companies' and users' engagement. We also focus on the analysis of textual User-Generated Content in order to present some of their salient features by answering the following questions:

- Are there enough social data on Algerian productivity that can be harnessed to improve Recommender systems applications?
- What are the most used social signals?

Table 1: Details on some User/Brand Engagement studies.

Work	Dataset	Platform	Metrics and Factors
(Pletikosa Cvijikj and Michahelles, 2013a)	100 Brand pages	Facebook	Content type, Media type, posting day and time
(Olczak and Sobczyk, 2013)	10 pages belongs to 4 mobile brands	Facebook	Number of likes, number of shares and posting time.
(Jayasingh and Venkatesh, 2015)	10 169 Posts of 134 Brand pages	Facebook	Number of fans, Customer interaction and Posts type
(Yang et al., 2019)	12K posts of business pages of 500 companies in 6 industries	Facebook	Number of likes and posts' linguistic features, poster characteristics, post context heterogeneity.
(Aldous et al., 2019)	3 M social posts from 53 news organizations	Facebook, Instagram, Twitter, YouTube, and Reddit	shares, external posting, Topic variations

- How are Algerian Brand owners exploiting Social Media?
- How are the users engaging in Social Media Marketing?
- Is social data quality significant to build learning models? Such as Ranking Algerian products, Predicting some economic phenomenon, etc.

The rest of this paper is structured as follows. In the next section, we present some background on Social Signals, concepts of Brand-communities and brand-owners engagement, and how they can be measured. In addition, we review some related work. In Section 3, we describe the followed process in this investigation, starting from the targeted sample of data to the data analytics step. Section 4 is dedicated to reporting results and findings with discussion. We conclude in Section 5.

2 Background and Related Work

In this section we give some preliminaries on the engagement of brand-owners and their brand-communities (users) through social signals and how this engagement can be measured. Then, some related work are discussed.

Engagement in social media, is a multi-faceted complex phenomenon that can be measured by a number of potential approaches (Lalmas et al., 2014; An and Weber, 2018) : i) Self-Reporting Approaches ii) Physiological Approaches and iii) Web Analytic Approaches. This latter refers to the extraction of parameters thought to influence users' engagement, from the digital traces (UGC)left by users while interacting with a website. The most popular UGC on the Web are social signals such as comment, tag, Emotion, Post Message, Reaction, Share, vote, etc Most of these signals are mainly introduced to enable users to express whether they support, recommend or dislike a content (text, image, video, etc.). We can distinguish between social activities' actions and reactions. The actions (e.g., like, share) with counters indicate the rate of interaction with the Web resource. While the reactions, introduced last years, are emotional signals that allow users to interact with posts in a quick way using one of the reactions(Like, Love, Haha, Wow, Sad, and Angry) to react even if the content is difficult to like, as in the case of gloomy news.

Concerning the metrics, for *i) Brand Engagement*, we consider the metrics related to brand's posts: Content and Media Type and their related users interactions. While for *ii) User Engagement*, the considered metrics are: Reaction rates, the relevance of textual generated content regarding the related Brand/service.

Considering the scarcity of investigations on measuring Brand/User engagement for the Algerian Brands, we have narrowed our literature review to some related work from the Western world (Pletikosa Cvijikj and Michahelles, 2013a; Jayasingh and Venkatesh, 2015; Olczak

Table 2: Corpora for Algerian Social Data.

Corpus	Purpose	Corpus Details	Available
Algerian Lexicon (Mataoui et al., 2016)	Sentiment Analysis	206 posts, 7698 comments, Manually collected and annotated	No
ARAACOM (Rahab et al., 2017)	Opinion Mining	Comments on Algerian newspaper	No
(Soumeur et al., 2018)	Sentiment Analysis	20 Algerian brand pages, 25475 annotated comments.	No

and Sobczyk, 2013; Yang et al., 2019; Aldous et al., 2019). Table 1 gives some details on the used metrics and factors. The salient remark is that most used metrics are based on quantitative measurements, namely, the number of reactions and posting times. For news organisations, Aldous et al. (Aldous et al., 2019) defined a more efficient engagement metric based on the user behavior leading to external posting (Spreading content through public sharing to other public networks or platforms). This is performed by means of studying topic variation.

Concerning related work from a Natural Language Processing (NLP) point of view, we can consider that there is a lack of statistical and content analysis of social signals in the Algerian context. For that, we restricted our review to some Algerian online content corpora built for the purpose of content-based analysis, mainly opinion mining and sentiment analysis.

For the sentiment analysis purpose, Mataoui et al. (Mataoui et al., 2016) have built a dataset for Algerian dialect from some main frequented Algerian pages. The chosen social signals are textual (text of posts and comments). They have annotated the dataset manually and they built three Algerian lexicons.

Rahab et al. (Rahab et al., 2017) have built ARAACON (ARAbic Algerian Corpus for Opinion Mining), a corpus of comments collected from online Algerian Arabic journals. These comments are mostly written in Algerian Dialect.

From an economic side, recently, some studies, have investigated the impact of social media on digital businesses. For instance, Graa et al. (Graa et al., 2017) have studied the impact of social media on Algerian purchase behavior. While (Abuljadail and Ha, 2019) have studied the impact of post content type (Hedonic and utilitarian benefits) on the engagement rate. However, these studies are done by means of traditional questionnaire surveys.

In (Soumeur et al., 2018), authors have focused on the specificity of Algerian dialect. They performed a specific pre-processing that improved the data quality. In order to perform sentiment analysis, they used two machine learning models: a Multilayer Layer Perceptron (MLP) neural network and a (Deep) Convolutional Neural Network (CNN).

3 Methodology

In this section, we present an overview of the followed steps, as illustrated in Figure 1. We start by data collection, followed by data preparation (annotation and pre-processing), then data analytics by means of some measured aspects.

3.1 Data collection

Considering the scarcity of datasets on Algerian social signals related to brands and their communities. We have been constrained to collect a sample that encompasses the most powerful and well known brands/services and industrial companies in Algeria. In addition to their visibility on Social Media. The dataset categorizes the collected Brands and Services according to their topic of interest.

Following a similar recipe to the one suggested by authors in (Bougrine et al., 2017). The sample dataset has been collected by following these stages:

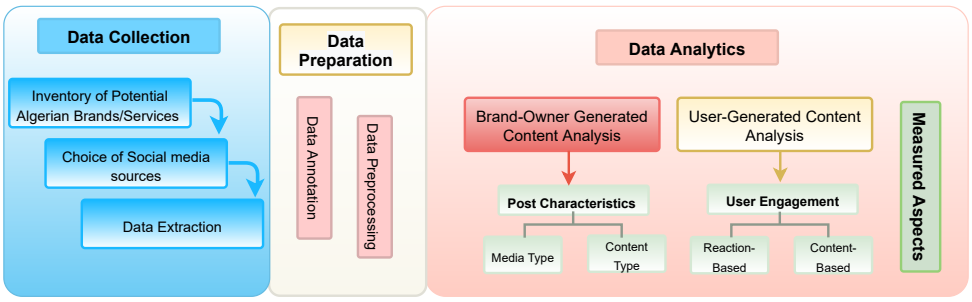


Figure 1: Methodology Overview

Table 3: Details on Chosen Brands and Services.

Category	Subcategory	#	Illustration	#Post	#Comment
Brand	Appliance	5	Condor Electronics, ENIEM, Cobra Electronics, ENIE, StartLight	1 247	21 786
	Beauty/Hygiene	4	Awane,Bimbies,finessecepro,Venus	577	
	Beverage	6	CAFE-Boukhari, Aroma-Café, Rouiba-Jus, Vita-Jus, Cevital-boissons, Ngaous	1 106	15 305
	Dairy	3	Soummam, FALAIT-Tartino, Berber-fromage	423	
	Electronics/Phone	4	LG Algerie, Oppo Algerie, HuaweimobileDZ, SonymobileDZ	937	
	Food	6	Benamor, Safina, Sim, CevitalCulinaire, Jumbo, Bimo	1451	
	Furniture	2	Dz-meuble, Sotrabois menuiserie d'art	199	31 398
	Household Goods	4	Nassah, El-Bahdjadetergents, Aigle, Force Xpress	347	26 576
	Industrial	4	Imetal-SIDER EL-ADJAR, SNVI, TEXALG ex. Sonitex, ENAP	81	256
	Industrial/Auto	2	Renault'DZ, Dacia'DZ		
Services	Accommodation	3	El-Djazair, ElAurassi, El Biar hotel	54	34
	Telecommunication	3	Djezzy'DZ, Mobilis, Ooredoo'DZ	1 960	665 284
	Transportation/Airlines	2	Air Algerie, Tassili Airlines	257	35 274
	Web Service	1	Ouedkniss.com	565	45 539
Total	14	50		9 977	906 705

1. *Inventory of Potential Algerian Brands/Products/Services :*

   First, we have identified Brands/Products/Services that are the most representative of Algerian productivity. This is mainly done using direct expert advice and some social media analytic platforms such as SocialBakers[1]. This step leads to a preliminary list of Brands and services.

2. *Inventory of Potential social Media sources*:

   we have identified the common social media platforms used by communities in concerns. Indeed, depending on their culture and preferences, some communities show preferences of some social media over others. For example, in the time span of this study Algerian users are less interested in *Instagram* or *Snapchat* compared to Middle Est and Gulf communities. In fact, they commonly use Facebook and YouTube[2]. These statistics show that from the period between January and November 2017(the period of our dataset collection), Facebook represents the most used social media platform with 75.94% followed by Youtube and Twitter with only 11.37% and 8.28% respectively.

3. *Extraction Process*

   In order to avoid collecting useless data. This step is achieved in two stages: (i)*Providing Lists*: We define the main keywords that can help automatically search targeted lists. When such lists are

[1] www.socialbakers.com : social media analytics platform.

[2] http://gs.statcounter.com/#social_media-DZ-monthly-201601-201701-bar

established, a first filtering is performed to keep only the potential suitable data. It helps to enlarge our *Brand-list* by Brands that are well visible via Social networks (i.e. well ranked) but not considered by experts as a powerful Brand/Service. (ii)Downloading Data: in this step, we use customized scripts, and Facebook Graph API to scrape the data.

3.2 Data Annotation & Cleaning

We have prepared the data following two step, namely, annotation and cleaning. We manually annotated users' comments according to their:

- *Relevance* : we have considered two classes. Relevant: that says that the topic comments have relation with the targeted post and Irrelevant: which does not have any relation with the related post.
- *Polarity* : Positive, Negative or Neutral.
- *Language distribution and used scripts*: We have considered the most used languages for the Algerian community which are Modern Standard Arabic (MSA), the first and second foreign languages (French and English), and the Algerian Dialect as the common communicated language in the community. In fact, for each comment, we considered the ratio of words by language.

For the purpose of the textual content analysis, we adopted the following data-cleaning steps for all comments in our dataset. First, we remove all photos, stickers, and punctuations, keeping only textual data. Then, we remove stop words (Arabic and French stop words). After that we apply tokenization. We also remove emojis in a second round of cleaning the data.

3.3 Data Analytics & Measured Aspects

In order to investigate the nature and rates of both users and brands' owners engagements, we adopted two types of analysis considering User Generated Content *UGC* and Brand Generated Content *BGC* respectively. In what follows, we demonstrate the considered metrics for both types.

3.3.1 UGC analysis

We addressed user engagement in two ways. One relies on statistical reaction-based analysis, where Engagement Rates consider simple metrics like the number of shares, comments, and reactions) (Pletikosa Cvijikj and Michahelles, 2013b; Perreault and Mosconi, 2018). The second metric relies on content analysis (linguistic features, comments' text analysis) where we deploy some (NLP) techniques to measure the quality and rate of the engagement. These techniques include applying Topic Modeling on comments using Latent Dirichlet Allocation model(LDA) 4.3) (Blei et al., 2003) .

Furthermore, we measure the user engagement rate based on content analysis for post/brand using the relevance of users' comments regarding the post content. Thus, we suggest the following formulas:

First, the content-based engagement rate with a specific post $CPER$ (1) metric.

$$CPER = \frac{RCP}{NCP} \quad (1)$$

Where RCP and NCP are the number of relevant comments per post and the total number of comments per post, respectively. We rely on Topic Modeling of comments' to achieve such goal(see Section 4.3).

Second, we measure the user engagement rate with a brand $CBER$ 2 as an average of the total number of user engagement rate for all the posts of a specific brand $CPER$ (1)

$$CBER = (\sum_{All_{Posts}} CPER)/NP \quad (2)$$

Where NP is the total number of posts per Brand.

3.3.2 BGC analysis

We perform an analysis related to post characteristics where we have considered:

- Content Type (CT) : we considered three classes: Information *Info.* about product/service, remuneration *Renum.* where competitions with rewards and offers are proposed, entertainment *Enter.* any pleasant and hedonic content and *other*.

- Media Type (MT): status, photo, video, link or event. Some media types keep the user more engaged like videos.

Another fundamental metric called *Post Engagement Rate PER* is considered. According to Facebook this metric has different ways of measuring. Bellow, the details about two of them :

$$PER = \frac{R + C + S}{f} * 100 \quad (3)$$

$$PER' = \frac{R + C + S}{Reach} * 100 \quad (4)$$

Where R is the total number of posts' reactions and C is the total number of posts' comments, S is the total number of posts' shares, f is the total number of followers on the day of posting. Although, the second formula gives a more accurate result than the first one, because it uses the $Reach$ metric which is considered as a private data (visible only to the platform and the page owners). And it can't be applicable by simple users. Thus, we will only use the first Formula for this study.

4 Results and Discussion

As reported in Figure 3, the resulted chosen sample consists of 50 brands/Services pages with 9977 posts. It is worth mentioning that we have extracted posts with their related information that might be essential for our study like the number of comments, shares, and reactions. We have also scraped all posts' comments for 12 pages that belongs to different subcategories and their related user interactions. In total, we obtain around 900K comments.

In addition, to fairly assess these statistics, we have compared them to the 50 first well engaged world Brands as a baseline dataset with a similar distribution of Brands. This latter is collected from the Website *Ranking the Brands* [3]. This Website presents statistics about the world most engaged brands.The data we used from this source was bound to the same period of collection of our Algerian Facebook pages data.

For some other metrics, we have relied on SocialBakers studies and those of Buffer and Buzzsumo [4] which is considered as one of the largest studies, where they analyzed more than 43 millions Facebook posts from the top 20 k brands in the world [5].

In order to interpret the results, we have chosen to separately analyse them on both sides : Users and Brands' Owners.

4.1 Brand's Owner Engagement

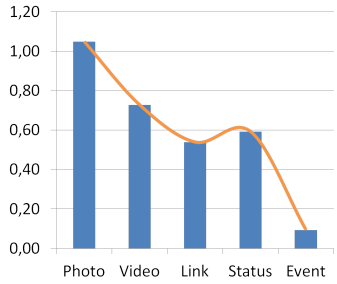
Concerning the engagement of Brands' Owners through their pages. We have examined three facts: the used media type (MT) in posts, content type (CT) and the post engagement rate metric (PER).

Figure 3a reports the distribution of all posts according to their media type (Event, Link, Photo, Status, Video) currently allowed by Facebook. While it is known that the richest media is Video as it describes the product or service better than a photo. We observe that the most used media by our chosen Brands is the "Photo" with more than 85% while only 8.5% of posts deploy videos. In addition, video posts are less used on the Algerian Brand/Service posts counter to the world baseline one with 46%. Furthermore, Event Type is the less used.

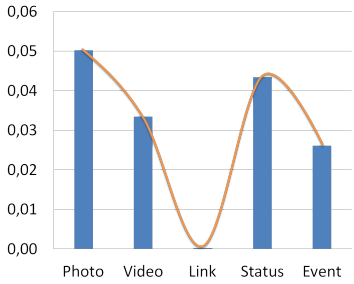
Figure 2 reports the average post Engagement Rates in term of MT for both Algerian's and the worlds' Brands/Services. According to these comparative results, we notice that Photos, which are the most used MT in Algerian brand/service posts, gives them the highest ER compared to others. We notice a considerable ER for videos and status. Even though they are less used.

Comparing the curve of Algerian ER in Figure 2a and the world baseline one in Figure 2b,

[3] https://www.rankingthebrands.com"
[4] https://buzzsumo.com/
[5] https://buffer.com/7



(a) Average of ER received by Algerian Brands/Services



(b) Average of ER received by World Brands/Services

Figure 2: Distribution of Posts & ER by MT (Algeria vs. World)

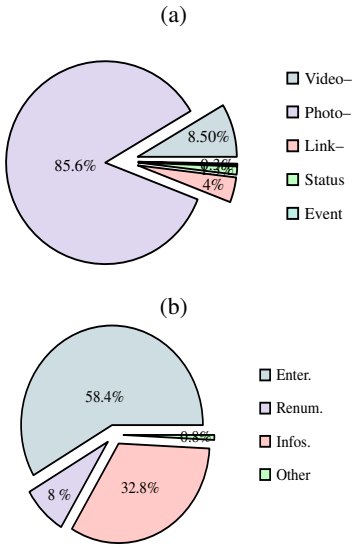


Figure 3: Distribution of Posts by (a) Media Type, (b) Content Type.

we observe that link MT gives a considerable ER for the Algerian sample, while this is not the case for the world baseline sample, counter to the event MT which gives high ER for worlds' Brand/Service posts and low ER for the Algerian ones.

Figure 3b reports the distribution of all posts according to the used Content type. These results show that most of Algerian Brand/Service posts have Entertainment Content Type, and just 8% of them have Remuneration Content Type. While we notice that information posts are the most published posts in the World baseline brands/services (60%) while they are less considered in the Algerian ones.

Comparing the curves of the worlds' ones in Figure 4b and the Algerian's ones in Figure 4a in terms of post ER according to the CT, we notice that they have the same magnitude. In fact, the remuneration is the most attractive CT followed by Information Content Type for both Algerians' and worlds' posts samples.

By comparing the results reported in Figure 3a with those in Figure 4a, we notice that the most used CT is "Entertainment". However, the "Remuneration" and the "Information" ones bring higher Post ER than those of "Entertainment" in Algerian brand/service posts.

In summary, the engagement of Algerian brands' owners is quite significant.

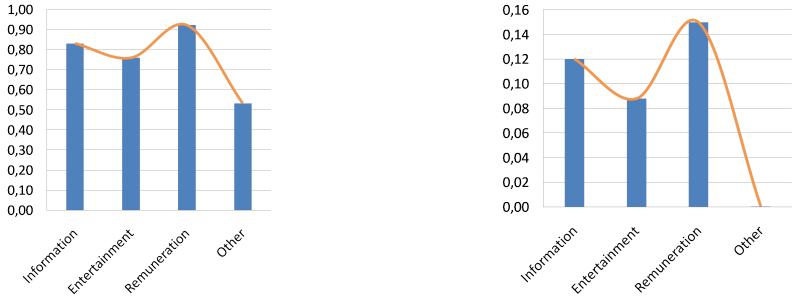
4.2 User' Engagement

For the user side involvement in social marketing, we have analyzed their interest through the quantitative and qualitative measure of interactions. In fact, a users' (potential consumer) comments can provide a better feedback and more information.

Figure 5a and Figure 5b report the distribution of users' interactions by type of deployed Social Signal and the distribution of emotional reactions, respectively.

We observe that users mainly use Reactions (more than 73%), while comments are used with a distribution of 23%. However, users are less active on sharing action. Even though sharing is considered as a deep level of engagement (Aldous et al., 2019).

A more fine analysis of emotional reactions, shows (Figure 5b) that Algerian users are less used to emotional reactions. This latter can be explained by the fact that the emotional reactions has been just introduced by Facebook



(a) Average of ER received by Algerian Brands/Services

(b) Average of ER received by World Brands/Services

Figure 4: Distribution of Posts & ER by CT (Algeria vs. World)

at the time of collection of our dataset.

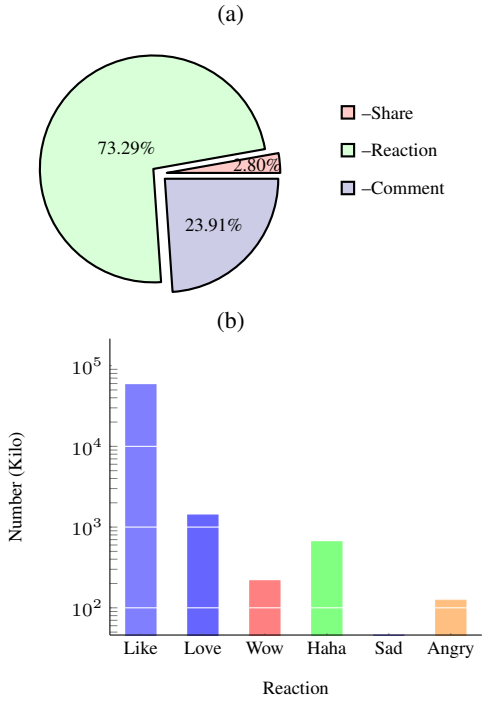


Figure 5: Distribution by type of (a) Social Signal (b) Emotional Reactions

4.3 User's content-based engagement

In addition to the previous statistical analysis of the collected data. We performed some textual data analytics using Topic Modeling. The aim of Topic Modeling here is to discover the notable topics discussed in comments and quantify to what extent users are engaged with a brand/service content. Differently put, Topic Modeling investigates the comments' relevance by checking similarities between the top words of the topic describing a brand/service and the brand/service related comments. t

In the current work, we used Latent Dirichlet Allocation (LDA) method (Blei et al., 2003) that exhibits smooth scalability applied on large textual corpora.

Topic modeling here serves as an aggregation tool to discover the latent discussed topics in users' comments.

By examining the resulting topics presented in Table 4, we can clearly notice that they represent most of the Brand/service categories covered by this study.

For instance, most of the words in *Topic* 0 are related to the Algerian Airlines company (i.e, most of the aggregated words are from *AirAlgerie* Facebook page. For example, The words: الله , *alger, paris, billet, bonne, prix, vol, air,* are respectively in English: *Allah* (the God), *Algiers* (the capital of Algeria), *Paris* (the capital of France), *Ticket* (the flight ticket), *good or nice* (we assume that it is a typo in writing *"bon"* from *"bon voyage", Nice trip*), *Price, Flight* and *Air* from the company name *AirAlgerie*.

Another example, is *Topic 4* which is about the two biggest telecommunication companies in Algeria: *Ooredoo* and *Mobilis*. Most of the words indicating that users' comments are about these companies services. Some of these words are: جيغا، موبيليس ، تم ، صالح ،دج ,*ooredoo, prix , max , win, da*. The English translation of the previous words, respectively is: *Dzd* (the Algerian currency), *valid*(most probably about the phone credit), *done*(a com-

Table 4: Example of generated topwords of a topic using LDA model.

Topic Number	Top words	Top words(English)	Description
Topic 0	air, الله , alger, paris, billet, bonne, algerie, bien, prix, share, site, mohamed, saha, da, vol, ايام	air,Allah(God), Algiers, Paris, ticket, good Algeria, price, share, site, Mohamed, okay, Dzd, flight, days	this represents the airlines Facebook page
Topic 1	hada, aroma , chaba , Mohamed , قهوة, top , prix , السحب , روعة , اروما، الله ، تمانشاء	This,aroma,nice, Mohamed,coffee,top,price,lottery,fantastic,aroma ,God willing, done	This is the Aroma coffee Facebook page
Topic 2	merci, lg, participe, chance, produits, force, prix, aigle, bien, top, express, lave, xpress,مبروك، الجزائر، الحي ، الله ، شكرا	Thanks, LG, participate, luck, product, force, price, Aigle, good, top , express, wash, xpress, congratulation, LG, Algeria, Allah, thanks	The cleaning stuff topic for AigleGroupt, and Force Express Brands
Topic 3	ooredoo, prix , max , win, da, go ، اوكتيه ، دج، ساعة، صالح ، موبيليس، وين، تم ، جيغا، شهر، الرسوم، الجواب	Ooredoo, price, max , win, Dzd, GO, valid, an hour, Dzd, octet, , the answer, fee, month, Giga, done,la win, Mobilis	This topic is about telecommunication companies Ooredoo and Mobilis

mon word used in the Algerian content indicating that someone has seen the post), *Mobilis* (the first company name), *giga* (or *gigaoctet*: a unit for computer memory), *Ooredoo* (the second company name), *Price, Max* and *Win*(plans names offered by the company), *Dzd*(the abbreviation of tha algerian curruncy in french; *Dinar Algerien*).

We can also highlight the repetition of the word "اللّه" in most of the topics' word set. A possible interpretation could be that it is a very common for Algerian users to *overuse* the expression "إن شاء اللّه ، انشاء الله" English: *God willing* with all its variations, in their daily life, thus in their online comments .

4.4 Used Languages and Scripts

Concerning the used languages, Figure 6 illustrates their distribution in users' comments. It shows that French is the most used language, followed by MSA Arabic with 35.7% then Algerian dialect with 19.4%. The category "Other" includes Tamazight, Espagnol, Korean and others.

While for the distribution of used Scripts, we have classified comments according to Arabic scripts, Latin scripts, mixed Arabic, Latin scripts and other scripting sets like emoticons or numbers. We observe that 65% of textual comments use Latin characters while 25% of them use Arabic ones.

Moreover, we have analyzed the usage of Emojis where we have reported that just 7% of Algerian customers' comments are using emoticons while 93% of them are textual.

These findings could help brands in term of marketing get closer to their customers through understanding their language. In addition, these findings help any Natural Language Processing research problem to leverage such linguistic features.

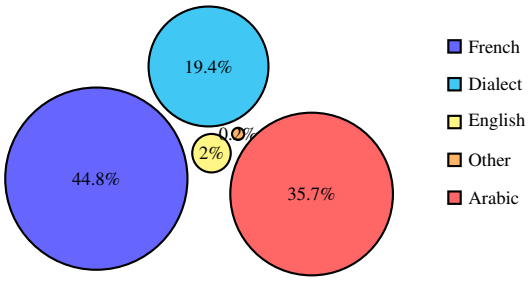


Figure 6: Distribution of used languages in User comments

5 Conclusion

We proposed an analytical study based on statistical and textual analysis of User/Brand Generated Content on social media. We investigated the level of Users/Brands engagement by two means: using the common User engagement formulas in the literature for reactions case. And we suggested a content-based user engagement approach based on LDA Topic Modeling method. Our finding highlight the quantitative and qualitative significance of the existing social signals in the Algerian productivity context. Which can efficiently help

Brands' owners to improve their productivity and online marketing strategy. In the future, we intend to normalise the whole content of the data set by, automatically detecting the language and translate it to Modern Standard Arabic. We will also investigate the effectiveness of Topic Modeling on assessing users' engagement.

References

Mohammad Abuljadail and Louisa Ha. 2019. Engagement and brand loyalty through social capital in social media. *International Journal of Internet Marketing and Advertising*, 13(3):197–217.

Kholoud Khalil Aldous, Jisun An, and Bernard J. Jansen. 2019. View, like, comment, post: Analyzing user engagement by topic at 4 levels across 5 social media platforms for 53 news organizations. *Proceedings of the International AAAI Conference on Web and Social Media*, 13(01):47–57.

Jisun An and Ingmar Weber. 2018. *Diversity in Online Advertising: A Case Study of 69 Brands on Social Media*, pages 38–53.

Anubha and Samik Shome. 2021. Customer engagement and advertising effectiveness: A moderated mediating analysis. *Journal of Internet Commerce*, 0(0):1–41.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

Soumia Bougrine, Aicha Chorana, Abdallah Lakhdari, and Hadda Cherroun. 2017. Toward a web-based speech corpus for algerian dialectal arabic varieties. In *Proceedings of the Third Arabic Natural Language Processing Workshop*, pages 138–146.

Amel Graa, Soumia Abdelhak, and Hayat Baraka. 2017. Les médias sociaux: L'étude de l'effet médiateur de la confiance et l'utilité perçue des commentaires dans le contexte algérien. *International Journal of Marketing, Communication and New Media*, 0(2).

Sudarsan Jayasingh and Rajagopalan Venkatesh. 2015. Customer engagement factors in facebook brand pages. *Asian Social Science*, 11(26):19.

Mounia Lalmas, Heather O'Brien, and Elad Yom-Tov. 2014. Measuring user engagement. *Synthesis lectures on information concepts, retrieval, and services*, 6(4):1–132.

Mhamed Mataoui, Omar Zelmati, and Madiha Boumechache. 2016. A Proposed Lexicon-Based Sentiment Analysis Approach for the Vernacular Algerian Arabic. *Research in Computing Science*, 110:55–70.

Artur Bernard Olczak and Rita Karolina Sobczyk. 2013. Brand Engagement on Facebook Based on Mobile Phone Operators' Activity Within Four European Countries. *Studia Ekonomiczne*, 150:97–108.

Marie-Catherine Perreault and Elaine Mosconi. 2018. Social Media Engagement: Content Strategy and Metrics Research Opportunities.

Irena Pletikosa Cvijikj and Florian Michahelles. 2013a. Online engagement factors on facebook brand pages. *Social Network Analysis and Mining*, 3(4):843–861.

Irena Pletikosa Cvijikj and Florian Michahelles. 2013b. Online engagement factors on Facebook brand pages. *Social Network Analysis and Mining*, 3(4):843–861.

Hichem Rahab, Abdelhafid Zitouni, and Mahieddine Djoudi. 2017. Araacom: Arabic algerian corpus for opinion mining. In *Proceedings of the International Conference on Computing for Engineering and Sciences*, ICCES '17, pages 35–39, New York, NY, USA. ACM.

Daniel Richter, Kai Riemer, and Jan vom Brocke. 2011. Internet social networking. *Business Information Systems Engineering*, 3(2):89–101.

Irene Santoso, Malcolm Wright, Giang Trinh, and Mark Avis. 2020. Is digital advertising effective under conditions of low attention? *Journal of Marketing Management*, 36(17-18):1707–1730.

Assia Soumeur, Mheni Mokdadi, Ahmed Guessoum, and Amina Daoud. 2018. Sentiment analysis of users on social networks: overcoming the challenge of the loose usages of the algerian dialect. *Procedia computer science*, 142:26–37.

Hilde A.M. Voorveld. 2019. Brand communication in social media: A research agenda. *Journal of Advertising*, 48(1):14–26.

Mochen Yang, Yuqing Ren, and Gediminas Adomavicius. 2019. Understanding user-generated content and customer engagement on facebook business pages. *Information Systems Research*, 30(3):839–855.

Chayma Zatout, Ahmed Guessoum, Chemseddine Neche, and Amina Daoud. 2019. Prediction of the engagement rate on algerian dialect facebook pages. *Recent Advances in NLP: The Case of Arabic Language*, 874:163.

Automatic Assessment of Speaking Skills Using Aural and Textual Information

Sofia Eleftheriou
Institute of Informatics & Telecommunications,
NCSR "Demokritos"
Athens, Greece

Panagiotis Koromilas
Institute of Informatics & Telecommunications,
NCSR "Demokritos"
Athens, Greece

Theodoros Giannakopoulos
Institute of Informatics & Telecommunications,
NCSR "Demokritos"
Athens, Greece

sofiaeleftheriou13@gmail.com
{pakoromilas, tyianak}@iit.demokritos.gr

Abstract

In this work we propose a multimodal speech analytics framework for automatically assessing the quality of a public speaker's capabilities. For this purpose, we present the Public Speaking Quality (PuSQ) dataset, a new publicly available data collection that contains speeches from various speakers, along with respective annotations of how are these speeches perceived by the audience in terms of two labels namely: "expressiveness" and overall "enjoyment" (i.e. if the listener enjoys the speech as a whole). Towards this end, several annotators have been asked to provide their input for each speech recording and inter-annotator agreement is taken into account in the final ground truth generation. In addition, we present a multimodal classifier that takes into account both audio and text information and predicts the overall recordings' label with regards to its speech quality (in terms of the two aforementioned labels). To this end, we adopt a hierarchical approach according to which we first analyze the speech signal in a segment-basis (50ms of audio and sentences of text) to extract emotions from both text and audio and then aggregate these decisions for the whole recording, while adding some high-level speaking style characteristics to produce the overall representation that is used by the final classifier.

1 Introduction

Public speaking (also called oratory or oration) is the act of giving speech face to face to live audience. However, due to the evolution of public speaking, it is lately viewed as any form of speaking (formally and informally) between an audience and the speaker. Traditionally, public speaking was considered to be a part of the art of persuasion. The act can accomplish particular purposes including information, persuasion, and entertainment. Additionally, differing methods, structures, and rules can be utilized according to the speaking situation.

Currently, technology continues to transform the art of public speaking through newly available techniques such as videoconferencing, multimedia presentations, and other nontraditional forms. Knowing when speech is most effective and how it is done properly are key to understanding the importance of it.

While most current methods for evaluating speech performance attend to both verbal and nonverbal aspects, almost all existing assessments in practice require human rating (Ward, 2013; Schreiber et al., 2012b; Carlson and Smith-Howell, 1995). Due to the obvious need to use our speech in our daily lives, its evaluation and its improvement is also very important. This evaluation becomes easier and faster if it is performed by an automated process that mostly uses machine learning methodologies.

Speech is everywhere and the way we speak is just as important as what we say. Therefore, multimodal speech analytics (using text and audio) is an important process that can be applied not only to assess public speakers speech quality, but also in other speech-related fields of application such as the identification of learning disabilities related to speech (dyslexia, autism), the analytics of call center data and the speech-based assessment of psychological and psychiatric conditions. The proposed pipeline for assessing the public speech quality can be adopted in such applications, as soon as respective ground truth have been made available for training the supervised models.

The related works in assessing public speakers' skills is very limited and usually focuses in two specific tasks, namely learning analytics and persuasive analysis. In particular, (Chen et al., 2016) focuses on the design of a multimodal automated assessment framework for public speaking skills an-

alytics, which is based on the public speaking competence rubric (PSCR)(Schreiber et al., 2012a) for scoring and uses both audiovisual and textual features. With regards to the persuasiveness prediction application domain, a widely used dataset, named Persuasive Opinion Multimedia (POM) (Park et al., 2014) has been created, which contain multiple communication modalities (audio, text and visual). A deep learning approach for this task that is evaluated on POM is presented by (Nojavanasghari et al., 2016), where the authors design a deep multimodal fusion architecture, that has the ability to combine signals from the visual, acoustic, and text modalities effectively.

However, the aforementioned methodologies do not address the task of assessing the public speakers' skills in a generalized manner. This paper proposes an ML framework for classifying long speech recordings in terms of: (a) overall speech "expressiveness", as perceived by the audience and (b) perceived "enjoyment", i.e., how much the listeners enjoyed each speech recording. Towards this end, we demonstrate a Python open-source library that utilizes segment-level (size of 20ms to 50ms) audio and text classifiers related to emotional and speaking style attributes. The final recording-level decision is extracted by a long-term classifier that is based on feature aggregates of the segment-level decisions. Apart from the open-source library, we present an openly available dataset of real-world recordings, annotated in terms of perceived expressiveness and enjoyment. Extensive experimental results prove that the proposed ML framework can discriminate between positive and negative speech samples, despite the simplicity of the baseline segment-level classifiers.

The paper is organized as follows: Section 2 shows the conceptual diagram of the proposed methodology, Section 3 presents the segment-level audio and text classifiers related to emotional attributes, Section 4 introduces the aggregation of class posteriors among with some high-level features that are calculated across the entire recording, Section 5 refers to the newly constructed Public Speaking Quality (PuSQ) dataset, Section 6 is responsible for the reporting of the implemented experiments and Section 7 sets out the final conclusions.

2 System overview

The system architecture developed in the context of speech quality assessment is divided into two parts: segment-level analysis and recording-level analysis. In the first, we break the information (audio or text) into temporal segments and use segment-classifiers related to emotional content. In the second, we aggregate the previously produced class posteriors and combine with high-level features that characterize the overall speaking style. This rationale is followed for both textual and audio modalities and the final decisions can either be used independently or combined in the final recording-level classifiers. The conceptual diagram of the proposed system architecture is shown in Figure 1.

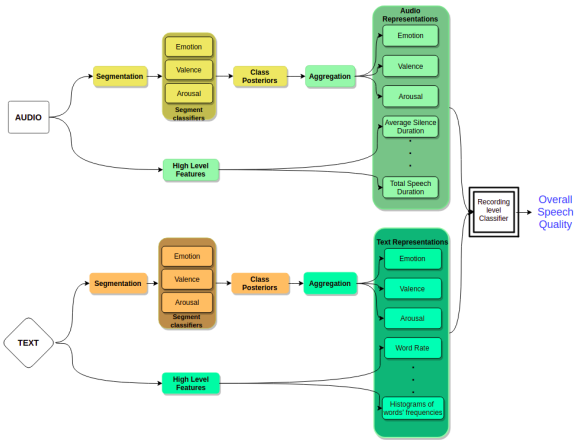


Figure 1: System Architecture

3 Segment-Level Analysis

3.1 Audio Analysis

The audio recording is split into segments and for each segment audio feature extraction is performed and segment classifiers are applied to produce a series of emotion-specific classification decisions, which are then aggregated for the whole recording's classification in terms of overall speech quality. The goal of this section is to describe this segment-level process.

3.1.1 Segmentation and Feature Extraction

For each 3s segment, a short-term window process is followed, i.e. the segment is further split into short-term windows (frames) of 50 ms long with a step of 50 ms (no overlapping). For each short-term window, a series of hand-crafted audio features is extracted, that have been widely used in speech classification tasks. These *low-level audio*

features are: Zero-crossing rate, Energy, Energy entropy, Spectral centroid, Spectral spread, Spectral Entropy, Spectral Flux, Spectral Rolloff, the first 13 MFCCs , the Chroma Vector (12-dimensional) and Chroma Standard Deviation. All these features summing up to 34 in total. We further add the deltas of these features, i.e. the difference between each feature in the current short-term window and the value it had in the immediately preceding short-term window. So we end up with 34 such derivatives (deltas), so 68 features in total for each frame.

Then, for each segment, we extract two feature statistics for each sequence of short-term features described above. The statistics are: the average μ and the standard deviation σ^2 of the respective short-term feature sequences, among the whole 3s segment. Therefore, each segment is now represented by 134 (68×2) feature statistics. To extract this representation, we used the Pyaudioanalysis (Giannakopoulos, 2015) open source Python library.

3.1.2 Speech Segment Classifiers

As described above, each speech segment is represented by 134 audio feature statistics. Then, we have selected to train segment-level classifiers related to the underlying *emotions*, because emotions are strongly associated to the overall speaking style of a public speaker and therefore to the respective speech quality. Towards this end, we have adopted both categorical and dimensional Speech Emotion Recognition annotations (attributes).

The categorical attributes consist of some basic classes of emotions. According to Ortony and Turner (1990), basic emotions are often the primitive building blocks of other non-essential emotions, which are considered variations, or mixtures of basic emotions. In Ekman (1992) six basic emotions are suggested, based on the analysis of facial expressions (anger, disgust, fear, joy, sadness and surprise). We choose to use the 4 basic emotions: anger, sadness, neutral and happiness, as provided by the, widely used in the literature (Koromilas and Giannakopoulos, 2021), processed version of the IEMOCAP dataset.

The main disadvantage of the categorical model is that it has a lower resolution than the, associated with continuous values, dimensional model because it uses categories. The true number of individual emotions and their tones encountered in different types of communication are much richer than the limited number of emotion categories in the model. The smaller the number of classes in the categorical model,the greater the simplification of the description of emotions (Grekow, 2018). That is why dimensional attributes are also widely used in emotion recognition. These representations allocate emotions in dimensional spaces that can mainly capture the similarities and differences between them. Wundt and Judd (1897), proposed the first dimensional model by disassembling the space of emotions along three axes, namely: *valence* (positive-negative), *arousal* (calm-excitement) and intensity (intensity-relaxation). In this work, a usual scheme in the literature (Le et al., 2017) has been adopted with the use of a discretized version of the first two axes (valence and arousal) with the following classes: negative, neutral, positive for valence and high, neutral, low for arousal.

For the training and the evaluation of the three above-mentioned models (emotions, valence, arousal), we use 5 open-source speech emotion datasets, as well as a proprietary dataset that had been created by the authors. The open source datasets are: Emovo (Costantini et al., 2014), EmoDB (Burkhardt et al., 2005), Savee (Jackson and ul haq, 2011), Ravdess (Livingstone and Russo, 2018) and IEMOCAP (Busso et al., 2008). The 6th dataset is named Emotion Speech Movies and contains audio files from movie scenes that are divided into 5 emotional classes.

Some of the aforementioned datasets contain more classes of emotions, therefore we only used the samples corresponding to the classes of interest. Also ”excitement” was merged with happiness, since they are quite related expressions. For the valence and arousal tasks, one can observe that the only dataset which contains corresponding labels is IEMOCAP. These labels are continuous values and as we address classification problems, we divide these value ranges into three identical intervals. For valence: the samples with value in the range [1,2.5) are considered to be ”negative”, the samples with value in the range [2.5,4) are considered to be ”neutral” and the samples with values in the range [4,5.5] are considered to be ”positive”. For arousal: the samples with value in the range [1,2.53) are considered to be ”low”, the samples with value in the range [2.3,3.6) are considered to be ”neutral” and the samples with values in the range [3.6,5] are considered to be ”high”. For all other data sets that do not contain valence and arousal tags, we distribute the emotion tags in the above 6 valence

Classification Task	Dataset						Average
	IEMOCAP	Savee	Emovo	Emo-db	Ravdess	EmotionSpeechMovies	
Emotion	79.7	71.2	75.5	80.6	67	50.4	**70.7**
Valence	52.9	62.3	59.5	76	63.9	53.2	**61.3**
Arousal	60.3	75.3	79.9	81.9	68.3	64.1	**70**

Table 1: Inner-dataset Evaluation of Audio Models

Classification Task	Test Dataset						Average
	IEMOCAP	Savee	Emovo	Emo-db	Ravdess	EmotionSpeechMovies	
Emotion	39	36	45.5	57.6	29.8	36.6	**40.8**
Valence	39.7	37.9	32.7	37	26.3	42	**35.9**
Arousal	40.1	41.8	40.3	51.1	38.4	38.6	**41.7**

Table 2: Cross-dataset Evaluation of Audio Models

and arousal classes based on the circumplex model shown in Figure 2.

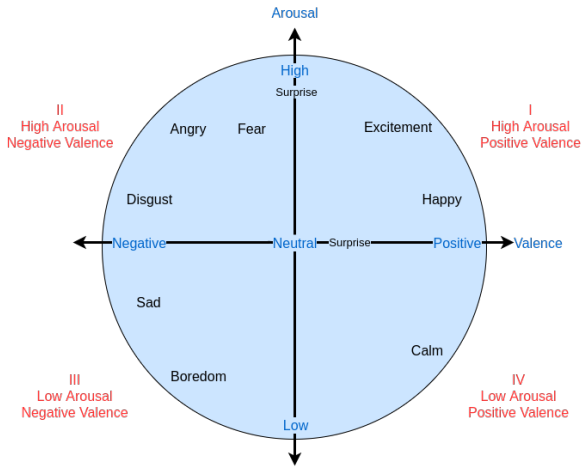


Figure 2: Distribution of Emotions in Circumplex Space

3.1.3 Segment Classifiers Training and Evaluation

As described above, each audio segment is represented by a 134-dimensional feature vector, while three classification tasks have been defined: emotion, valence and arousal, using 6 different datasets. For these classification tasks, we have first performed a separate evaluation pipeline for each different dataset. The results of this *inner-dataset evaluation* procedure is shown in Table 1. The classification algorithm used for this experimentation was the SVM with RBF kernel, which outperformed all traditional classification methods (decision trees, random forests and k-nearest-neighbors).

Apart from these dataset-dependent results we have also conducted experiments using a "leave-one-dataset-out" rationale, in order to perform a *cross-dataset evaluation*. In both evaluations a repeated cross validation approach has been adopted with a 80% - 20% train-test data split and 100 iterations. The cross-dataset evaluation results for the best classifier (again SVM with RBF kernel) are shown in Table 2. All the metrics shown are f1 macro-averaged.

Comparing the results of the above two tables, we can observe that in the cross-dataset evaluation

Classification Task	Merged Dataset		
	Xgboost	CNN	SVM
Emotion	60.4	60.5	64.4 (+/-0.9)
Valence	51.7	52.6	55.2 (+/-1.2)
Arousal	64.2	69.3	66.8 (+/-1.1)

Table 3: Merged-dataset Evaluation of Audio Models

(Table 2), the results are just slightly better than random guess on average (25%). This implies that the problem that these models are called upon to solve is directly dependent on the specific sub-domain. By "sub-domain", we mean the set of context conditions and types of speakers in each dataset. Cross-domain adaptation is one of the most common difficulties in speech emotion recognition. And it is beyond the scope of this paper to handle this issue. The most straightforward way for our scope (which is to create segment classifiers that can be used as feature extractors for the recording-level decisions), is to simply train our segment models on a *merged emotional dataset*. The results of cross-validation on this merged dataset are presented in Table 3. The machine learning algorithm used for this type of experiments is SVM with RBF kernel, as well as xgboost (Chen and Guestrin, 2016). In addition, to manage the imbalance of datasets and to increase the performance, we also used a sampler SMOTE-Tomek (Wang et al., 2019), a StandardScaler, a VarianceThreshold with threshold set equal to zero and a PCA (Kabari and Nwamae, 2019). During the training, a gridsearch was applied which uses RepeatedStratifiedKFold cross validation with 5 folds. Hyperparameter tuning is performed in three hyperparameters: the number of components of the pca that are maintained, the hyperparameters γ and C of the SVM.

For comparison reasons, we also evaluated the performance of a Convolutional Neural Network (CNNs) with melgrams used as audio features, which is a common approach in Deep Learning for audio classification. Towards this end, the open-source Python library *deep_audio_features* (Theodoros Giannakopoulos, 2020) was used. The CNN has 4 convolutional layers with kernels 5×5, single stride and zero padding, while max pooling of size 2 was used. The output channels (i.e. the

third dimension), for the first layer are 32, for the second 64, for the third 128 and for the forth 256. After the convolution layers, we use 3 linear layers, with the first having an output dimension of 1024, the second 256 and the third equal to the number of classes.

The above results show that the best performance is achieved when using the SVM classifier. CNN is outperforming only for the arousal task, which indicates that more data may be needed for this deep approach to outperform. Of course, more sophisticated approaches could be used also capturing temporal dependencies between features (such as LSTMs or Transformers), but this is to be considered for future work. Finally, we have experimented with speaker independent experiments, by evaluating the SVM classifier on a subset of audio segments of unseen speakers (i.e. speakers whose segments were not available in the training data). This speaker-independent evaluation showed results that were on average 3% worst than the ones appearing on Table 3. This indicates that the speaker independence assumption does not significantly affect the performance for this particular model.

3.2 Text Analysis

3.2.1 Segmentation and Feature extraction

The Speech API provided by Google was used in order to extract textual information from the initial audio signal. The text from the whole recording can be segmented using three different approaches: sentence-level splitting, splitting into windows of predefined number of words or splitting in fixed time windows. In order to train the models described in the next paragraphs, the samples are pre-segmented in sentences.

In Natural Language Processing (NLP), word embedding is a term used to represent words in text analysis, usually in the form of a real valued vector that encodes the meaning of the words so that they can be represented in a joint representation space where the closest words are expected to have a similar meaning (Mikolov et al., 2013). To obtain word embeddings, experimentations with two pre-trained natural language models, ie. FastText (Bojanowski et al., 2016) (trained on data of English Wikipedia) and BERT (Devlin et al., 2018) (trained on data of BooksCorpus (Zhu et al., 2015) and English Wikipedia), were conducted. For the BERT architecture, given a text segment/sentence, the embeddings of the last 4 layers were averaged in order to get a more general representation.

	IEMOCAP		
Classification Task	SVM with FastText Embeddings	XGBOOST with BERT Embeddings	SVM with BERT Embeddings
Emotion	66.5 (+/-1)	63.9 (+/-1.7)	**69.5 (+/-1.4)**
Valence	61.5 (+/-1)	59.4 (+/-0.9)	**63.8 (+/-1)**
Arousal	48.8 (+/-1.1)	48.2 (+/-1)	**51 (+/-1.1)**

Table 4: IEMOCAP Evaluation of Text Models

As the IEMOCAP dataset is the only data collection, from the ones presented in section 3.1.2, that contains transcriptions (textual information), this is the one that will be used for the training/evaluation of proposed models (emotions/valence/arousal).

3.2.2 Segment classifiers training and evaluation

Experiments with both Fasttext and BERT embeddings were conducted on all three classification tasks. For training, an appropriate parameter tuning of a pipeline cosnisting of SMOTETomek (to handle imbalance), StandardScaler, VarianceThreshold, PCA and either SVM with RBF kernel or XGBOOST classifier was held. The evaluation procedure was performed using RepeatedStratifiedKFold validation scheme with 5 folds and 3 repetitions. The macro-averaged f1 score metric is shown in Table 4, where the +/- sign indicates the standard deviation of the metrics across different test folds.

From the listed results it can be clearly seen that *(i)* the use of BERT embeddings results in better performance probably due to stronger word representation power, and *(ii)* the SVM classifier is superior to the XGBOOST for all three classification tasks.

4 Recording-Level Analysis

The overall goal of segment-level analysis is to be used in order to extract recording-level information. Towards this end, we will combine segmented information with high-level features in order to perform a recording analysis.

4.1 Aggregation of Class Posteriors

Segment-classifiers result in three labels associated with emotion, valence and arousal respectively. In order to characterize the whole speech signal, an **aggregation** of the class posteriors across the recording length is performed. For example, the average emotion confidence per label may be: $P(emotion = sad) = 0.3$, $P(emotion =$

$neutral) = 0.4$, $P(emotion = happy) = 0.1$, $P(emotion = angry) = 0.2$

4.2 High Level Features

In order to capture long-term dependencies some high-level features, for both audio and text, need to be calculated across the input signal.

For the aural modality, a voice activity detection is firstly performed using features of the pyAudioAnalysis library (Giannakopoulos, 2015), in order to train in a semi-supervised fashion an SVM classifier so as to detect periods of silence. After identifying the parts of voice and silence in speech, the following high-level features are calculated: average silence duration, silence segment per minute, standard deviation of silence duration, speech ratio and word rate in speech. In order to extract different kinds of silence (ie. inter-word and intra-word), the aforementioned features are calculated for 2 different short-term windows: windows of 0.25 step and 0.5 length and windows of 0.25 step and 1 length respectively, resulting in 10 high-level features for each audio file.

As for the textual modality, the following high-level features are extracted: word rate, unique word rate and 10-bin histogram of word frequencies (frequency must range between 0 to 0.1 in order to filter out non-informative words), resulting in 12 high-level features.

5 Public Speaking Quality Dataset

Speech Quality Assessment is a task of interest in psychology, public speaking, rhetoric and a variety of other related sciences. However, this problem is quite difficult to track with the use of computational approaches. In order to address this need we introduce the Public Speaking Quality (PuSQ) Dataset, a data collection that contains speech audio and text files annotated from human listeners.

5.1 Data Acquisition

For the needs of the data collection process, a web application[1], named RecSurvey was created and the participants could use it through their personal recording set-up (ie. headset or PC microphone). The participants had to firstly fill out their demographic information (age, ethnicity, gender, English fluency etc) and then record themselves either reading some of the 40 predefined English texts (4-5 lines each) of different topics (politics, books, machine learning, etc) or answering some of the 20 general questions such as "What do you like most about your current job?". Here it has to be noted that, although our purpose is to evaluate the quality of free speech, a variety of predefined texts were used in order to have a quantitative control of the result.

The process of data acquisition resulted in a total of 695 recordings/speeches from 42 different individuals, of which 26 were female. In addition, people of different nationalities were considered so as to have a variety of pronunciation and speaking styles.

5.2 Annotation

The annotation process is mandatory in order to create a labeled dataset. Towards this end another web application[2] was created and used so as to annotate the collected speeches based on three indicators:

- **expressiveness:** how active, emotional or passionate the speech is, regardless of its content.
- **ease of following:** the evaluation of verbal clarity, fluency and rate of speech, for the specific content described. It is noted that fluency, clarity and rate can be correlated. For example one speaker, despite speaking fast, may deliver an easy-to-follow speech, while the opposite may hold for another speaker.
- **enjoyment:** defines the listener's / annotator's personal view of whether the speech was exciting, entertaining or motivating.

The marking/annotation in each of the above classes was done using 5 staggered labels, ie. the annotator had to rate each recording in the range from 1 to 5, with 1 being the worst and 5 the best measure.

In total, 14 annotators made 2687 markings by labeling 689 out of the 695 recordings for each of the 3 tasks. Further details on the number of annotations per user are illustrated in Figure 3.

5.3 Annotations Aggregation

Although the labels are distinct (5 labels / values per task), they have a continuous, scalable form as the smaller label (1) corresponds to the worst performance, while the larger label (5) corresponds to

[1] https://github.com/lobracost/RecSurvey

[2] https://github.com/sofiaele/audio_annotator

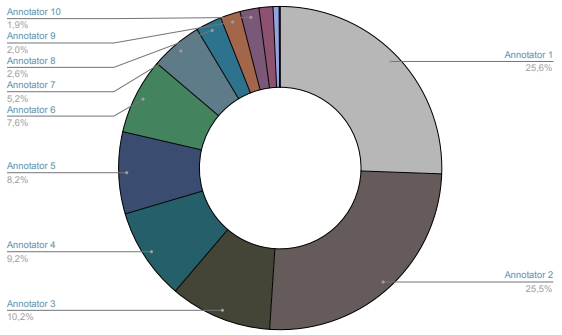


Figure 3: Annotations per User

	Expressiveness			
	Female		Male	
	Positive	Negative	Positve	Negative
Mean Thresholding	$\mu >= 4$	$\mu <= 2$	$\mu >= 3.1$	$\mu <= 2$
Number of samples after Mean Thresholding	72	80	70	67
Deviation Thresholding	$\sigma < 0.75$	$\sigma < 0.75$	$\sigma < 0.75$	$\sigma < 0.75$
Number of samples after Deviation Thresholding	63	71	48	60
Number of samples after Minimum Annotators	52	53	41	50
Average Disagreement	**0.52**		**0.53**	

Table 5: Definition of Expressiveness Dataset

the best one. Therefore, instead of aggregating the annotations of a recording based on majority voting, averaging have been used. More specifically, for each task (expressiveness, ease of following, enjoyment) and for each gender (female, male), 2 classes / labels were exported: one negative and one positive. Here it has to be noted that the gender separation was required since, the annotated quality of speech was biased to favor female speakers and adding the fact that the two genders are easily distinguishable due to different speech sound frequencies, the problem of over-fitting arose.

To aggregate the annotations and produce the binary datasets, the following three filterings have been applied on each sample:

- samples with less than three annotators are excluded
- the mean value of the labels given by different annotators, must either be under a lower or above an upper threshold. Thus, the instance is either labeled as negative/positive or excluded from the dataset (when $> lower$ and $< upper$).
- the median absolute deviation of the annotations is required to be less than or equal to a predetermined threshold. Practically, this value indicates the average deviation that results from the deviations of each annotation from the average.

Furthermore, some agreement metrics have been calculated in order to get an idea of how well defined the final labels are. Firstly, the average disagreement of annotators is defined as the average value of the median absolute deviations from all samples. A second metric that indicates the annotators validity, is the average disagreement for each participant. That is, for each sample that the user annotated, the deviation of the label she/he has set from the average value of all the annotations of this sample is calculated and then averaged across all the user's annotations in order to get the average user disagreement.

The results of the filtering procedure and the calculated agreement metrics for the expressiveness task are listed in Table 5. The corresponding tables for the remaining tasks can be found on the dataset's repository[3].

Here, it has to be noted that the average disagreement of the task "ease of following" is high enough (ie. 0.57 female - 0.58 male) which indicates that this task is ill-defined and thus it will not be accounted for the experiments.

5.4 Data Availability

PuSQ is publicly available in https://github.com/sofiaele/PuSQ in the form of extracted audio and text features and ASR text files for the two valid tasks (expressiveness and enjoyment).

6 Experiments

For each of the two tasks (expressiveness, enjoyment), 8 different types of experiments, in terms of the features used, were conducted. More specifically, the below features, together with early or late fusion among them, were used:

1. **Meta Audio (MA):** Audio features derived from the segment-level classifiers together with the high-level audio features described in section 4.2, resulting in a 20d feature vector.
2. **Text (T)**: Text features derived from segment-level classifiers together with the high-level text features described in previous sections, resulting in a 22d feature vector.

[3] https://github.com/sofiaele/PuSQ

	Individual Modalities			Fusion Methods				
	Meta Audio MA	Text T	Low Level Audio LLA	MA + T	MA and LLA Late Fusion	MA and LLA Early Fusion	MA + T and LLA Late Fusion	MA + T and LLA Early Fusion
Female Expressiveness	71	37	**77**	66	**77**	76	75	75
Male Expressiveness	69	41	**71**	71	75	-	**79**	-
Female Enjoyment	44	**65**	57	51	44	57	51	**62**
Male Enjoyment	57	48	**70**	60	64	70	**74**	66
Free Text Expressiveness	75	65	**87**	91	87	84	**93**	86
Free Text Enjoyment	**71**	57	56	**86**	66	62	75	67

Table 6: Evaluation of recording-level classification (AUC metric)

3. **Low Level Audio (LLA)**: Low level audio features which are a long-term average of the features that were presented in section 3.1.1 and were used for segment classifiers, resulting in a 136d feature vector.

During the conducted experimentation three types of classifiers were tested: *(i)* SVM with RBF kernel, *(ii)* Gaussian Naive Bayes, and *(iii)* Logistic Regression. After the appropriate data pre-processing and parameter tuning techniques, a Leave-One-Speaker-Out (LOSO) validation was used in order to evaluate the performance of the classifiers in a speaker-independent manner. The metric of interest was the Area Under the ROC Curve (ROC-AUC) calculated on the aggregated probabilities of the LOSO validation. This metric was chosen instead of other widely used classification metrics, such as the f1-score, since the data are minimal and thus f1-score is prone to small changes.

In Table 6, the final results are summed up. The last two rows include the outcome of the evaluation of free-text only samples, ie. only the answers to questions and not predefined texts. It has to be noted that the Gaussian Naive Bayes was the best performing algorithm for all tasks, except from Male Expressiveness/Meta Audio where Logistic Regression was chosen.

From the presented evaluations, it can be easily seen that in all cases, the best fusion method has either increase or keep equivalent performance compared to the best individual method. The only exception is Female Enjoyment, where there is a slight deterioration of an absolute 3%, which however can be considered negligible.

Another important observation is associated with the tasks of Male Expressiveness and Male Enjoyment, where the combination of Meta Audio with Text features (MA + T), seems to result in increased performance compared to MA and T individually. This fact shows that the textual information can significantly help in distinguishing the two classes (negative, positive), mostly when involving free text, where the recording differs among participants and contains different semantical information.

In addition, it is observed that in most cases (4 out of 6), the combination of information from all feature spaces results in the best performance. Also, most of the times, late fusion marks better results than early fusion, which indicates that late fusion introduces a normalization factor in that dataset, since the models are not directly exposed to the low level features that may result in over-fitting.

The code of the experimentations is open sourced and can be accessed in `https://github.com/tyiannak/readys`.

7 Conclusion

In this work we presented PuSQ, a public speaking quality dataset, that introduces the tasks of speech expressiveness and enjoyment in public speech data. In order to address these speech quality assessment tasks, we designed a hierarchical classifier that is based on both segment-level emotion analysis and recording-level analysis where the aforementioned information is aggregated along with some high-level speech features. It is noteworthy that the presented pipeline can be used for any multimodal (audio and text) speech analytics process, and that both the dataset and the proposed ML framework are openly provided.

In a future work, several issues can be addressed, such as the extension of the dataset, the integration of learning methods that take into account the annotation confidence (eg. Sharmanska et al., 2016; Fornaciari et al., 2021), the use of more robust segment-level classifiers (CNNs,LSTMs,Transformers etc) (eg. Fayek et al., 2017; Jiang et al., 2020) as well as the inclusion of domain adaptation techniques (eg. Mao et al., 2016, 2017; Ocquaye et al., 2019; Huang et al., 2017) and the application of transfer learning from unsupervised temporal models (eg. Wang and Zheng, 2015; Feng et al., 2019).

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Felix Burkhardt, Astrid Paeschke, M. Rolfes, Walter Sendlmeier, and Benjamin Weiss. 2005. A database of german emotional speech. volume 5, pages 1517–1520.

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower Provost, Samuel Kim, Jeannette Chang, Sungbok Lee, and Shrikanth Narayanan. 2008. Iemocap: Interactive emotional dyadic motion capture database. *Language Resources and Evaluation*, 42:335–359.

Robert E. Carlson and Deborah Smith-Howell. 1995. Classroom public speaking assessment: Reliability and validity of selected evaluation instruments. *Communication Education*, 44(2):87–97.

Lei Chen, Gary Feng, Chee Wee Leong, Jilliam Joe, Christopher Kitchen, and Chong Min Lee. 2016. Designing an automated assessment of public speaking skills using multimodal cues. *Journal of Learning Analytics*, 3:261–281.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system.

Giovanni Costantini, Iacopo Iaderola, Andrea Paoloni, and Massimiliano Todisco. 2014. EMOVO corpus: an Italian emotional speech database. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, pages 3501–3504, Reykjavik, Iceland. European Languages Resources Association (ELRA).

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

P. Ekman. 1992. An argument for basic emotions. *Cognition & Emotion*, 6:169–200.

Haytham M. Fayek, Margaret Lech, and Lawrence Cavedon. 2017. Evaluating deep learning architectures for speech emotion recognition. *Neural Networks*, 92:60–68. Advances in Cognitive Engineering Using Neural Networks.

Kexin Feng, Megha Yadav, Md Nazmus Sakib, Amir Behzadan, and Theodora Chaspari. 2019. Estimating public speaking anxiety from speech signals using unsupervised transfer learning. In *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1–5.

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.

Theodoros Giannakopoulos. 2015. pyaudioanalysis: An open-source python library for audio signal analysis. *PLOS ONE*, 10:e0144610.

Jacek Grekow. 2018. *From Content-based Music Emotion Recognition to Emotion Maps of Musical Pieces*. Springer, Cham.

Zhengwei Huang, Wentao Xue, Qirong Mao, and Yongzhao Zhan. 2017. Unsupervised domain adaptation for speech emotion recognition using pcanet. *Multimedia Tools and Applications*, 76.

Philip Jackson and Sana ul haq. 2011. Surrey audio-visual expressed emotion (savee) database.

Changjiang Jiang, Junliang Liu, Rong Mao, and Sifan Sun. 2020. Speech emotion recognition based on dcnn bigru self-attention model. In *2020 International Conference on Information Science, Parallel and Distributed Systems (ISPDS)*, pages 46–51.

Ledisi Kabari and Believe Nwamae. 2019. Principal component analysis (pca) -an effective tool in machine learning.

Panagiotis Koromilas and Theodoros Giannakopoulos. 2021. Deep multimodal emotion recognition on human speech: A review. *Applied Sciences*, 11(17):7962.

Duc Le, Zakaria Aldeneh, and Emily Mower Provost. 2017. Discretized continuous speech emotion recognition with multi-task deep recurrent neural network. In *Interspeech*, pages 1108–1112.

S. R. Livingstone and F. Russo. 2018. The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. *PLoS ONE*, 13.

Qirong Mao, Guopeng Xu, Wentao Xue, Jianping Gou, and Yongzhao Zhan. 2017. Learning emotion-discriminative and domain-invariant features for domain adaptation in speech emotion recognition. *Speech Communication*, 93:1–10.

Qirong Mao, Wentao Xue, Qiru Rao, Feifei Zhang, and Yongzhao Zhan. 2016. Domain adaptation for speech emotion recognition by sharing priors between related source and target classes. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2608–2612.

Tomáš 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.

Behnaz Nojavanasghari, Deepak Gopinath, Jayanth Koushik, Tadas Baltrušaitis, and Louis-Philippe Morency. 2016. Deep multimodal fusion for persuasiveness prediction. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, pages 284–288.

Elias Nii Noi Ocquaye, Qirong Mao, Heping Song, Guopeng Xu, and Yanfei Xue. 2019. Dual exclusive attentive transfer for unsupervised deep convolutional domain adaptation in speech emotion recognition. *IEEE Access*, 7:93847–93857.

Andrew Ortony and Terence Turner. 1990. What's basic about basic emotions? *Psychological review*, 97:315–31.

Sunghyun Park, Han Suk Shim, Moitreya Chatterjee, Kenji Sagae, and Louis-Philippe Morency. 2014. Computational analysis of persuasiveness in social multimedia: A novel dataset and multimodal prediction approach. In *Proceedings of the 16th International Conference on Multimodal Interaction*, pages 50–57.

Lisa Schreiber, Gregory Paul, and Lisa Shibley. 2012a. The development and test of the public speaking competence rubric. *Communication Education*, 61.

Lisa M. Schreiber, Gregory D. Paul, and Lisa R. Shibley. 2012b. The development and test of the public speaking competence rubric. *Communication Education*, 61(3):205–233.

Viktoriia Sharmanska, Daniel Hernandez-Lobato, Jose Miguel Hernandez-Lobato, and Novi Quadrianto. 2016. Ambiguity helps: Classification with disagreements in crowdsourced annotations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Theodoros Giannakopoulos. 2020. *Pytorch implementation of deep audio embedding calculation Resources*.

Dong Wang and Thomas Fang Zheng. 2015. Transfer learning for speech and language processing. In *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pages 1225–1237.

Zhe Wang, Chunhua Wu, Kangfeng Zheng, Xinxin Niu, and Xiujuan Wang. 2019. Smotetomek-based resampling for personality recognition (july 2019). *IEEE Access*, PP:1–1.

A. E. Ward. 2013. The assessment of public speaking: A pan-european view. In *2013 12th International Conference on Information Technology Based Higher Education and Training (ITHET)*, pages 1–5.

Wilhelm Max Wundt and Charles Hubbard Judd. 1897. *Outlines of psychology*. Leipzig, W. Engelmann, New York, G.E. Stechert.

Yukun Zhu, Ryan Kiros, Richard 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.

A New Approach for Arabic Text Summarization

Samira Lagrini
Labged Laboratory, Computer Science Department,
Badji Mokhtar University, P.O. Box 12,
Annaba, 23000, Algeria

samira.lagrini@univ-Annaba.dz

Mohammed Redjimi
LICUS Laboratory, Computer science Department
Université 20 Aout 1955, Skikda,
21000, Algeria

medredjimi@gmail.com

Abstract

Due to the increasing number of online textual information, acquiring relevant information quickly has become a challenging task. Automatic text summarization (TS) offers a powerful solution for the quick exploitation of these resources. It consists of producing a short representation of an input text while preserving its relevant information and overall meaning. Automatic text summarization has seen a great attention for Indo-European languages. However, for Arabic, researches in this field have not yet attaint a notable progress. Most of the existing approaches in Arabic text summarization literature rely mainly on numerical techniques and neglect semantic and rhetorical relations connecting text units. This affects negatively the global coherence of the generated summary and its readability. In this paper, we attempt to overcome this limitation by proposing a new approach that combines a rhetorical analysis following the rhetorical structure theory (RST) and a statistical-based method. The proposed approach relies on exploiting rhetorical relations linking text units to generate a primary summary, which will be pass by a second phase where a statistical processing is applied in order to produce the final summary. Evaluation results on Essex Arabic Summaries Corpus (EASC) using ROUGE-N measures are very promising and prove the effectiveness of the proposed approach.

1. Introduction

Automatic text summarization is a fundamental task in Natural Language Processing (NLP). It consists of producing a brief representation of an input text covering its relevant content and overall meaning. This allows researcher acquiring needed information with minimum effort and accurately exploiting available resources.

The idea of designing text summarization systems dates back to 1950s (Luhn, 1958; Baxendale, 1958; Edmundson, 1969) in order to satisfy the first needs in term of automatic text summaries. But this need has become even more excessive with the advent of the Internet and the exponential increase of textual information in electronic format. This situation has sparked a great attention within the Natural Language Processing (NLP) community. Many researchers have fully invested in this field and a lot of research works have been devoted to produce automatic text summarizers in different languages. However, up to date there are several problems without effective solutions.

Generally, producing automatic text summaries can be done following two main paradigms: extractive summarization and abstractive summarization. In abstractive summarization, producing a summary involves an in-depth analysis of the source text in order to select the relevant content and to product the summary (abstract) using other words not necessary presented in the source text. This requires many advanced linguistic resources for text representation, sentences fusion and automatic text generation. However, in extractive summarization, relevant sentences in the source text are selected and directly assembled without any reformulation to produce the summary. Compared to abstractive summarization, extractive approaches are simple to implement and require only certain linguistic aspects. This is why most researches in this field focuses on extractive text summarization. The approach developed here is also based on extractive Arabic text summarization.

Several extractive summarization approaches have been developed to date to produce Arabic extracts including numerical, linguistic and hybrid methods. Numerical methods rely on computational values or a

statistical distribution of particular features to judge the relevance of textual segments including statistical methods (Douzidia, and Lapalme, 2004; Alotaiby et al., 2012), supervised learning based methods (Sobh et al., 2006; Boudabous et al. 2010; Belkebir and Guessoum, 2015; Qaroush et al., 2019), clustering based methods (El Haj et al., 2011; Oufaida et al., 2014; Waheeb et al., 2020; Alqaisi et al., 2020; Alami et al., 2021) and graph based methods (Alami et al., 2017).
Linguistic methods rely on semantic relations or discursive structure to assess the relevance of sentences in the text (Kumar et al., 2016). The hybrid method is a combination of the two former methods used to produce summary (Azmi and Al-Thanyyan, 2012; Al Khawaldeh and Samawi, 2015; Azmi and Altmami, 2018).
By analyzing Arabic text summarization literature, we can notice that most of reported research rely on numerical methods based on traditional bag of word representation and don't take into account semantic and rhetorical relations linking textual units. This affect the global coherence of the generated summary and its readability.

In this research we try to overcome this limitation by proposing a new approach that combine a rhetorical analysis following the rhetorical structure theory (Mann and Thompson, 1988) and a statistical method. The proposed approach rely on exploiting rhetorical relations linking elementary discourse units to generate a primary summary, which will be pass by a second phase where a statistical processing is applied in order to produce the final summary.

The remainder of this paper is as follows: the second section copes with a general overview of the rhetorical structure theory (RST). In Section 3, the proposed approach will be described. In section 4, the results and evaluation of the proposed approach are presented, and finally the conclusion and future works are addressed in section 5.

2. Rhetorical structure theory

The Rhetorical structure theory (RST) (Mann and Thompson, 1988) is a prominent theory in discourse analysis (Taboada and Mann, 2006). It focuses on rhetorical analysis, which aims to represent an input text in a hierarchical form of rhetorical relations linking its text units. In the RST framework, if two non-overlapping atomic textual units called elementary discourse units (EDUs) are linked via a discourse relation (also called rhetorical or coherence relation), they constitute together another discourse unit which can in turn participates in another discourse relation (Mann and Thompson, 1988). Under a full analysis, a coherent text can be represented as a labelled tree, called discourse tree (or RST-tree).

The Rhetorical analysis in RST framework involves three tasks:

- Segmenting the text into elementary discourse units.
-Identifying discourse relations between adjacent discourse units.
- Linking all discourse units into labelled trees (RST-trees).

Figure 1 shows a sample of an RST-tree for the following text segment consisting of two sentences segmented into five EDUs.

[*The impact won't be that great,*]$_1$ [*said Graeme Lidgerwood of First Boston Corp.*]$_2$ [*This is in part because of the effect*]$_3$ [*of having to average the number of shares outstanding,*]$_4$ [*she said.*]$_5$

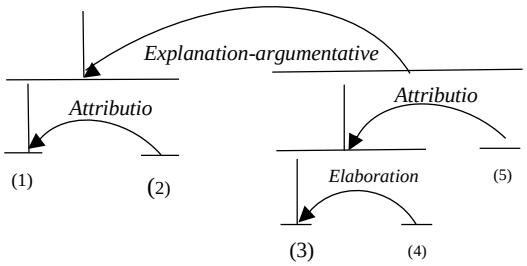


Figure 1. Example of an RST- tree for two sentences in RST–DT (Carlson et al., 2003)

Discourse Units (EDUs or larger discourse unit) linked via rhetorical relations are assigned a nuclearity attribute 'Nucleus' or 'satellite' depending on their relative importance in the text. The 'Nucleus' expresses what is more important for the author purpose, while the satellite provides a secondary information. In Figure 1, the 'Nucleus' are denoted by vertical line, Horizontal lines indicate discourse units, the Satellites are linked to their nucleus by curved arrows.

Rhetorical relations can be either '*mononuclear*' when they connect two discourse units having different status: 'Nucleus' and 'satellite', or '*multi-nuclear*' linking discourse units of equal importance, all nucleus.

The authors of RST defined a set of 24 relations, including 21 mononuclear relations. This set was extended by (Carlson et al., 2003) to 78 relation grouped into 16 class, which allows a high level of expression.

3. Proposed approach

In this research, we propose a new approach for Arabic single document summarization that combines rhetorical structure theory (RST) and a statistical-based method. Our aim is to exploit rhetorical relations linking text units in order to produce a coherent extracts. To this end, a rhetorical analysis is firstly performed to produce a primary summary relying mainly on rhetorical relations that exist between elementary discourse units (EDUs) rather than the discourse structure of the text. Then each sentence within the primary summary is assigned a score based on some statistical and linguistic features. Sentences having the best score will be selected to produce the output summary. Thus, the production of the summary go through two main phases; rhetorical analysis phase and statistical processing phase. Each phase consist of three main steps as shown in Figure 2.

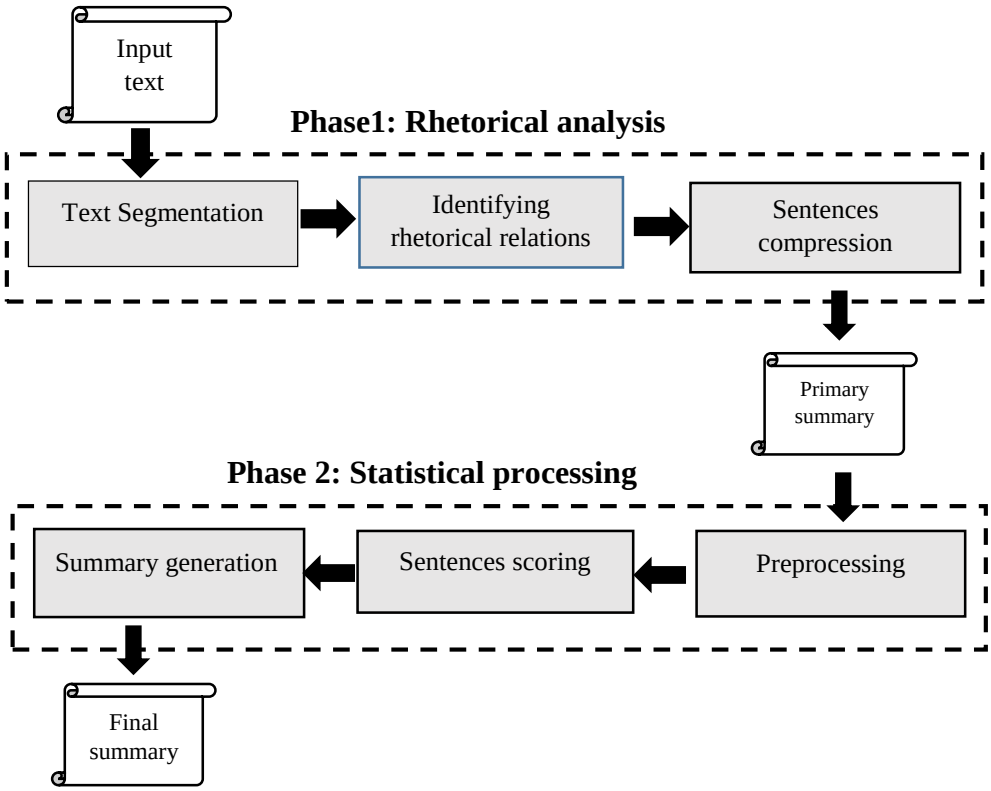


Figure 2: Proposed approach main steps.

3.1. Rhetorical analysis phase

The rhetorical analysis of the text includes the following subtasks: text segmentation, identifying rhetorical relations, and sentences compression.

3.1.1 Text segmentation

Text segmentation consists of breaking the text into non overlapping clauses called

elementary discourse units (EDUs). In our segmentation rules, an EDU can be a verbal or a nominal clause that begin with a discourse markers. Thus, segmenting the text into EDUs is performed as follow:

- First, segmenting the text into sentences. A sentence is defined as a textual passage delimited by (.).
- Then segmenting each sentence into EDUs based on Arabic discourse markers and a set of segmentation rules.

Arabic discourse markers such as (' وبالتالي'/therefore,' حيث'/ where,' حتى'/even, 'كذلك'/ as well as, 'بالرغم'/though) are already defined in a rich lexicon during the annotation process of our Arabic RST annotated corpus (Lagrini et al., 2019).

The following example presents a sentence segmented into four EDUs (between brackets), discourse markers are written in red bold.

[أكد سفير دولة فلسطين] $_1$[**ان** الأوضاع الصعبة التي يعيشها الشعب الفلسطيني]$_2$ [**هي نتيجة** الفرقة التي يشهدها الشارع الفلسطيني]$_3$[**و**تراجع الموقف العربي عن نصرة القدس]$_4$

[The ambassador of Palestine state confirmed $_1$[that the difficult circumstances that the Palestinian people live in,]$_2$ [are the result of the division seen in Palestinian street]$_3$[and the retreat of the Arab people on the support of Jerusalem]$_4$

3.1.2 Identifying rhetorical relations

Identifying rhetorical relations between two text segments has been shown to be useful in many Natural Language Processing tasks. Discourse markers or discourse connective such as: *because, although, since, but …, etc* strongly indicate the sense of explicit relations. For example 'because' is a strong indicator for causal relation. However, in the absence of such connectives the relation is called implicit and identifying such relation is still a big challenge. In our summarization process, we focus on identifying explicit relations and more precisely fine-grained relations between two adjacent elementary discourse units within the same sentence. In our previous work (Lagrini et al., 2019a), we have already defined a set of 23 fine-grained Arabic relations that can hold between two adjacent EDUs at the intra- sentential level.

These relation are grouped into seven classes: /causal, المقارنة /comparison, العطف' /joint, تفصيل /elaboration, توضيح /explanation, اسناد/ attribution, الشرط /conditional. For more details see (Lagrini et al., 2019a).

In the RST framework, fine-grained relations are enriched with nuclearity annotation:' SN', 'NS 'for mononuclear relations and 'NN' for multinuclear relations. These notations specify the rhetorical status of the connected discourse units. Taking as an example the following sentence composed of two EDUs:

[العلم يتقدم نافيا ما سبقه] [بمعنى ان اخر ما ينجزه العلم هو الأكثر صحة]

[Science advances in denial of what preceded it]$_1$[that's mean that the last thing that science accomplishes is the most correct]$_2$

These two EDUs are linked by the fine grained relation 'تفسير ' /'explanation/NS' signaled by the discourse marker 'بمعنى ان'/ that's mean'. The notation NS attached to the name of the relation means that the first EDU is the most important segment (Nucleus) denoted by N. while the second is the satellite, (denoted by S). It provides an optional information about the nucleus.

The nuclearity annotation attached to the name of relations is very interesting in our work, since it provides us with information about the relative importance of linked EDUs.

To automatically identify fine-grained rhetorical relations between adjacent EDUs, we have used a multi-class supervised learning approach based on a multi-layer perceptron model (Lagrini et al., 2019). We proceeded as follows:

- For each pair of adjacent EDUs, a feature vector is computed. We used ten group of lexical and semantic features. See (Lagrini et al., 2019) for a detailed description of used features.
- Then, all features vectors are fed as input to the model in order to predict fine-grained relations classes. Figure 3 summarizes this process.

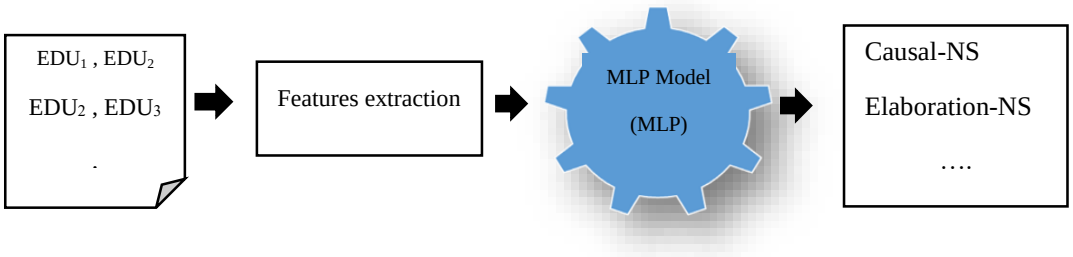


Figure 3: Identifying rhetorical relations process

3.1.3 Sentences compression

Once rhetorical relations between each pair of EDUs were identified, we proceeded to sentences compression. This step consists of removing satellite EDUs from each sentence while taking into consideration its overall coherence.

Sentence compression relies not only on the rhetorical status of its constituents EDU, but also on rhetorical relations. Some relations such as: تمثيل/example-NS, تشبيه/simile-NS, مقابلة/contrast-NN are not useful for the summary task. Thus, the presence of such relations involves the deletion of its constituents EDUs even if these EDUs are nucleus. Consider as an example the following sentence:

[ولقد دفع ذلك التوتر هذا البلد الى شن غارات قاتلة على جيرانه]$_1$[مما ولد رد فعل قاس لدى السلطات العسكرية بزعامة ضباطها الذين، كانوا يحاولون تهدئة الأوضاع]$_2$ [مراعاة لخاطر الاميركيين من جهة،]3[وتأجيلاً للانفجار المحتمل من جهة ثانية] $_4$.

This tension prompted this country to launch deadly raids on its neighbors (1) which generated a harsh reaction on the part of the military authorities led by its officers, who were trying to calm the situation, (2) taking into account the dangers of the Americans on the one hand, and (3) postponing the possible explosion on the other hand (4)

The sentence is composed of four EDUs linked by the following fine-grained relations:

Relation (1,2)= نتيجة/ result-NN;
Relation (2,3)=غاية /purpose-NS;
Relation (3,4)=وصل / joint-NN

Sentence compression involves removing EDU3 because it is a segment satellite and removing EDU4 since it is linked by a multinuclear relation with EDU3. That's means, they have the same level of importance. Therefore the compressed version of the sentence is as follow:

ولقد دفع ذلك التوتر هذا البلد الى شن غارات قاتلة على جيرانه مما ولد رد فعل قاس لدى السلطات العسكرية بزعامة ضباطها الذين، كانوا يحاولون تهدئة الأوضاع .

Compressing all sentences in the input text results in an abbreviate version of the source text which we consider as a primary summary.

3.2. Statistical processing phase

The goal of this phase is to reduce the number of compressed sentences in the primary summary and selecting the most relevant ones to produce the final summary. This phase includes the following subtasks: preprocessing the primary summary, sentence scoring, and summary generation.

3.2.1 Preprocessing of the primary summary

Preprocessing consists of three sequenced steps: tokenization, stop-words removal, and stemming.

Tokenization: consists of segmenting an input text into paragraphs, sentences, and words called tokens (Attia, 2007). As the primary summary is already segmented into sentences, AraNLP tool (Althobaiti et al., 2014) was used to segment each sentence into list of tokens.

Stop-words removal: Stop words are non-informative words that are frequently used in the text such as conjunctions, pronoun, prepositions, .. etc. They serve only a syntactic function but not indicate any relevant information. Removing these words is necessary to avoid affecting words weighting process (El-Khair, 2006). In our system, we have used the general stop-words list of AraNLP tool (Althobaiti et al., 2014) containing environ 168 words.

Stemming: Stemming is a morphological technique that consists of reducing inflected words to their stem or root by removing affixes

attached to them. For example the words ' استعمل' 'استعمالات' 'عامل' ' can be stemmed to word 'عمل'. For Arabic language, there are two main approaches for stemming: *Light-Based Stemming* and *root based stemming*. Following a comparative study between these two approaches regarding text summarization (Alami et al., 2016), it has been shown that root based stemming performs better than light stemming. This is why we choose to use in our summarizer khoja stemmer (Khoja and Garside, 1999) as a root based stemmer.

3.2.2 Sentence scoring

After preprocessing, each sentence was assigned a score based on certain features to assess its relevance. In text summarization literature, several features have been explored including key terms, indicative phrases, sentence position, sentence cohesions ... etc. In our summarization method, we used the following features: sentence position, sentence length and title similarity. These features have been successfully used in several works reported in Arabic summarization literature (Al-Radaideh and Bataineh, 2018; Al-Abdallah and Al-Taani, 2017; Douzidia and Lapalme, 2004; Fattah et al., 2009).

Title Similarity: As the title usually covers the main topic covered in the text, title words can be considered as key terms. Therefore, sentences that contain title words are relevant sentences and should be included in the final summary. Title similarity score assigned to each sentence is a function of the co-occurrences of title words in the sentence. This score is calculated using the following formula:

$$\boldsymbol{T}itle - sim(Si,T) = \frac{\text{title words} \cap \text{sentence (Si)}}{\text{title words}} \quad (1)$$

Sentence position: Sentence position in the text can be a good indicator that reflects its degree of importance. Generally in news articles, relevant sentences are either located at the beginning of the document or at the end. This is why we consider the first and the last sentence in the primary summary very important and should be included in the final summary. The position score assigned to each sentence is calculated as follows:

$$pos(Si) = \begin{Bmatrix} 1 & if\ i = 1\ or\ i = N \\ 0.5 & otherwise \end{Bmatrix} \quad (2)$$

With:
i: sentence number
N: total number of sentences within the input text.

Sentence length: As the majority of sentences in the primary summary only contain nucleus EDUs, we can say that the longer sentences are those which are most likely to contain more relevant information. Thus the score assigned to each sentence based on its length (the length in terms of words) is calculated as follows:

$$length(Si) = \frac{N(Si)}{N(Sl)} \quad (3)$$

With:
N (Si): number of words in the sentence Si
N (SL): number of words in the longest sentence in the primary summary.

The final score of each sentence is a linear combination of its scores assigned for each feature. This score represents the degree of relevance of the sentence in the primary summary. It is calculated using the following formula:

Score (Si) = Title-similarity (Si, T) + length (Si) + position (Si) *(4)*

3.2.3 Summary generation

In this phase, Sentences were ranked in descending order according to their final scores. Best scored sentences were then selected to produce the final summary. The selected sentences were assembled and arranged according to their order of appearing in the primary summary. The number of selected sentences depends on user's compression rate.

4. Evaluation and results

To evaluate the performance of our system we relied on intrinsic evaluation. Such evaluation seeks to evaluate automatic summaries based on their forms and contents. Content assessment measures the ability of the system to identify relevant sentences from the source document, this can be done automatically by comparing the generated summaries with reference summaries produced by human experts.

4.1 Evaluation dataset

For automatic evaluation we used Essex Arabic Summaries Corpus (EASC) (El Haj et al., 2010). The corpus consists of 153 documents extracted from Wikipedia and two Arabic newspapers: ALRai and Alwatan, covering ten different topics: art and music, education, environment, finance, health, politic, religion, science and technology, sport and tourism.

For each document in EASC, five reference summaries produced by humans are available. That is, the corpus is composed of a 765 reference summaries whose size does not exceed 50% of the size of the source text. EASC is available online with two encodings: UTF-8 and ISO-Arabic.

To evaluate our system, we selected a collection of 40 news articles from EASC corpus with their fives references summaries.

4.2 Evaluation measures

We used ROUGE (Recall-Oriented Understudy for Gisting Evaluation) metric (Lin, 2004) to evaluate our system. ROUGE is an automatic evaluation method that assess the quality of generated summary by comparing its content against one or more reference summaries. This comparison can be made by computing overleaping words such as Ngram (ROUGE-N) or word pairs (ROUGE-S and ROUGE-SU) or word sequence (Rouge-L, ROUGE-W)

ROUGE-N calculates the number of overleaping N-grams (N successive words) between the machine summary and reference summaries. Different metrics can be used such as ROUGE-1 (unigrams) , ROUGE-2(bi-grams), ROUGE-3 (trigrams)..etc.

In our system evaluation we used two metrics: ROUGE-1 and ROUGE-2 of ROUGE-N. For each metric, the precision, recall and F-score are calculated in order to provide a complete information about the system.

Recall: indicates the coverage of the system. It is calculated using the following formula:

$$Recall = \frac{nomber\ of\ overloapping\ N-grams}{nomber\ of\ N-grams\ in\ the\ set\ of\ refrence\ summaries} \quad (5)$$

Precision: Indicates the accuracy or the exactitude of the system. It is calculated as follow:

$$precision = \frac{nomber\ of\ overloapping\ n-grams}{nomber\ of\ \ n-grams\ in\ generated\ summary} \quad (6)$$

F-score: combines precision and recall, this measure is calculated as follows:

$$F-score = \frac{2*precision*rapell}{precision+rapell} \quad (7)$$

4.3 Results and analysis

Figure 4 shows performance evaluation of our summarization system on a collection of 40 articles from EASC with a compression ratio CR=50%. Each generated summary was compared against five reference summaries in EASC corpus using ROUGE-1 and ROUGE-2 metrics.

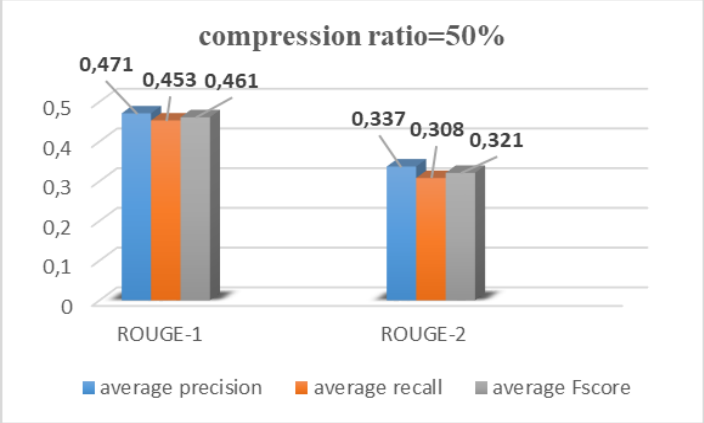


Figure 4: Performance of the proposed system with compression ratio=50%

Results analysis shows that our system achieves very good performance in term of precision, recall and F-score for both metrics ROUGE-1 and ROUGE-2. The average recall of our system attaint 0.453 using ROUGE-1 and 0.308 using ROUGE-2 indicating a high level of completeness and coverage. We can also note that the average precision of our system for both metrics is very good (average precision = 0.471 using ROUGE-1 and 0.337 using ROUGE-2) this means that the proposed system is quite preferment in excluding irrelevant sentences.

5. Conclusion and future works

To conclude, in this article, we have presented a new approach for automatic Arabic text summarization. The proposed approach combines a linguistic processing based on rhetorical analysis with a statistical processing. Rhetorical analysis is firstly applied to compress text sentences while keeping relevant segments. Sentence compression task is based on the exploitation of rhetorical relations defined within the rhetorical structure theory framework. Statistical processing is then used to reduce the number of compressed sentences based on three features: sentence position, similarity to the text title and sentence length. Results analysis on a text collection from EASC Data set proved clearly the efficacy of the proposed approach in terms of ROUGE-1 and ROUGE-2 measures.

As a future work, we will investigate the use of more linguistic features in statistical processing phase as well as exploiting both Intra-sentence and inter-sentence rhetorical relations to produce Arabic extracts.

References

Luhn, H. P. 1958. The automatic creation of literature abstracts. *IBM Journal of research and development*, *2*(2): 159-165.

Baxendale, P. B. 1958. Machine-made index for technical literature—an experiment. *IBM Journal of Research and Development*, *2*(4): 354-361.

Edmundson, H. P. 1969. New methods in automatic extracting. *Journal of the ACM (JACM)*, *16*(2): 264-285

Fouad Soufiane, Douzidia, and Guy Lapalme. (2004). Lakhas, an Arabic summarization system. In *Proceedings of DUC* (Vol. 4), pages 128-135.

Alotaiby, F., Foda, S., Alkharashi, I. 2012. New approaches to automatic headline generation for Arabic documents. *Journal of Engineering and Computer Innovations*, *3*(1):11-25.

Ibrahim, Sobh, Nevin Darwish, and Magda Fayek.. 2006. An optimized dual classification system for Arabic extractive generic text summarization. *In Proceedings of the 7th Conference on Language Engineering, pages 149–154.*

Fattah, M. A., Ren, F. 2009. GA, MR, FFNN, PNN and GMM based models for automatic text summarization. *Computer Speech & Language*, *23*(1):126-144.

Mohamed Mahdi, Boudabous, Mohamed Hédi Maaloul, and Lamia Hadrich Belguith. 2010. Digital learning for summarizing Arabic documents. *In proceeding of International Conference on Natural Language Processing*, pages 79-84. Springer, Berlin, Heidelberg.

Riadh Belkebir, Ahmed Guessoum. 2015. A supervised approach to Arabic text summarization using adaboost. In *New contributions in information systems and technologies*, pages 227-236. Springer International Publishing

Al-Radaideh, Q. A., Bataineh, D. Q. 2018. A Hybrid Approach for Arabic Text Summarization Using Domain Knowledge and Genetic Algorithms. *Cognitive Computation*, pages 1-19.

Qaroush, A., Farha, I. A., Ghanem, W., Washaha, M., & Maali, E. 2019. An efficient single document Arabic text summarization using a combination of statistical and semantic features. *Journal of King Saud University- Computer and Information Sciences, https://doi.org/10.1016/j.jksuci.2019.03.010.*

Mahmoud El-Haj, Udo Kruschwitz, and Chris Fox. 2011. Exploring clustering for multi-document Arabic summarisation. In *Asia Information Retrieval Symposium*, pages 550-561. Springer, Berlin, Heidelberg.

Oufaida, H., Nouali, O., Blache, P. 2014. Minimum redundancy and maximum relevance for single and multi-document Arabic text summarization. *Journal of King Saud University-Computer and Information Sciences*, *26*(4): 450-461.

Hamzah Noori, Fejer, Nazlia, Omar. 2014. Automatic Arabic text summarization using clustering and keyphrase extraction. *In proceeding of International Conference on Information Technology and Multimedia (ICIMU)*, IEEE. *Putrajaya, Malaysia*, pages 293-298.

Waheeb, S. A., Khan, N. A., Chen, B., and Shang, X. 2020. Multi-document Arabic text summarization based on clustering and Word2Vec to reduce redundancy. *Information*, *11*(2), 59.

Alqaisi, R., Ghanem, W., and Qaroush, A. 2020. Extractive Multi-Document Arabic Text Summarization Using Evolutionary Multi-Objective Optimization With K-Medoid Clustering. *IEEE Access*, *8* : 228206-228224.

Alami, N., Meknassi, M., En-nahnahi, N., El Adlouni, Y., & Ammor, O. 2021. Unsupervised neural networks for automatic Arabic text summarization using document clustering and topic modeling. *Expert Systems with Applications*, *172*: 114652.

Nabil, ALAMI, Yassine El Adlouni, Noureddine En-nahnahi, Mohammed Meknassi. 2017. Using Statistical and Semantic Analysis for Arabic Text Summarization. In proceeding of *International Conference on Information Technology and Communication Systems*, Springer, Cham, pages 35-50.

Kumar, Y. J., Goh, O. S., Halizah, B., Ngo, H. C., Puspalata, C. 2016. A review on automatic text summarization approaches. Journal of Computer Science, *12*(4):178-190.

Azmi, A. M., Al-Thanyyan, S. 2012. A text summarizer for Arabic. *Computer Speech & Language*, *26*(4), pp. 260-273.

Al Khawaldeh, F., Samawi, V. 2015. Lexical cohesion and entailment based segmentation for Arabic text summarization (LCEAS). *The World of Computer*

Science and Information Technology Journal (WSCIT), 5(3):51-60.

Azmi, A. M., and Altmami, N. I. 2018. An abstractive Arabic text summarizer with user controlled granularity. *Information Processing & Management, 54*(6): 903-921.

Mann, W. C., Thompson, S. A. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text-Interdisciplinary Journal for the Study of Discourse, 8*(3):243-281.

Taboada, M., and Mann, W. C. 2006. Rhetorical structure theory: Looking back and moving ahead. *Discourse studies, 8*(3): 423-459.

Lynn Carlson, Daniel Marcu, Mary Ellen Okurowski. 2003. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In: van Kuppevelt J., Smith R.W. (eds) Current and New Directions in Discourse and Dialogue. Text, Speech and Language Technology, volume 22, page 85-112, Springer, Dordrecht. https://doi.org/10.1007/978-94-010-0019-2_5

Lagrini, S., Azizi, N., Redjimi, M., and Dwairi, M. A. 2019. Automatic identification of rhetorical relations among intra-sentence discourse segments in Arabic. *International Journal of Intelligent Systems Technologies and Applications, 18*(3): 281-302.

Lagrini, S., Azizi, N., Redjimi, M., and Dwairi, M. A. 2019a. Toward an automatic summarisation of Arabic text depending on rhetorical relations. *International Journal of Reasoning-based Intelligent Systems, 11*(3): 203-214.

Attia, Mohammed. A. 2007. Arabic tokenization system. In *Proceedings of the 2007 workshop on computational approaches to semitic languages: Common issues and resources*, pages 65-72.

Maha Althobaiti, Udo Kruschwitz, Massimo Poesio. 2014. AraNLP: A Java-based library for the processing of Arabic text. *In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC), pages 4134-4138.*

El-Khair, I. A. (2006). Effects of stop words elimination for Arabic information retrieval: a comparative study. *International Journal of Computing & Information Sciences, 4*(3): 119-133.

Nabil Alami; Mohammed Meknassi; Said Alaoui Ouatik; NourEddine Ennahnahi. 2016. Impact of stemming on Arabic text summarization. In *2016 4th IEEE International Colloquium on Information Science and Technology (CiSt), IEEE, pages 338-343.*

Shereen Khoja and Roger Garside. (1999). Stemming Arabic text. Lancaster, UK, Computing Department, Lancaster University.

Al-Abdallah, R. Z., Al-Taani, A. T. 2017. Arabic Single-Document Text Summarization Using Particle Swarm Optimization Algorithm. *Procedia Computer Science, 117*: 30-37.

El Haj, M., Kruschwitz, U., Fox, C. (2010).Using Mechanical Turk to Create a Corpus of Arabic Summaries. *Editors & Workshop Chairs*, pages 36-39.

Chin-Yew, Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. *Proceedings of Workshop on Text Summarization Branches Out,* Barcelona, Spain, pages: 74-81.

Appendix. Sample Arabic text

والحقيقة ان قطاع غزة كان يقض مضاجع السلطات الاسرائيلية منذ زمن بعيد ، اذ وقبل اندلاع الأعمال الفدائية بزمن، كان هناك مناضلون فلسطينيون ينطلقون من ذلك القطاع وعبره للقيام بعمليات قاسية ضد قوات الاحتلال الاسرائيلية.وكان الوضع يصل الى لحظات توتر قصوى، عند بدايات 1955، حيث اندلعت أعمال عنف ضد القوات الاسرائيلية وكذلك ضد المنشآت التابعة للامم المتحدة. ولقد دفع ذلك التوتر اسرائيل الى شن غارات قاتلة على القطاع، مما ولد رد فعل قاس لدى السلطات المصرية بزعامة جمال عبدالناصر الذي، كان يحاول ان يهدئ الأوضاع، مراعاة لخاطر الاميركيين من جهة، وتأجيلاً للانفجار المحتمل بين مصر واسرائيل من جهة ثانية . ومن هنا حين احتلت القوات الاسرائيلية غزة في العام 1956 خيل للكثيرين انها لن تنسحب منها بعد ذلك، على رغم الضغوط الدولية . ولقد حاولت فئات نيابية كثيرة، ومنها مجموعات من حزب العمل الحاكم، نفسه، ان تطرح الثقة في الحكومة، لكن هذا كله لم يوهن من عزيمة حكومة بن غوريون التي لم تبد أي اهتمام حتى بالتظاهرات التي نظمتها المعارضة في الشارع داعية الى الابقاء على احتلال قطاع غزة.

In fact, Gaza Strip has been a long time ago disturbing the Israeli authorities, as long before the outbreak of guerrilla actions, there were Palestinian militants who launched from and through that strip to carry out harsh operations against the Israeli occupation forces. The situation reached moments of extreme tension, at the beginning of 1955, Where violence erupted against the Israeli forces as well as against United Nations Establishments. This tension prompted Israel to launch deadly raids on the Gaza Strip, which generated a harsh reaction from the Egyptian authorities led by Gamal Abdel Nasser, who was trying to calm the situation, taking into account the dangers of the Americans on the one hand, and postponing the possible explosion between Egypt and Israel on the other. Hence, when the Israeli forces occupied Gaza in 1956, many imagined that they would not withdraw from it after that, despite international pressures. Many parliamentary groups, including groups from the ruling Labor Party, have tried to raise confidence in the government, but all this did not weaken the resolve of the Ben-Gurion government, which did not show any interest even in the demonstrations organized by the opposition in the street calling for maintaining the occupation of the Gaza Strip.

The final summary followed by its translation is:

والحقيقة ان قطاع غزة كان يقض مضاجع السلطات الاسرائيلية منذ زمن بعيد. ولقد دفع ذلك التوتر اسرائيل الى شن غارات قاتلة على القطاع، مما ولد رد فعل قاس لدى السلطات المصرية بزعامة جمال عبد الناصر الذي كان يحاول ان يهدئ الأوضاع . ولقد حاولت فئات نيابية كثيرة، ومنها مجموعات من حزب العمل الحاكم، نفسه، ان تطرح الثقة في الحكومة، لكن هذا كله لم يوهن من عزيمة حكومة بن غوريون.

In fact, the Gaza Strip has long been a sleeper of the Israeli authorities. This tension prompted Israel to launch deadly raids on the Gaza Strip, which generated a harsh reaction from the Egyptian authorities led by Gamal Abdel Nasser, who was trying to calm the situation. Many parliamentary groups, including groups from the ruling Labor Party itself, tried to put confidence in the government, but all this did not weaken the resolve of the Ben-Gurion government.

Compressive Performers in Language Modelling

Anjali Ragupathi, Siddharth Shanmuganathan, Manu Madhavan
Department of Computer Science and Engineering
Amrita School of Engineering - Coimbatore
Amrita Vishwa Vidyapeetham, India
anjaliragupathi99@gmail.com
siddhsham@gmail.com
m_manu@cb.amrita.edu

Abstract

This work introduces the Compressive Performer, a hybrid Transformer variant based on two existing model architectures: the Performer, which reduces the memory requirement and processing time of the Transformer to linear complexity, and the Compressive Transformer, which retains contextual dependencies over a long range by compressing old activations instead of discarding them. Experiments in language modelling at the character level, the word level, and the sub-word level demonstrate that the Compressive Performer shows improved perplexity scores on the enwik-8 dataset, compared to its base models. This work also compares convolutional compression with autoencoder compression, determining that both show similar perplexity scores.

1 Introduction

Language models were initially based on the relative frequencies of words occurring in a corpus. Traditionally, statistical models like n-grams (Jurafsky and Martin, 2000) and temporal neural models like RNNs (Rumelhart et al., 1986; Hochreiter and Schmidhuber, 1997) were preferred to simulate the structure of natural language. However, they had many deficits like the inability to generalize over unseen words (for n-grams) or to capture information from prior parts of the sentence while taking less time to train(for RNNs).

Recent state-of-the-art deep learning models such as the Transformer (Vaswani et al., 2017) and its subsequent incarnations were created to mitigate these issues. They relied on parallelization techniques that efficiently utilised multiple GPUs - unlike RNNs, which could only use one at a time. Their core feature was "attention", which used $O(n^2)$ operations to find correlations between token pairs for a sequence of n tokens. Although attention had the advantage of large context windows, it also placed constraints on the performance of the Transformer in environments with small memories and few CPUs/GPUs. Since a quadratic number of operations was required ($n \times n$), the Transformer took longer to train. Additionally, each layer of the Transformer stored the output of its activation function in the form of an $n \times n$ matrix, which made the space complexity quadratic as well.

There have been many attempts to mitigate either the space complexity or the time complexity of Transformers using techniques that would reduce the number of computations (Vyas et al., 2020), the amount of data being processed (Li et al., 2016; Panahi et al., 2019), or the amount of information being stored in memory (Child et al., 2019; Katharopoulos et al., 2020; Lan et al., 2019). These were explored in a survey conducted by Tay et al. (2020). Drawbacks of these methods included loss of information due to sparse representations of data (Child et al., 2019), as well as the

uncertainty around the optimal kernel function to be used (Katharopoulos et al., 2020). While models like the Reformer (Kitaev et al., 2020) were effective in reducing the space complexity of the model, they could not adequately capture context over long-range sequences.

Conversely, the Transformer-XL model (Dai et al., 2019) aimed to preserve dependency across long sequences. It introduced relative positional encoding and applied a recurrence relation over segments of data, to extend the length of context dependency learnt by the model.

The success of Transformers sparked the development of two important variants - Compressive Transformer (Rae et al., 2019) and Performers (Choromanski et al., 2020; Likhosherstov et al., 2020). The former was developed as an extension of the TransformerXL which distinguished between long-term and short-term memories. Unlike its predecessor, it compressed old activations instead of discarding them, which allowed extended amounts of context to be preserved with minimal information loss.

In the Performer, a method called FAVOR - Fast Attention Via Positive Orthogonal Random Features - was proposed to scale attention linearly. The attention matrix was decomposed into random features of lower dimensionality, allowing information to be encoded to take up less space than a complete attention matrix. An extension to the algorithm also demonstrated that it was not necessary to construct the complete attention matrix, as the random features could be rearranged to form an approximation that had lower space and time complexities.

Based on the advantages demonstrated by the above models, this work proposes the Compressive Performer-a hybrid model which combines the linear space and time complexity of the Performer with the long-range context-sensitivity of the Compressive Transformer. The empirical performance of this integrated model is compared with the original models on which it is based. The goal is to preserve the low space complexity shown by the Performer without losing any valuable contextual information over a long-range sequence. To demonstrate the consistency of the model over different use cases, this work also involves tokenization of the dataset at three different levels of natural language - word, sub-word, and character.

2 Proposed Model

This section briefly describes the FAVOR+ algorithm implemented in the Performer (Choromanski et al., 2020), along with the two compression techniques used in the proposed model. Further, the integration of compression with FAVOR is discussed in Section 2.2.

2.1 FAVOR+ - Fast Attention

Fast attention was developed as an alternative to multiplicative attention / scaled dot-product attention(Luong et al., 2015) and Bahdanau's attention (Bahdanau et al., 2014). To bring down the space complexity for the model, the complete attention matrix is never constructed. Instead, the softmax function is approximated by using suitable kernel functions which make use of "random orthogonal features". Using this approximation, the original Q and K matrices cannot be reconstructed, but similar matrices of the same dimension ($L \times d$, where L is the input sequence length and d is the inner dimensionality) can be approximated.

The original attention formula is defined in Equation 1, while its corresponding FAVOR approximation is defined in Equation 2.

$$Attn(Q, K, V) = softmax(\frac{Q \cdot K^T}{\sqrt{d_k}}) \cdot V \quad (1)$$

where
$Attn$ = Attention Function

Q = Query Matrix
K = Key Matrix
V = Value Matrix
$\sqrt{d_k}$ = Scaling factor of dimension k

$$Attn(Q, K, V) = Q' \cdot ((K')^T \cdot V) \quad (2)$$

where
$Attn$ = Attention Function
Q' = Approximated Query Matrix
K' = Approximated Key Matrix
V = Original Value Matrix

2.2 Compression algorithm integrated with FAVOR

The algorithm used in the Compressive Performer is based on the original paper on the Compressive Transformer (Rae et al., 2019). It uses a FIFO queue as the data structure in which the attention weights at each memory layer are stored (Wang, 2020). This allows a time-independent Transformer to access "memories" in a definite temporal order. The queue is split into the short-term memory (*mem*) and the long-term memory (*cmem* for compressive memory). At each time step, after the attention weights have been calculated, they are pushed into the FIFO queue's short-term memory. Once this section is filled, the short-term memory data is compressed and pushed into the long-term memory, and the complete memory queue is updated. During backpropagation, the gradients used for the compressive network itself are not propagated into the main network and are instead used to improve reconstruction loss. However, since the memory layers are generally hidden layers, the stored activations are used to update the weight matrices on subsequent passes through the network, thus preserving context.

The proposed work compares two compression algorithms. The first is basic convolutional compression, which is generally used in images. It uses convolutional layers to extract features from data, and pooling layers to reduce the dimensions of data by discarding unnecessary features (as in Equation 3). When text data is represented in the form of vectors or tensors, the underlying numerical representation of language structure can also be compressed in a way similar to images (Mahoney, 2000; Cox, 2016; Goyal et al., 2018). The second type of compression is autoencoder compression. An autoencoder is an artificial neural network that can learn an encoding for data by training itself to ignore "noise". This encoding is a projection of the input data in latent space, typically of lower dimensionality than the original data. The proposed work uses convolutional autoencoders, which compress data with some loss of information. Hence, Mean Squared Error loss is used as the evaluation metric for reconstructing the original input from its encoded counterpart.

$$dim' = (dim - f)/(s + 1) \quad (3)$$

where dim' is the new length or width of image, dim is the original length or width of image, f is the filter dimension and s is the stride length.

While integrating the compression algorithm with FAVOR, no complete attention matrices are constructed, as in ordinary Performers. Pre-normalization is used at each layer instead of post-normalization, as described by Nguyen and Salazar (2019). Since backward propagation for the computation of attention reconstruction loss is nearly identical to the one described by Rae et al. (2019), its discussion remains beyond the immediate scope of this paper.

In Algorithm 1, the batch size is represented by b, the embedding dimensions by d, the length of the memory by l_{mem}, and the length of the compressed memory by l_{cmem}. The variable $h^{(i)}$ represents the hidden state at layer i. At time step 0, the memory queue is initialized to be empty. At each succeeding time

step, the subsequent input is extracted using random sampling. It is then passed to the embedding layer, where the embedding weights are calculated and relative positional encoding is applied to get the first hidden state (Dai et al., 2019). The long-term compressed memory and the short-term uncompressed memory are concatenated with the generated hidden state to form the queue. By applying a suitable projection function (for example, a simple linear function or multi-layer perceptron), the input is converted into the query, key, and value matrices. The FAVOR+ algorithm is then applied to these matrices as in Equation 2. Here, matrix associativity is applied to rearrange the three matrices, ensuring that the output of the attention function has linear complexity. Residual connections are made to prepare the model for the backward pass of training. In the compression phase, the oldest memories are extracted and compressed by the specified compression ratio c. The current hidden state is pushed into the short-term memory, while the compressed memories are pushed into the long-term memory. Per convention, normalization is applied to the output activations to speed up training time by making the gradients of the network stable. The next hidden state is generated and propagated to the following layers.

3 Experiments

3.1 Pre-processing and Tokenization

Raw data (a subset of the enwik8 dataset - originally curated by Mahoney (2006)) was read from its source file and pre-processed. The training and evaluation phases were tested at all three levels of tokenization. The number of bytes read from the input file had to be decreased from the character level model having the highest number (95 MB), to the word level having the lowest (25 MB). This was done because the large vocabulary of word level models would impede execution by creating many parameters, thus resulting in memory mismanagement.

Pre-processing included removing XHTML tags, delimiting sentences, and tokenizing them based on the level of natural language considered. At the character level, it was enough to split the data into separate characters. At the sub-word level, it was essential to identify the most common morphemes. While stemming can be used for this purpose, it remains inefficient when quick processing is desired. A Byte-Pair Encoding Tokenizer from HuggingFace was, thus, used to build common subwords based on the frequency of co-occurrence of characters. At the word level, a simple word tokenizer from the NLTK library was incorporated; it was complemented by a multi-word tokenizer to add delimiters to the vocabulary.

Post-tokenization, a vocabulary of the appropriate size was built and passed as a parameter to the learning algorithm. At the character and sub-word levels, the data was split into training, validation, and testing using a 90:5:5 split. For the word level model, the data was split using an 80:10:10 split, as per convention.

3.2 Training and Tuning

The model was trained using Kaggle Kernels on a Tesla P100 GPU with 16 GB of memory. At each iteration, gradient clipping was done to prevent exploding gradients. Perplexity (Equation 5) was calculated as an exponent of validation cross-entropy loss (Equation 4). Further discussion on suitable metrics for language models can be found in the article by Huyen (2021).

Another mechanism that was implemented was a learning rate scheduler (Rath, 2021). The implementation scaled down the learning rate by a factor of 0.5 if it observed no improvement in the validation loss for at least 5 epochs in a row. Early stopping (Rath, 2021) was

Algorithm 1 Proposed Algorithm: Compressive Performer Main Forward Propagation

At time t_0

$mem_0 \leftarrow 0$ ▷ Shape of memory is *b x* l_{mem} *x d*

$cmem_0 \leftarrow 0$ ▷ Shape of compressed memory is *b x* l_{cmem} *x d*

for t in 1, 2, ... $n_{timesteps}$ **do**

  $h^{(1)} \leftarrow xW_{emb}$ ▷ W_{emb} is the weight matrix for embeddings, x is the input tensor

  for i in 1, 2, ... n_{layers} **do** ▷ n_{layers} represents the number of memory layers

    $fifo^{(i)} \leftarrow concat(cmem_t^{(i)}, mem_t^{(i)}, h^{(i)})$ ▷ Create FIFO queue with both memory queues and input

    $q^{(i)}, k^{(i)}, v^{(i)} \leftarrow projection(fifo^{(i)})$ ▷ Extract Q, K, V matrices

    $a_{favor}^{(i)} \leftarrow FastAttention(q^{(i)}, k^{(i)}, v^{(i)})$ ▷ Apply FAVOR on the queue

    $a^{(i)} \leftarrow a_{favor}^{(i)} + h^{(i)}$ ▷ Apply residual connection

    $old\_mem^{(i)} \leftarrow mem_t^{(i)}[: n_{oldest}]$ ▷ Extract oldest memories to be compressed *(n_{oldest} x d)*

    $new\_cmem^{(i)} \leftarrow f_c^{(i)}(old\_mem^{(i)})$ ▷ Compress oldest memories by factor *c* ($\frac{n_{oldest}}{c}$ *x d*)

    $mem_{t+1}^{(i)} \leftarrow concat(mem_t^{(i)}, h^{(i)})[-l_{mem} :]$ ▷ Push current input into short-term memory

    $cmem_{t+1}^{(i)} \leftarrow concat(cmem_t^{(i)}, new\_cmem^{(i)})[-l_{cmem} :]$ ▷ Push new compressions into long-term memory

    $h^{(i+1)} \leftarrow pre\_norm(linear(a^{(i)})) + pre\_norm(a^{(i)})$ ▷ Generate next hidden state

  end for

end for

also included as a mechanism to stop training when the validation loss showed no sign of improvement. This was done in an attempt to avoid overfitting, based on suggestions in the paper (Komatsuzaki, 2019). Finally, the Adam optimizer (Kingma and Ba, 2014) was used because of its advantages over stochastic gradient descent. The addition of these optimization techniques reduced training time significantly from a few hours to a minute.

Using various tunable parameters like compression ratio, batch size, memory size, and sequence length, the learning algorithm iteratively calculated a set of weights that would minimize the cross-entropy loss and the auxiliary MSE loss which is related to the reconstruction error after compression. In addition to these parameters, the highest batch size that showed the least error was found to be 16 batches. A model depth of 6 was also found to be better suited to the size of the dataset than a model depth of 8. Most hyperparameter values have been taken from the original paper on Compressive Transformers (Rae et al., 2019), such as sequence length and memory length being 768, compressed memory length being 1152, and compression ratio being 4.

$$Loss_{CE} = -\sum_{i=1}^{k} y_{o,i} \log_2 p_{o,i} \tag{4}$$

where

$Loss_{CE}$ = Cross Entropy Loss

k = number of classes

$y_{o,i}$ = binary value which tells if observation o was correctly categorized as class i (true value)

$p_{o,i}$ = predicted probability of observation o being in class i

$$Perplexity = 2^{Loss_{CE}} \tag{5}$$

4 Results and Discussion

The two baseline models that have been considered for comparison and inference are the Compressive Transformer (Rae et al., 2019) and the Performer (Choromanski et al., 2020). The Performer has been fitted with reversible layers as described in the paper by (Kitaev

et al., 2020), based on the implementation by (Wang, 2021).

Each model has been studied at the character level, sub-word level, and word level. Vocabulary size was found to have increased, with *n(character)* $<$ *n(sub-word)* $<$ *n(word)*. This was primarily because the set of characters in ASCII comes out to 256 tokens, whereas the number of unique words is significantly larger. Sub-words may share morphological stems of words.

4.1 Character Level

When the models were trained at a character level, they had access to 256 possible classes or categories. Since this number was significantly lower than the vocabulary size at the other two levels, the cross-entropy loss function mentioned in Equation 4 had to compute the summation over fewer classes. Hence, the training and validation losses were part of a lower range of values. However, there was significant volatility in the losses as seen in Figure 1a, because the model could not learn relationships between characters in a structurally or semantically cognizant manner. However, the model did learn which characters occur together frequently. The losses remained much lower than that of the Performer and the Compressive Transformer, despite starting at similar points; this showed that the Compressive Performer tended to have a much better record of achieving a relevant prediction compared to the baseline models, which could be attributed to its ability to learn complex context dependencies.

It was also noted that all the character level models were highly unstable, but their curves plateaued after a certain elbow point. Since character level models had not implemented early stopping, the graphs in Figure 1a indicate that this would be a good strategy to follow in subsequent implementations.

4.2 Sub-word Level

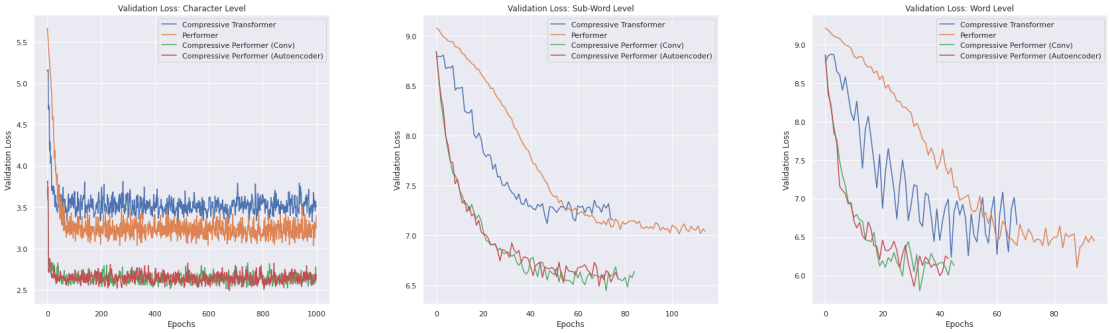
At the sub-word level, the training losses of both versions of the Compressive Performer were highly volatile, indicating that the number of classes was not easily determined. This was due to the numerous morpheme combinations that could potentially be predicted next. The model thus struggled to predict the best possible class consistently. Still, it can be seen in Figure 1b that the slope of the graphs was much steeper and decreased a lot more than for the baseline models. This showed that the Compressive Performer learned much faster and also more uniformly than both the Compressive Transformer and the Performer.

4.3 Word Level

At the word level, the loss graphs were much less volatile than at the sub-word level because of the relative ease of predicting a class. With a fixed-size vocabulary like that used in the word level model, the loss curves decreased with much less fluctuation. It can be seen in Figure 1c that the Performer still found predictions difficult because it could not preserve context dependencies as easily as the three compressive models.

4.4 Comparison and Inference

From the results, it can be seen that the proposed model could utilize compression algorithms without incurring large space overheads; thus, the space complexity of the Compressive Performer was closer to that of the Performer than to the Compressive Transformer. A comparison of both Compressive Performer variants with their baseline models is shown in Tables 1, 2 and 3. It is clear that the Compressive Performer has outperformed the baseline models at all three levels of experimentation, by achieving the lowest relative perplexity. Though the GPU utilization of the proposed model was greater than that of the Performer



(a) Character Level (b) Sub-word Level (c) Word Level

Figure 1: Cross-entropy Loss over Validation

Cross-entropy loss of Compressive Performer is much lower than the base models at all three levels. Differences in graph lengths in Figure (1b) and Figure (1c) are due to early stopping. Steep curves imply faster learning. High jitter implies greater uncertainty in predicting the next token (seen in all four models in Figure (1a) and in Compressive Transformer in Figure (1c))

Table 1: Performance Comparison of Models - Word Level

Model Name	Prediction Time(ms)	GPU RAM(GB)	Training Perplexity	Test Perplexity
Compressive Transformer	70.7	6.9	173.45	102.36
Performer	170	3.1	220.32	109.334
Compressive Performer (Conv)	**67.7**	**3.6**	**110.2**	**75.896**
Compressive Performer (Auto)	**57**	**3.6**	**111.69**	**75.141**

Table 2: Performance Comparison of Models - Sub-Word Level

Model Name	Prediction Time(ms)	GPU RAM(GB)	Training Perplexity	Test Perplexity
Compressive Transformer	32	6.9	209.69	179.28
Performer	140	3.2	230.93	171.316
Compressive Performer (Conv)	**55.375**	**3.5**	**128.4**	**127.82**
Compressive Performer (Auto)	**55.05**	**3.5**	**133.32**	**126.727**

Table 3: Performance Comparison of Models - Character Level

Model Name	Prediction Time(ms)	GPU RAM(GB)	Training Perplexity	Test Perplexity
Compressive Transformer	96	8.6	11.61	13.078
Performer	166	4.7	9.57	9.3378
Compressive Performer (Conv)	**51**	**7.2**	**6.26**	**6.4314**
Compressive Performer (Auto)	**51**	**6.8**	**6.295**	**6.025**

due to the inclusion of queues, it was considerably less than that of the Compressive Transformer.

At the word and sub-word levels, the Performer fared the worst at capturing semantic and contextual dependencies over long sequences. The Compressive Transformer used the highest amount of GPU RAM because of its dependence on traditional softmax attention, as opposed to the linear attention mechanism preferred by the three Performer variants.

At the character level, it was noted that

the activations computed by the convolutional compression algorithm took up slightly more RAM than those computed using autoencoder compression. The comparable training perplexity scores in Table 3 also showed the effectiveness of autoencoders in learning contextual representations from compressed data. The low perplexity scores compared to the word level and sub-word level models were due to the fact that there was a very small set of tokens (256 ASCII characters) that the model could predict. Additionally, the amount of GPU RAM used by the convolutional model was slightly higher than that used by the autoencoder model.

When the models were tested based on their prediction time (i.e., from input submission to output prediction), it was found that the Compressive Performer took much less time to predict the next tokens in a sequence at all three levels of tokenization. These results are demonstrated in the column "Prediction Time" in Table 1, Table 2, and Table 3. The models were also generalizable over new data and overfitted less than the base models, as demonstrated in the corresponding column "Test Perplexity".

Overall, the models proposed in this paper have yielded promising results in terms of space complexity and perplexity scores. Despite the use of random sampling, the perplexity scores of both Compressive Performers were observed to be consistently lower than the base models, while small variations were observed in the ranking of the base models themselves.

5 Conclusion

The goal of this work was to be able to train Transformer models on consumer-grade devices, without resorting to expensive cloud services and specialized hardware. It was determined that the Compressive Performer showed comparable space complexity to the Performer and maintained perplexity scores that were much lower than the base models. It was also determined that the autoencoder compression mechanism offered similar results to the convolutional compression mechanism. In addition to this, it was seen that the proposed model performed equally well on both training and testing data when compared to existing state-of-the-art models.

However, the model showed major limitations in the quality of text generated. Owing to a scarcity of suitable high-performance computing resources, the experiments in this paper were conducted with minimalistic models and a small dataset. While it is expected that the results can be replicated on a large scale, it would be necessary to attempt training the model from scratch using a benchmark dataset like PG-19 (Rae et al., 2019). Experiments may also be done with custom tokenizers and cleaning algorithms, as well as with other kernels and activation functions, to determine what results arise from them. Pre-training approaches like ELECTRA (Clark et al., 2020) could be implemented to help the model learn context adversarially.

References

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

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamás Sarlós, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. 2020. Rethinking attention with performers. *CoRR*, abs/2009.14794.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Elec-

tra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

David Cox. 2016. Syntactically informed text compression with recurrent neural networks. *arXiv preprint arXiv:1608.02893*.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

Mohit Goyal, Kedar Tatwawadi, Shubham Chandak, and Idoia Ochoa. 2018. Deepzip: Lossless data compression using recurrent neural networks. *arXiv preprint arXiv:1811.08162*.

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

Chip Huyen. 2021. Evaluation metrics for language modeling. https://thegradient.pub/understanding-evaluation-metrics-for-language-models/.

Daniel Jurafsky and James H. Martin. 2000. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 1st edition. Prentice Hall PTR, USA.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. 2020. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pages 5156–5165. PMLR.

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

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*.

Aran Komatsuzaki. 2019. One epoch is all you need. *arXiv preprint arXiv:1906.06669*.

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*.

Xiang Li, Tao Qin, Jian Yang, and Tie-Yan Liu. 2016. Lightrnn: Memory and computation-efficient recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 4385–4393.

Valerii Likhosherstov, Krzysztof Choromanski, Jared Davis, Xingyou Song, and Adrian Weller. 2020. Sub-linear memory: How to make performers slim. *arXiv preprint arXiv:2012.11346*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Matt Mahoney. 2006. Enwik-8 source version. https://cs.fit.edu/~mmahoney/compression/textdata.html.

Matthew V Mahoney. 2000. Fast text compression with neural networks. In *FLAIRS conference*, pages 230–234.

Toan Q Nguyen and Julian Salazar. 2019. Transformers without tears: Improving the normalization of self-attention. *arXiv preprint arXiv:1910.05895*.

Aliakbar Panahi, Seyran Saeedi, and Tom Arodz. 2019. word2ket: Space-efficient word embeddings inspired by quantum entanglement. *arXiv preprint arXiv:1911.04975*.

Jack W Rae, Anna Potapenko, Siddhant M Jayakumar, and Timothy P Lillicrap. 2019. Compressive transformers for long-range sequence modelling. *arXiv preprint arXiv:1911.05507*.

Sovit Ranjan Rath. 2021. Using learning rate scheduler and early stopping with pytorch. https://debuggercafe.com/using-learning-rate-scheduler-and-early-stopping-with-pytorch/.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2020. Efficient transformers: A survey. *arXiv preprint arXiv:2009.06732*.

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.

Apoorv Vyas, Angelos Katharopoulos, and François Fleuret. 2020. Fast transformers with clustered attention. *arXiv preprint arXiv:2007.04825*.

Phil Wang. 2020. Compressive transformer implementation pytorch. https://github.com/lucidrains/compressive-transformer-pytorch.

Phil Wang. 2021. Performer implementation pytorch. https://github.com/lucidrains/performer-pytorch.

Learning ULMFiT and Self-Distillation with Calibration for Medical Dialogue System

Shuang Ao
Doti Health Ltd, UK
The Open University, UK
ao.shuang@u.nus.edu

Xeno Acharya
Doti Health Ltd, UK
xeno.acharya@gmail.com

Abstract

A medical dialogue system is essential for healthcare service as providing primary clinical advice and diagnoses. It has been gradually adopted and practiced in medical organizations, largely due to the advancement of NLP. The introduction of state-of-the-art deep learning models and transfer learning techniques like Universal Language Model Fine Tuning (ULMFiT) and Knowledge Distillation (KD) largely contributes to the performance of NLP tasks. However, some deep neural networks are poorly calibrated and wrongly estimate the uncertainty. Hence the model is not trustworthy, especially in sensitive medical decision-making systems and safety tasks. In this paper, we investigate the well-calibrated model for ULMFiT and self-distillation (SD) in a medical dialogue system. The calibrated ULMFiT (CULMFiT) is obtained by incorporating label smoothing (LS) to achieve a well-calibrated model. Moreover, we apply the technique to recalibrate the confidence score called temperature scaling (TS) with KD to observe its correlation with network calibration. Furthermore, we use both fixed and optimal temperatures to fine-tune the whole model. All experiments are conducted on the consultation backpain dataset collected by experts then further validated using a large publicly medial dialogue corpus. We empirically show that our proposed methodologies outperform conventional methods in terms of accuracy and robustness.

1 Introduction

The medical dialogue system is becoming a necessary tool for the doctor-patient interaction as it provides the primary clinical advice and long-distance diagnoses, shortening the checking duration and reducing the manpower cost. It is gradually applied and accepted especially during the pandemic time.

In order to provide an integrated conversational system for back pain management, the system needs to be equipped with evidence on the aforementioned determinants of health. This could best be facilitated by incorporating this evidence through a medical dialogue system. However, the insufficient medical corpus is one of the biggest restrictions for the training of the neural conversational model. We build the dataset particularly for back pain consultation, including the query and suggestions regarding possible causes, symptoms, and treatment of back pain, which to our best knowledge, is the first medical conversational dataset subjected in the backpain field.

To achieve a promising accuracy of the sentence generation model, we choose the state-of-the-art transformer model (Vaswani et al., 2017) as the benchmark. As transfer learning has shown great success in machine learning tasks such as classification and regression (Pan and Yang, 2009), in this paper, we choose two well-known and efficient techniques: Universal Language Model Fine Tuning (ULMFiT) (Howard and Ruder, 2018) and Knowledge Distillation (KD) (Hinton et al., 2015). ULMFiT provides additional help to improve the model accuracy by transferring information from language modeling to NLP downstream tasks, such as conversational model, sentiment analysis, and Machine Translation. Due to its obvious advantages, we implement the pretrained language model on top of the conversation model to get better feature extraction. Furthermore, KD transfers the knowledge from a cumbersome model to a lighter-weight model so that the small model can replicate the result. KD has been used in the recent NLP research, such as text classification and sequence labeling (Yang et al., 2020) and got the promising result. However, due to the limitation of data size and robust model, the application KD is not flexible to some extent. To resolve these issues, Self Distillation (SD) (Yuan et al., 2019) is proposed, where the student model is used as the teacher model as

well. Results show that SD can almost replicate the accuracy regardless of a well-trained large model or big dataset such as in the image classification task (Zhang et al., 2019). SD has also been applied to NLP tasks such as language model and neural machine translation (Hahn and Choi, 2019) and obtains promising results.

Despite obtaining higher accuracy and better performance, modern deep learning models face drawbacks of miscalibration and overconfidence (Müller et al., 2019; Naeini et al., 2015; Lakshminarayanan et al., 2016). Recent studies resolve this issue by using techniques like label smoothing (Müller et al., 2019) and temperature scaling (Naeini et al., 2015), and Dirichlet calibration (Kull et al., 2019). These works show that the well-calibrated model can improve the model performance as well as feature representation. As for the NLP downstream tasks, research has shown that calibration benefits both sentence quality and length in the sentence classification (Jung et al., 2020), and helps to improve the model fine-tuning in text generation (Kong et al., 2020).

As transfer learning techniques and calibration contributes to NLP tasks, we investigate the correlation of improving calibrated feature representation with ULMFiT and SD. Label smoothing is integrated with ULMFiT to extract significant features from language modeling. To improve KD by recalibrating predicted probability, we incorporate temperature scaling (TS) with knowledge distillation loss. We also observe the correlation of a well-calibrated trained network in whole model fine-tuning. We conduct extensive experiments to validate our observations with two datasets of (1) the consultation back pain and (2) medical dialogue. Results show that a well-calibrated model is highly correlated with ULMFiT and SD, as well as fine-tuning, in terms of both accuracy and calibration error.

Our contributions can be concluded as following:
(1) We introduce the calibrated ULMFiT (CULMFiT) by applying label smoothing on conventional ULMFiT. Results are showing that the CULMFiT outperforms the vanilla ULMFiT, proving the impact of calibration of language modeling.
(2) We measure optimal calibrated temperature and replace the fixed temperature value in KD loss and demonstrate that calibrated temperature outperforms the fixed value.
(3) We incorporate temperature scaling with the whole model fine-tuning and observe that calibration benefits model performance and uncertainty.
(4) We build the consultation backpain dataset, consisting of patients' queries and clinicians' responses into conversational pairs.

2 Proposed Method

2.1 Preliminaries

ULMFiT Natural Language Processing has picked up the pace in recent years and caught researchers' attention greatly, essentially attributed to the conquer of inductive transfer learning, which was seen as the major obstacle that NLP was lagging behind Computer Vision (CV). Universal Language Fine Tuning (ULMFiT) was proposed (Howard and Ruder, 2018) as obtaining the success of passing the acquired knowledge of pre-trained model to other similar tasks. ULMFiT is to pretrain the model on a large general domain corpus such as Wikipedia data, then fine-tune it on the target tasks. As a source task trained with a large corpus, the pre-trained language model can capture most facets and contexts of the data, which is ideal for NLP downstream tasks. Hence including Text Classification that ULFMiT was firstly introduced with, it gets great success and applied in almost all NLP fields. It is believed that with the language model trained on the large-scale data, the model with small or medium data will also replicate similar results to the vanilla model.

Label Smoothing (LS) Label smoothing has been widely applied in various fields of deep learning, such as image classification (Real et al., 2019) and speech recognition (Chorowski and Jaitly, 2016). It achieves promising results since Szegedy et al. (Szegedy et al., 2016) first introduced it, then gets further development after the extension explanation on its mechanism of how it improves the model calibration (Müller et al., 2019). As the regularization technique to tackle the overconfidence of a model, label smoothing softens the one-hot labels in the penultimate layer's logit vectors, to improve the calibration and further help the robustness and reliability of the model. Here is the mathematical illustration of label smoothing: suppose $\hat{p}_c$ is the probability and p_c is the ground truth of the c-th class, where p_c is 1 for the correct class and 0 for the rest classes, the cross-entropy loss of network trained with a hard target can be

demonstrated as: $CE = -\sum_{c=1}^{C} p_c \log(\hat{p}_c)$.For a network trained with a label smoothing hyper-parameter α, the one-hot true value will be clipped as: $p_c^{LS} = p_c(1-\alpha) + \alpha/C$. Hence the cross-entropy loss with label smoothing can be illustrated as:

$$CE^{LS} = -\sum_{c=1}^{C} p_c^{LS} \log(\hat{p}_c). \quad (1)$$

Temperature Scaling (TS) It has been observed that most of the modern neural networks are poorly calibrated even with a high confidence score. To solve this issue and make the model better calibrated, among all possible factors that may influence the calibration, temperature scaling (TS), as a straightforward extension of Platt Scaling, has been verified as the most efficient and least time-consuming and computationally expensive way (Guo et al., 2017). A single scalar T (T>1) called temperature is applied on the logit then it passes to the softmax function (denoted as σ), which will not change the maximum value in it, so the prediction remains intact. Here is the equation for TS given the logit vector:

$$\hat{p_c}^{TS} = \max_c \sigma_(logit_c/T)^{(c)}. \quad (2)$$

Self-Distillation (SD) Knowledge distillation (KD) targets compressing a cumbersome teacher model into a lighter-weight student model. The distilled model can still replicate similar or better accuracy due to the privileged information captured by the teacher model. Suppose the logits for teacher model and student model are $logit^T$ and $logit^S$, and fixed T value as T^{fix}, the loss function with of Kullback-Leibler divergence (KL divergence) L_{KD} can be formulated as:

$$L_{KD} = \sum \mathrm{KL}\Big(\sigma\big(\frac{logit^T}{T^{fix}}\big), \sigma\big(\frac{logit^S}{T^{fix}}\big)\Big) \quad (3)$$

It is generally believed that the teacher model should be well-trained with a large corpus and has a bigger capacity than the student model. However, the insufficiency of the dataset and the untrustworthiness of the model are substantial restrictions to KD. Yuan et al (Yuan et al., 2019) argue that the student model can achieve similar results with a poor-trained or smaller teacher model, even under the circumstance of no teacher model, which is called self-distillation (SD). By making the model be their own teacher, SD is to train the student model first to get a pre-trained model, then using it as the teacher to train itself. It has been further proved the positive effect that self-distillation has on calibration (Zhang and Sabuncu, 2020).

2.2 ULMFiT with Label Smoothing

ULMFiT has obtained great success in NLP tasks as it transfers information from the pre-trained model to the target application domain, and LS helps in calibration and better uncertainty. We apply LS to ULMFiT to gain a calibrated ULMFiT (CULMFiT) to further improve the feature representation and extract more distinctive information from language modeling. Given θ^{ULMFiT} is the pre-trained ULMFiT weight, x as the input of the conversational model, the loss function of ULMFiT with LS can be written as follows:

$$CE_U^{LS} = -\sum_{c=1}^{C} p_c^{LS} \log(\hat{p_c}|x, \theta_U). \quad (4)$$

2.3 Self-Distillation with TS

Self-distillation (SD) has been proved to replicate the similar accuracy as the knowledge distillation (KD) with the teacher model training on student model, and temperature scaling helps to prevent miscalibration. We integrate TS on SD to attain a well-calibrated distilled model. For this purpose, we adopt KD loss of KL divergence with calibration as in the paper (Hinton et al., 2015). However, temperature set as a scalar value is a similar technique as network calibration, and the optimal temperature is expected to be a better option. In our work, we measure optimal T and assign it to the KD, aiming at preventing inappropriate calibration and investigating the relation between calibration and SD. Suppose the logits for the teacher model and student model are $logit^T$ and $logit^S$, and the optimal temperature is T^{opt}. The loss function with KL divergence L_{KD} can be formulated as:

$$L_{SD} = \sum \mathrm{KL}\Big(\sigma\big(\frac{logit^T}{T^{opt}}\big), \sigma\big(\frac{logit^S}{T^{opt}}\big)\Big) \quad (5)$$

The final loss L can be demonstrated as:

$$L = L_{SD} + L_{CE} \quad (6)$$

2.4 Fine-tuning with TS

As an approach of transfer learning, fine-tuning can propagate the acquired knowledge from one domain to another and enhances the learning capacity.

Table 1: Samples of the backpain dataset.

ID		Medical Dialogue
0	Enquiry	What is musculoskeletal pain condition?
	Reply	A great change of lifestyle and behaviour, such as too much workload, adjustments in the workplace, work breaks and sudden exercise would improvement of musculoskeletal pain.
1	Enquiry	Why my foot pain cause back pain?
	Reply	The possible reason is your spine's alignment or overstressing lower back muscles
2	Enquiry	The back pain cause me unable to carry groceries, what should I do?
	Reply	Try the grocery delivery or ask help from your close family or friends. If it is severe, contact your clinician immediately.
3	Enquiry	Will back pain influence the enjoyment between couples?
	Reply	Yes, studies have shown that higher lever of back pain can impair the leisure activities with the spouse.
4	Enquiry	I feel pain in my joints after exercise, what is the problem?
	Reply	If your joint feels particularly painful afterwards for longer than two hours after an exercise session, reduce the intensity of your next exercise session.

On the other hand, TS produces a well-calibrated confidence score. To further improve the information transformation and feature representation, we apply TS to the logit for cross-entropy loss calculation while fine-tuning the entire model. Given p_c^{TS} is the temperature scaled logit (as shown in formula 2), the loss function with TS can be illustrated as:

$$CE^{TS} = -\sum_{c=1}^{C} p_c \log(\hat{p}_c^{TS}). \quad (7)$$

3 Experiments

3.1 Datasets

Backpain Dataset To develop an evidence-based skillful conversational model, we collect the backpain dataset with pairs of the query from a patient and the response from a clinician. Table 1 shows samples of conversational pairs. Sources of queries are various sites people would generally ask health-related questions, such as Google and Quora, and responses are collated from either peer-reviewed journal articles (Hayden et al., 2005) (Henschke et al., 2010) (Cagnie et al., 2007) (Scheermesser et al., 2012) (Choi et al., 2010) (Van Dam et al., 2018) or other sources recognized for providing valid health advice and suggestions like NHS website [1]. It covers five highly related factors that cause back pain, namely sleep, mental health, exercise, nutrition, and social and environmental factors. The dataset contains 1000 conversational pairs for the train set and 200 pairs for the validation set, and the minimum and maximum length of the reply are 16 and 40.

[1] https://www.nhs.uk/conditions/back-pain/

MedDialog Due to the disadvantages of the small volume of our backpain dataset, we also use the MedDialog Dataset (Zeng et al., 2020) to further testify our hypothesis of calibration. It consists of conversational pairs of symptoms description from patients and follow-up questions and diagnoses from doctors, which covers various medical fields such as pathology and family medicine. We randomly divide the dataset into train and validation set with the ratio of 0.8 and 0.2.

3.2 Implementation Details

We choose the well-known transformer model as the benchmark in our project. The language modeling architecture for ULMFiT is the encoder part of the Transformer with Fully-Connected (FC) Layers, and the loss function is cross-entropy loss with label smoothing. To fine-tune the proposed model, we first get the optimal TS value, then apply it to re-calibrate the logit for the trained model. The GPU of Nvidia Tesla T4 with the memory of 16GB is used to conduct all the experiments in this work. The dataset is split with 0.8 and 0.2 for training and validation. All experiments are conducted with the Adam optimizer, 0.01 as the learning rate and batch size of 4. The best BLEU-1 score metric is used to find the best epoch.

4 Experiments

4.1 Results

4.1.1 Evaluation Metrics

We use the uni-gram similarity metrics BLEU-1 as the major evaluation for our dialogue system. To measure the word overlapping between the ground truth and prediction, we also apply Metric for Evaluation of Translation with Explicit Ordering (ME-

Table 2: Results of Backpain Dataset. Annotations of experimental models are as following: the vanilla transformer model and ULMFiT are labeled as Baseline; ULMFiT with label smoothing as CULMFiT; model with ULMFiT and fine tune with TS as Fine-tune.

Method	Model	BLEU-1	Perplexity	METEOR	ECE
Baseline	Transformer	0.4292	7.9895	0.4079	0.3702
	ULMFiT	0.4321	8.0603	0.4218	0.3764
LS	CULMFiT	**0.4632**	5.6155	**0.4552**	0.3674
TS	Fine-tune	0.4415	**5.2797**	0.4268	**0.2884**

Table 3: Results of MedDialog Dataset: the vanilla transformer model and ULMFiT are annotated as Baseline; ULMFiT is the Transformer model trained with the Medical Dialogue Dataset; regularized ULMFiT is annotated as CULMFiT; the proposed model fine tuned with TS is Fine-tune.

Method	Model	BLEU-1	Perplexity	METEOR	ECE
Baseline	Transformer	0.3387	11.5422	0.2280	0.2611
	ULMFiT	0.3609	**7.9134**	0.2556	0.3519
LS	CULMFiT	**0.3765**	10.2346	0.2578	0.3734
TS	Fine-tune	0.3747	12.6997	**0.2618**	**0.0580**

TEOR) metric (Banerjee and Lavie, 2005) in our work. Perplexity, as the measurement of model uncertainty to the training data, is calculated based on the cross-entropy loss for each sample. We use the Expected Calibration Error (ECE) (Naeini et al., 2015) to check the efficiency of calibration techniques. ECE divides predictions into N equally-spaced bins and takes the weighted mean of each bin's confidence gap. We choose N=15 bins in our work.

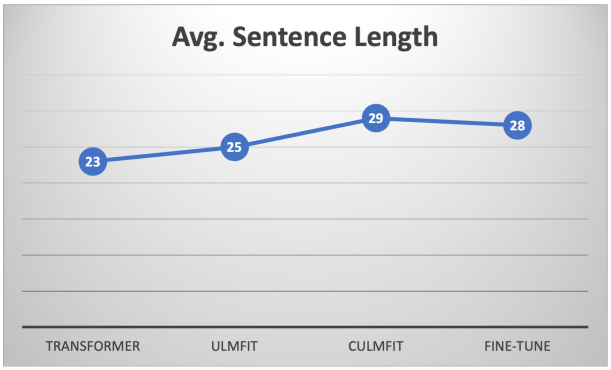


Figure 1: Average sentence length generated by each model in the backpain dataset.

4.1.2 Evaluation of the Backpain Dataset

The results of the dialogue system trained with the consultation backpain dataset are shown in table 2 and examples of generated responses are demonstrated in table 4. The calibrated ULMFiT with LS (CULMFiT) significantly outperforms the baseline transformer model by improving the BLEU-1 score by about 3.8%, and exceed the vanilla ULMFiT by approximately 1.5%. On the other hand, the fine-tuning TS improves both BLEU-1 score and ECE with 1% and 8%, respectively. Though the fine-tuning with TS does not provide the best BLEU-1 score, it provides the best calibrated confidence score with the lowest ECE. In terms of generated sentence length and quality, our proposed models of CULMFiT and fine-tuning outperform baseline models of transformer and ULMFiT. The diagram 1 illustrates that on average, the generated conversation length of CULMFiT and fine-tuning is longer than those from the benchmark models, where CULMFiT model produces the longest responses. Furthermore, the proposed models generate more logical and meaningful sentences. For example, in the first sample in table 4, the CULMFiT network predicts the verb "*stand*" and the fine-tuning model generates the phrase "*a short period every hour*" that exactly matches the ground truth, which makes the response more accurate for the symptom. Overall, CULMFiT demonstrates the best performance on most of the evaluation metrics. Evaluation results prove the effectiveness of proposed calibration techniques with the ULMFiT and fine-tuning on the probability and correctness adjustment.

4.1.3 Evaluation of MedDialog Dataset

To further verify the hypothesis that calibration benefits the model performance in both accuracy and robustness, we replicate the previous experi-

Table 4: Visualization of predicted responses. Query is the input and GT is the ground truth. Prediction is the response generated by the model.

	Sample		Prediction
Backpain	Query: What to do to solve long time sitting issue except exercise? GT: stand up and move about gently for a short period every hour would help relieve the muscle stiffness	Transformer	up and move for a short time would help the stiffness.
		ULMFiT	be up and move for a short time every hour would help muscle stiffness.
		CULMFiT	stand up and move around gently for a short period every hour would help to solve the muscle stiffness.
		Fine-tune	be up and move gently for a short period every hour would help the muscle stiffness.
MedDialog	Query: Hi doctor, I am 25 years old and I have a history of hair fall for almost 5 years. I am very concerned of it. Could you give me some advice and possible treatment? GT: Hi, as per you case history of hair fall, my treatment advice is to take good nutritious diet full of green leafy vegetables and milk, and to use a good herbal shampoo and coconut hair oil.	Transformer	Hello, per your case of hair fall, my advice be good diet with vegetables and milk, use shampoo and oil out of it.
		ULMFiT	Hello, per your case history of hair fall, my advice be take good herbal diet full of green leafy vegetables and milk, use a good shampoo and oil for it.
		CULMFiT	Hi, per you case history of hair fall, my treatment advice is to take nutritious diet of green vegetables and milk, and to use a good herbal shampoo and hair oil.
		Fine-tune	Hi, per you case history of hair fall, my treatment advice is to take herbal diet of green vegetables and milk, and to use a good herbal shampoo and green herbal oil.

Table 5: Results of self-distillation with Backpain dataset. Three methods are applied in this experiment: without self-distillation (standalone), self-distillation with a fixed value of TS (SD Fixed TS), and self-distillation with optimal TS (SD optimal TS). The fixed TS is 2. The optimal TS for the transformer model and CULMFiT is 3.025 and 4.789 respectively.

Method	Model	BLEU-1	Perplexity	METEOR	ECE
Standalone	Transformer	0.4292	7.9895	0.4079	0.3702
	CULMFiT	**0.4632**	**5.6155**	**0.4552**	0.3674
SD Fixed TS	Transformer	0.4331	7.8329	0.4221	0.3820
	CULMFiT	0.4236	6.3934	0.4135	0.1962
SD Optimal TS	Transformer	0.4334	7.8010	0.4187	0.3703
	CULMFiT	0.4473	5.8486	0.4402	**0.1788**

ments on the Medical Dialogue Dataset. The results of various evaluation metrics are illustrated in table 3, and the sample visualization is shown in 4. All results are mostly consistent with the previous experiments. For example, in the sample illustrated in the table 4, the length of predicted sentences from CULMFiT and fine-tuning model is longer than the baseline models. Besides, the adjective "*herbal*" for the noun "*shampoo*" from the proposed models can better explain the type of shampoo product, which makes the response more specific for the patient's inquiry. Overall, the proposed methodologies illustrate superior performance in most of the evaluation metrics. The calibrated ULMFiT (CULMFiT) with LS outperforms the benchmark and the vanilla ULMFiT by about 4% and 1.5% increment of BLEU-1 score correspondingly. The fine-tuning with the TS model significantly improves ECE by about 35%. Results from both experiments prove that calibration tech-

niques of LS and TS help to improve the robustness and uncertainty of the model.

4.1.4 Evaluation of Self-Distillation With TS

One of our observations is that the SD model with the optimal TS outperforms the one with fixed TS. All results are shown in table 5. We select the benchmark transformer model and the model with the calibrated ULMFiT in this experiment. It has been shown that SD with the optimal T value obtains better performance than with the fixed T (with $T = 4$) value for image classification (Hinton et al., 2015). Hence in our work, we also compare the SD with fixed T and optimal T applied in both benchmark and proposed model. To select the best fixed T value, we apply T values of 1.5, 2, 3, 4, and 5 and choose the one with the best BLEU-1 score. The diagram 2 indicates that $T = 2$ provides the best BLEU-1 score. Compared to the standalone, SD with fixed and optimal T of transformer and CULMFiT models in table 5, CLUMFiT without SD obtains the best BLEU-1 score, perplexity, and METEOR, while SD with optimal TS provides the best ECE. On the other hand, CULMFiT gets hampered with calibration, which has been evinced in the work (Müller et al., 2019). Overall, the performance of the model trained with optimal TS beats the one with fixed TS.

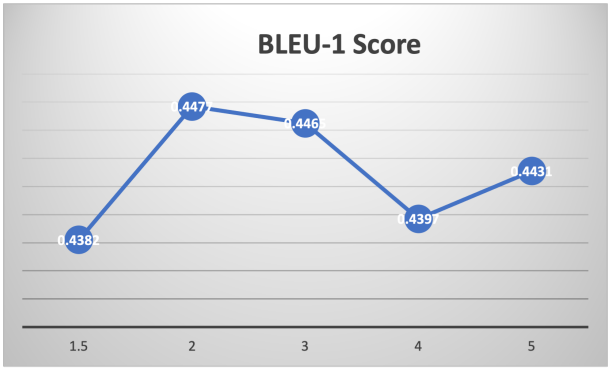


Figure 2: BLEU-1 score with different T values.

5 Discussion

In this paper, we apply calibration techniques of LS and TS to develop the medical dialogue system and get promising results. Table 2 and 3 are showing results that the well-calibrated model benefits ULMFiT, SD and fine-tuning. Table 5 demonstrates the observation of self-distillation on fixed and optimal temperature scaling. All our observations is presented with the sample visualization in table 4. Overall, the ULMFiT with LS provides the best BLEU-1 score and the fine-tuning TS improves the ECE mostly, which is consistent with experiments in both datasets. Despite the higher model performance in both accuracy and calibration, fine-tuning is a two-stage training, which can cause an additional computational burden. Even though LS and TS introduce additional computational parameters, the computational cost is negligible. On the other hand, ULMFiT with label smoothing hurts SD, which has been reported in (Müller et al., 2019).

6 Conclusion

In this paper, we propose the calibrated ULMFiT, self-distillation and fine-tuning to build a medical dialogue system. Label smoothing and temperature scaling are utilized to obtain calibrated network and improve the performance in terms of accuracy and robustness. We empirically demonstrate calibration is highly co-related with ULMFiT, SD and fine-tuning, which has been presented in table 2, 3,4 and 5. For future work, we will explore the calibration and knowledge-distillation impact on other NLP downstream tasks like Neural Machine Translation and Sentiment Analysis.

References

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, Ann Arbor, Michigan. Association for Computational Linguistics.

Barbara Cagnie, Lieven Danneels, Damien Van Tiggelen, Veerle De Loose, and Dirk Cambier. 2007. Individual and work related risk factors for neck pain among office workers: a cross sectional study. *European Spine Journal*, 16(5):679–686.

Brian KL Choi, Jos H Verbeek, Wilson Wai-San Tam, and Johnny Y Jiang. 2010. Exercises for prevention of recurrences of low-back pain. *Cochrane Database of Systematic Reviews*, (1).

Jan Chorowski and Navdeep Jaitly. 2016. Towards better decoding and language model integration in sequence to sequence models. *arXiv preprint arXiv:1612.02695*.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. *arXiv preprint arXiv:1706.04599*.

Sangchul Hahn and Heeyoul Choi. 2019. Self-knowledge distillation in natural language processing. *arXiv preprint arXiv:1908.01851*.

Jill Hayden, Maurits W Van Tulder, Antti Malmivaara, and Bart W Koes. 2005. Exercise therapy for treatment of non-specific low back pain. *Cochrane database of systematic reviews*, (3).

Nicholas Henschke, Raymond WJG Ostelo, Maurits W van Tulder, Johan WS Vlaeyen, Stephen Morley, Willem JJ Assendelft, and Chris J Main. 2010. Behavioural treatment for chronic low-back pain. *Cochrane database of systematic reviews*, (7).

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Taehee Jung, Dongyeop Kang, Hua Cheng, Lucas Mentch, and Thomas Schaaf. 2020. Posterior calibrated training on sentence classification tasks. *arXiv preprint arXiv:2004.14500*.

Lingkai Kong, Haoming Jiang, Yuchen Zhuang, Jie Lyu, Tuo Zhao, and Chao Zhang. 2020. Calibrated language model fine-tuning for in-and out-of-distribution data. *arXiv preprint arXiv:2010.11506*.

Meelis Kull, Miquel Perello-Nieto, Markus Kängsepp, Hao Song, Peter Flach, et al. 2019. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. *arXiv preprint arXiv:1910.12656*.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2016. Simple and scalable predictive uncertainty estimation using deep ensembles. *arXiv preprint arXiv:1612.01474*.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. 2019. When does label smoothing help? In *Advances in Neural Information Processing Systems*, pages 4694–4703.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29.

Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. 2019. Regularized evolution for image classifier architecture search. In *Proceedings of the aaai conference on artificial intelligence*, volume 33, pages 4780–4789.

Mandy Scheermesser, Stefan Bachmann, Astrid Schämann, Peter Oesch, and Jan Kool. 2012. A qualitative study on the role of cultural background in patients' perspectives on rehabilitation. *BMC musculoskeletal disorders*, 13(1):1–13.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Nicholas T Van Dam, Marieke K van Vugt, David R Vago, Laura Schmalzl, Clifford D Saron, Andrew Olendzki, Ted Meissner, Sara W Lazar, Catherine E Kerr, Jolie Gorchov, et al. 2018. Mind the hype: A critical evaluation and prescriptive agenda for research on mindfulness and meditation. *Perspectives on psychological science*, 13(1):36–61.

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.

Ziqing Yang, Yiming Cui, Zhipeng Chen, Wanxiang Che, Ting Liu, Shijin Wang, and Guoping Hu. 2020. Textbrewer: An open-source knowledge distillation toolkit for natural language processing. *arXiv preprint arXiv:2002.12620*.

Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. 2019. Revisit knowledge distillation: a teacher-free framework. *arXiv preprint arXiv:1909.11723*.

Guangtao Zeng, Wenmian Yang, Zeqian Ju, Yue Yang, Sicheng Wang, Ruisi Zhang, Meng Zhou, Jiaqi Zeng, Xiangyu Dong, Ruoyu Zhang, et al. 2020. Meddialog: A large-scale medical dialogue dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9241–9250.

Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. 2019. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3713–3722.

Zhilu Zhang and Mert R Sabuncu. 2020. Self-distillation as instance-specific label smoothing. *arXiv preprint arXiv:2006.05065*.

Automated Recognition of Hindi Word Audio Clips for Indian Children using Clustering-based Filters and Binary Classifier

Anuj Gopal
Pratham Education Foundation
anuj.gopal@pratham.org

Abstract

Speech recognition systems have made remarkable progress in the last few decades but most of the work has been done for adult speech. The rise of online learning during Covid-19 pandemic highlights the need for voice-enabled assistants for children so that they can navigate the menus and interfaces seamlessly. Speech recognition for children will also be very useful to develop automated reading assessment tools. However, such technology for children is challenging for a country like India where differences in accents, diction and enunciation is significant but available children speech data is limited. Through this paper, I tried various approaches to recognize hindi word audios. Commercially available Google Speech-to-Text performs poorly with only 49.7% accuracy at recall of 0.24 while recognising audio samples containing hindi words spoken by children. Using the same dataset, I experimented with clustering algorithm and logistic regression and found that the accuracy improves upto 81% with logistic regression. The paper also highlights the importance of data preprocessing by performing noise reduction using Butterworth low pass filters.

Keywords: *EdTech, Reading Skills, Assessment, Speech processing,Voice Command*

1 Introduction

According to a study by KPMG (2017), India's online education is set to grow to 9.6 million users by 2021. Online device-enabled education has been on a growth ever since the pandemic hit and it is estimated to rise even more in the coming years, with colleges and universities modifying their systems in accordance with the technological surge and Edtech companies investing heavily on creating platforms for the education of the future generation. According to Research and Markets (2019), the online education market will reach $350 Billion by 2025. This highlights the need for designing systems for children to have frictionless interactions with technology. Voice-enabled systems can play a crucial role in accessibility of education systems at an early age but recognizing words spoken by children is still a challenge. There has been limited progress in speech technology for children due to limited data.

Moreover, Speech recognition technology can play an important role in automating reading level assessments of children. With voice-enabled devices, the complete education system can be better controlled and automated, leading to high quality teaching and learning(Motiwalla, 2009; Azeta et al., 2010), especially for children as they are at a crucial stage of development.

This paper presents my experience with assessment of word reading audios by developing an automated assessment system for one of the low resource languages - Hindi which is the medium of instruction in government schools in many states of India. The children were of the age group 8-14 years from rural areas of India. The proposed Binary Classifier model uses Logistic Regression as the algorithm, along with preprocessing techniques using K-means clustering and Butterworth Low Pass filters to obtain an overall validation accuracy of 81.53%.The main contributions of this paper are:

- A robust speech recognition methodology based on Binary Classifier catering to the needs of Indian regional languages like Hindi, Marathi etc.

- A preprocessing based audio classification methodology specifically designed for children which can be incorporated into existing education system to enhance reading level of children

2 Related Works

A multitude of similar works have been done in the past. In this section, I summarize previous work related to speech recognition for children and/or for Indian regional languages:

2.1 Highly accurate children's speech recognition for interactive reading tutors using subword units(Hagen et al., 2007)

This paper presents methodologies for advancement in speech recognition in the context of an interactive literacy tutor for children that aims to improve accuracy and modelling capabilities. To improve oral reading recognition, a more focused approach towards a novel set of speech recognition techniques is presented, where they have also shown that error rates for interactive read aloud can be reduced by more than 50% through a combination of advances in both statistical language and acoustic modeling.The efficacy of the approach is demonstrated using data collected from children in grades 3–5, namely 34.6% of partial words with reasonable evidence in the speech signal are detected at a low false alarm rate of 0.5%

2.2 Automatic Speech Recognition Systems for Regional Languages in India(Bachate and Sharma, 2019)

This paper discusses various aspects of building an automated speech recognition system, the parameters affecting the development of speech recognition systems, tools and techniques used and also research done on regional languages. The paper concludes that the Deep Neural network provides a better and a more accurate way of recognising speech.

2.3 ASRoIL: a comprehensive survey for automatic speech recognition of Indian languages(Singh et al., 2020)

This paper provides a systematic survey of the existing literature related to automatic speech recognition (i.e. speech to text) for Indian languages. The survey analyses the possible opportunities, challenges, techniques, methods and to locate, appraise and synthesize the evidence from studies to provide empirical answers to the scientific questions. The survey was conducted based on the relevant research articles published from 2000 to 2018. The purpose of this systematic survey is to sum up the best available research on automatic speech recognition of Indian languages that is done by synthesizing the results of several studies.

2.4 Hindi speech recognition using time delay neural network acoustic modeling with i-vector adaptation(Kumar and Aggarwal, 2020)

In this work, they have selected the Time-delay Neural Network (TDNN) based acoustic modeling with i-vector adaptation for limited resource Hindi ASR. The TDNN can capture the extended temporal context of acoustic events. To reduce the training time, they used sub-sampling based TDNN architecture in this work. Further, data augmentation techniques have been applied to extend the size of training data developed by TIFR, Mumbai. The results show that data augmentation significantly improves the performance of the Hindi ASR. Further, 4% average improvement has been recorded by applying i-vector adaptation in this work. They found the best system accuracy of 89.9% with TDNN based acoustic modeling with i-vector adaptation.

2.5 Syllable based Hindi speech recognition(Bhatt et al. ,2021)

In this paper, one of the acoustic units of speech, the syllable, is used for the development of a continuous Hindi speech recognition system. Earlier research works related to Hindi speech recognition were performed using the word, phoneme, and context-dependent models. The authors proposed a syllable based Hindi speech recognition system in this study due to different advantages of syllable units such as longer acoustic units, fast decoding, reducing contextual effects, and reduction of irregularities due to phonemes. The continuous Hindi speech recognition system was developed utilizing syllable based acoustic units. The research outcomes reveal that by using syllables, the performance of the system was increased by 27% than phoneme and 20% than triphones.

3 Dataset

The dataset used was collected using a novel data collection methodology (Agarwal et al.,2020) where an Android app was designed to collect human evaluated training data to collect data from three states of India - Maharashtra, Uttar Pradesh and Rajasthan. While conducting the test, the audio clips were recorded for each section and the

assessor marked the correctness of every question. The details of the test were further recorded in a json file. Although dataset contains letters, paragraphs and stories too, I only used hindi words for my analysis. The details of dataset is shown in the following table:

No. of audio files	1071
Unique word count	37
Total Duration	51.56 mins
Avg Duration	3.13s
Total FileSize	415.29MB
Avg FileSize	430.43KB
Avg Unique Speakers	30

Table 1: Statistics of dataset used

I have also tried running the model using few audio files from other word samples to be used as 'False' so that there is an increase in the training dataset. But the accuracy obtained is lower than the one without using those audio files. The possible reason for this might be the difference in pattern learning for different words, which primarily depends on their frequency response and temporal variations. Finally, the dataset was divided into 70% Training and 30% Testing data.

4 Methodology

The proposed methodology utilised three different interconnected algorithmic approaches to obtain optimum results for the problem proposed:

4.1 Audio Quality based data segregation using Clustering algorithms

The clustering of audio files is important for understanding the features of the signals, as the audio signals are nothing but high-dimensional data. As presented in EURASIP research(Lil et al., 2017), distance calculation failures, inefficient index trees, and cluster overlaps, derived from the equidistance, redundant attribute, and sparsity, respectively, seriously affect the clustering performance.

Mel-frequency cepstrum coefficient(MFCC) is one of the most important features of any audio signal, which is extracted from the files using a multistep process starting with signal windowing followed by Direct Fourier Transformation. The next steps include taking the log of the magnitude, and then warping the frequencies on a Mel scale, followed by applying the inverse Discrete Cosine Transform.

I propose a K-means clustering for segregation of inaudible/low quality audio clips using the means of MFCC values for each of the sound signals.The MFCC features of a single audio file were clustered to observe differences in patterns within the audio, to detect human voice and silences, and to observe other variations. This can not only remove noise and silences as a preprocessing part, but can also provide valuable patterns for the whole audio file.There are differences in features for the two files as expected from audios as well.

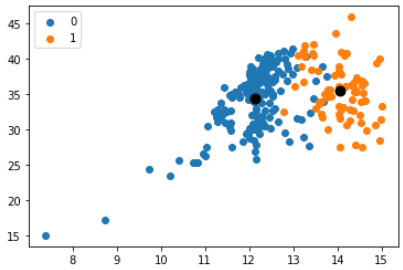


Figure 1:Plotting MFCC using 2 coefficients

The average value of mean MFCC for audio files that were Google STT transcripted is 1.23 while that for empty audio files is 0.72.

The K-means clustering method with reduced MFCC values [taking means] provided us with the sample set that can be directly fed to models. And, since plots of using MFCC mean values show patterns as shown, I used 264 audio files with label 0 as shown here, to infer that the accuracies are positively enhanced as the clusters are formed based on energy/frequency ranges that segregated bad quality audio files to separate clusters.

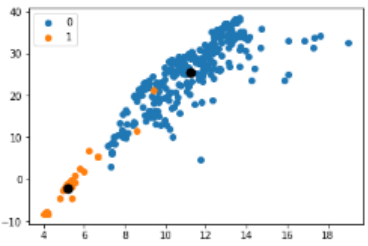


Figure 2: K-means clustering plot

4.2 Noise Reduction using Butterworth Low Pass Filters

The Butterworth filters are designed for signal processing which have flat frequency response in the pass-band and in the stop-band they have a zero roll off response. They are widely used filters in video motion analysis and in audio signals.The filter-frequency adjustability allows the user to remove noise without rolling off the signal. Variable low-pass filtering deals with aliasing by passing no more than 1% of the high-frequency content that causes aliases. This ensures that only the

aliased frequencies are removed, not the signal of interest.Tunable digital filters are widely used in medical electronics, digital audio instrumentation, telecommunications and control systems. It is often the essential requirement for removal of noise from the audio signal.

4.3 Logistic Regression based Binary Classifiers

Normal logistic regression maximizes the following log-likelihood function:

$$\ell(\beta) = \Sigma[y_i \log(\pi_i) + (1 - y_i) \log(1 - \pi_i)]$$

Since smaller datasets are more prone to overfitting and have outliers which can cause significant vulnerability in the model, regularization techniques such as Ridge regression and Lasso regression are used along with tuned Logistic Regression to add penalties which create a balance between producing accurate predictions and producing smaller coefficients. The objective of any model is to learn patterns from the available training dataset and generalise relationships between the dependent and the independent variables. At the same time, the model should be biased towards the training dataset and therefore a general pattern recognition approach is preferred over more specifically accurate models compatible only with the dataset provided to train. Complex models perform worse since they have a higher possibility of overfitting according to the dataset.

5 Experimental Results

The summarised experimental results are shown below for 4 samples of different words. The table contains word samples, their total instances used for the analysis and accuracy obtained using various machine learning models:

Word	Total	SVC	Logistic
लाल	66	0.70	0.85
ताला	57	0.78	0.89
नाक	42	0.60	0.80
आग	57	0.78	0.89

Table 2: Accuracy of ML models

The Logistic Regression with aforementioned preprocessing techniques provides the best results. The overall accuracy of the proposed method is 81.53% with a precision of 0.84 and a recall of 0.94.

6 Error Analysis

The small dataset poses a variety of problems in the modeling.The first is lack of available information, which produces less accurate models. Secondly, the small dataset does not reflect the population's distribution in an accurate way, thus creating a necessary condition to preprocess datasets and work sensitively to remove any kind of noise in the data. With audio data, the problem becomes more as the pattern recognition techniques available for audio datasets are relatively new and most of them use an indirect methodology of learning, converting files to required formats to extract necessary information. Getting closer to true population parameters, thus, becomes extremely important. The fluctuations in accuracy measures as the dataset reduces in number becomes extremely vulnerable to variations below 200 observations(Canario, 2020).

Using regularization techniques to penalise predictions leading to overfitting, I generalised Logistic Binary Classification to optimum requirements. The regularization techniques used was Ridge Regression which works well to reduce overfitting, optimizing the desired performance(Salehi et al., 2019), to finally develop a parsimonious model, creating a less complex, yet a more accurate model. Limited-memory Broyden–Fletcher–Goldfarb–Shanno Algorithm (lbfgs) was used as the solver for Logistic Regression as it performs best for small datasets, using low memory and utilising Quasi-Newton methodology(Dennis and Moree, 1977).

7 Conclusion

In this paper, I have presented Audio Quality based data segregation using Clustering algorithms, followed by Noise Reduction using Butterworth Low Pass Filters and finally Logistic Regression based Binary Classifiers to achieve optimum speech recognition results for dataset containing audio clips for Indian Children reading Hindi words as part of their reading level assessment. While Google Speech-to-Text could transcribe 49.7% of the audio with a recall of 0.24, our method of Logistic Regression based Binary Classifier with preprocessing resulted in an overall validation accuracy of 81.53%. This can be used to develop advanced learning tools for children in the future and to also significantly enhance Voice command based devices specific to regional languages.

8 References

Ambrose Azeta, Charles Ayo, Prof. Aderemi Atayero, and Nicholas Omoregbe. 2010. Intelligent Voice-based E-education system: A framework and evaluation. *International Journal of Computing*. 9. 327-334.

Amitoj Singh, Virender Kadyan, Munish Kumar, and Nancy Bassan. 2020. ASRoIL: a comprehensive survey for automatic speech recognition of Indian languages. *Artificial Intelligence Review*, 53(5), pp.3673-3704.

Andreas Hagen, Bryan Pellom, and Ronald Cole. 2007. Highly accurate children's speech recognition for interactive reading tutors using subword units. *Speech Communication*, 49(12), pp.861-873.

Ankit Kumar and Rajesh Kumar Aggarwal. 2020. Hindi speech recognition using time delay neural network acoustic modeling with i-vector adaptation. *International Journal of Speech Technology*, pp.1-12.

Dolly Agarwal, Jayant Gupchup, and Nishant Baghel. 2020. A Dataset for measuring reading levels in India at scale. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 9210-9214). IEEE.

Fariborz Salehi, Ehsan Abbasi, and Babak Hassibi. 2019. The impact of regularization on high-dimensional logistic regression. arXiv preprint arXiv:1906.03761.

John E. Dennis, Jr. and Jorge J. Moree. 1977. Quasi-Newton methods, motivation and theory. SIAM review, 19(1), pp.46-89.

KPMG in India and Google, 2017. *Online Education in India: 2021*

Luvai F. Motiwalla. 2009. A Voice-Enabled E-Learning Service (VòIS) Platform. In *Fifth International Conference on Networking and Services 2009*, (pp. 597-602). IEEE.

Ravindra Parshuram Bachate and Ashok Sharma. 2019. Automatic speech recognition systems for regional languages in India. *International Journal of Recent Technology and Engineering*, 8(2).

Remy Canario, 2020. *The Best Classifier for Small Datasets: Log-F(m,m) Logit*

Research and Markets, 2019. *Online Education Market Global Forecast, by End User, Learning Mode (Self-Paced, Instructor Led), Technology, Country, Company*. ID: 4876815.

Shobha Bhatt, Anurag Jain, and Amita Dev. 2021. Syllable based Hindi speech recognition. *Journal of Information and Optimization Sciences* 41, no. 6 (2020): 1333-1351.

Wenfa Li1, Gongming Wang, and Ke Li. 2017. Clustering algorithm for audio signals based on the sequential Psim matrix and Tabu Search. *EURASIP Journal on Audio, Speech, and Music Processing 2017*, pp.1-9.

BloomNet: A Robust Transformer based model for Bloom's Learning Outcome Classification

Abdul Waheed[α], **Muskan Goyal**[α], **Nimisha Mittal**[α], **Deepak Gupta**[α], **Ashish Khanna**[α], **Moolchand Sharma**[α]

[α] Maharaja Agrasen Institute of Technology, New Delhi, India.

{abdulwaheed1513, goyalmuskan1508, nimishamittal1999}@gmail.com

{deepakgupta, ashishkhanna, moolchand}@mait.ac.in

Abstract

Bloom's taxonomy is a common paradigm for categorizing educational learning objectives into three learning levels: cognitive, affective, and psychomotor. For the optimization of educational programs, it is crucial to design course learning outcomes (CLOs) according to the different cognitive levels of Bloom's Taxonomy. Usually, administrators of the institutions manually complete the tedious work of mapping CLOs and examination questions to Bloom's taxonomy levels. To address this issue, we propose a transformer based model named BloomNet that captures linguistic as well semantic information to classify the course learning outcomes (CLOs) . We compare BloomNet with diverse set of basic as well as strong baselines and we observe that our model performs better than all the experimented baselines. Further, we also test the generalisation capability of BloomNet by evaluating it on different distributions which our model does not encounter during training and we observe that our model is less susceptible to distribution shift compared to the other considered models. We support our findings by performing extensive result analysis. In ablation study we observe that on explicitly encapsulating the linguistic information along with semantic information improves the model's IID (independent and identically distributed) performance as well as OOD (out-of-distribution) generalization capability. The open-sourced codebase including data can be found here: https://github.com/macabdul9/BloomNet.

1 Introduction

One of the most difficult challenges faced by the science educators is preparing a curriculum that facilitates the learning process in a structured, planned, and productive manner. It is the goal of the scientific curriculum to educate students who can investigate, question, participate in collaborative projects, and effectively communicate. The expected improvements for the students are articulated in a curriculum as learning outcomes (Zorluoğlu et al., 2019). Learning outcomes are used to track, measure, and evaluate the standards and quality of education received by the students at educational institutions (Attia, 2021). In terms of these learning outcomes, we may also identify the level of any student. Various measurement and evaluation studies are thus incorporated to determine the level of individual learning outcomes.

Exam evaluation is critical for determining how well students understand the course material. Therefore, the objectivity and scientific relevance of the questions developed for exams must be questioned in order to guarantee that students' learning outcomes are tracked and judged effectively. One of the relevant scientific techniques for analyzing this is the Bloom's Taxonomy (Anderson et al., 2000), which is well-known among the educators around the world. The examinations should take account of the difficulty levels, which correspond to the basic objectives and course outcomes in conventional ways like the Bloom's taxonomy.

Dr. Benjamin Bloom, an Educational Psychologist, developed the Bloom's Taxonomy in 1965. Its goal was to encourage high-order thinking, such as analyzing and examining instead of rote memorization of information (Adesoji, 2018). The Bloom's taxonomy is divided into three categories: cognitive (mental skills), affective (emotional areas or attitude), and psychomotor (physical skills). Our study focuses on the cognitive domain, which involves knowledge and intellectual skill development. Researchers have recently demonstrated a growing interest in automatic assessment based on cognitive domains in Bloom's Taxonomy. (Abduljabbar and Omar, 2015; Mohammed and Omar, 2018; Yahya, 2019). The majority of previous re-

search focused on question classification from a specific domain, while Bloom's taxonomy across the multi-domain region is lacking ways for classifying questions (Sangodiah et al., 2017). This work therefore seeks to establish a question classification method based on the cognitive domain of Bloom's taxonomy. The Hierarchical order of levels in cognitive domain is: Knowledge, Comprehension, Application, Analysis, Synthesis and Evaluation. The first three levels are categorised as lower level of thinking, whilst the latter three levels are considered as high level of thinking.

The primary aim of this study is to assess the utility and efficacy of Bloom's Taxonomy as a framework for establishing course learning outcomes, optimizing curriculum, and evaluating various educational programs. In this paper, we propose BloomNet, a novel transformer-based model that incorporates both linguistic and semantic information for the classification of bloom's course learning outcomes. We also examine the generalisation capability of BloomNet on new distributions because train and test distributions are usually not distributed identically. The evaluation datasets rarely represent the entire distribution and the test distribution often drifts over time (Quionero-Candela et al., 2009), resulting in train-test discrepancies. Due to these discrepancies, models can face unexpected conditions at the test time. Therefore, models should be able to detect and generalise to out-of-distribution (OOD) examples.

In most NLP evaluations, the train and test samples are assumed to be independent and identically distributed (IID). Large pretrained transformer models can achieve high performance on a variety of tasks in the IID scenario (Wang et al., 2018). However, high IID accuracy does not always imply OOD robustness. Furthermore, because pretrained Transformers rely largely on false cues and annotation artifacts (Gururangan et al., 2018; Cai et al., 2017) that OOD instances are less likely to feature, their OOD robustness is unknown. Hence, we examine the robustness of BloomNet and other experimented models such as CNNs, LSTMs, pretrained transformers, and more.

The contributions of our research can be summarized as follows:

1. We propose a transformer-based model, BloomNet, that can distinguish between six different cognitive levels of Bloom's taxonomy (Knowledge, Comprehension, Application, Analysis, Synthesis and Evaluation).

2. We implement, train and evaluate multiple models to perform comparative analysis.

3. We evaluate experimented models for OOD generalization and we observe that pretrained transformers (RoBERTa, DistilRoBERTa (Liu et al., 2019; Sanh et al., 2019)) along with proposed model have better generalization capability compared to other models.

4. We perform ablation study to asses the contribution of various components in proposed model.

The following is the final exhibition. Section 2 and Section 3 describes the previous work and methodology respectively. Section 4 delves into the experiments and results. The conclusion and possible future directions are discussed in Section 5.

2 Related Work

Text classification is an important NLP research area with numerous applications. A number of scholars have concentrated on automatic text classification. In recent years, classification of exam questions for the cognitive domain of Bloom's taxonomy has received a lot of attention. Previous works have used different features and methods for text classification. Some of these works are discussed in this section.

In (Chang and Chung, 2009), an online examination system is created that supports automatic Bloom's taxonomy analysis for the test questions. The researchers introduce fourteen keywords for the analysis on questions. Each keyword is associated with a specific cognition level. The experiment is conducted on 288 test items and a 75% accuracy is achieved for the "Knowledge" cognition level.

A. Swart and M. Daneti (Swart and Daneti, 2019) analyzed the learning outcomes for Electronic fundamental module (of two universities) using Bloom's Taxonomy. To identify the proportion of each cognition level, the verbs of each learning outcome are connected to certain specific verbs in Bloom's taxonomy. This reflected the balance between theory and practice for the cognitive development of electrical engineering students. The consistency of the findings of the two universities demonstrated that students could blend theory and

practice because they had around 40 percent of higher level cognitive outcomes.

Likewise, (Mohammed and Omar, 2020) classified exam questions for the cognitive domain of Bloom's Taxonomy using TFPOS-IDF and pretrained word2vec. To classify the questions, the extracted features are fed to three distinct classifiers i.e. logistic regression, K-nearest neighbour, and Support Vector Machine. For the experiment, they employ two datasets, one with 141 questions and the other with 600 questions. The first dataset results in a weighted average of 71.1%, 82.3% and 83.7% while the second achieves a weighted average of 85.4%, 89.4% and 89.7%.

Adidah Lajis et al. proposed (Lajis et al., 2018) a framework for assessing students' programming skills. Bloom's taxonomy cognitive domain serves as the foundation for the framework. According to the findings, Bloom's taxonomy could be used as a basis for grading students. It said that the students would be judged based on their ability using Bloom's taxonomy. The authors also suggested that taxonomy be used as an evaluation framework rather than learning.

Based on their domain knowledge, teachers and accreditation organizations manually classify course learning outcomes (CLOs) and questions on distinct levels of cognitive domain. This is time-consuming and usually leads in errors due to human bias. As a result, this technique must be automated. Several scholars have sought to automate this process through the use of natural language processing and machine learning techniques (Haris and Omar, 2012; Jayakodi et al., 2015; Osadi et al., 2017; Kowsari et al., 2019). Deep learning has recently exhibited impressive results when compared to traditional machine learning methods, particularly in the field of text classification (Minaee et al., 2020).

For text classification tasks, several neural models that automatically represent text as embedding have been developed, such as CNNs, RNNs, graph neural networks, and a variety of attention models such as hierarchical attention networks, self-attention networks, and so on. The majority of previous efforts on Bloom's taxonomy have either used traditional machine learning approaches or representative deep neural models such as RNNs, LSTMs, and so on. In this research, we propose a transformer-based approach for performing text classification as per cognitive domains. Transformers, (Vaswani et al., 2017) provide significantly better parallelization than RNNs, allowing for efficient (pre-)training of very large language models and an enhanced performance rate.

3 Methodology

In this section we discuss the methodology part of our research. Our model is inspired by (Gupta et al., 2021) and (Yang et al., 2016b). In BloomNet , we encapsulate the linguistic information along with generic input representation and we also explicitly model word level attention. In following sections we describe each component of our model (shown in Figure 1) in detail.

Notation: We denote current input as set of tokens $x \in X = \{t_0, t_1, t_2, ...t_n\}$ where n is the number of tokens in input. We define a model as $f_{\texttt{model}} : x \longrightarrow h$ where $h \in \mathbb{R}^d$. We define our final classfieir as $f_{\texttt{c}} : x \longrightarrow C$ where C is the softmax output and its size is equal to number of classes in our data.

3.1 Representation Model

Representation model or language encoder is main component of BloomNet which gives contextualized embeddings (Devlin et al., 2019; Pennington et al., 2014) for the text input. and for this we use pretrained RoBERTa (Liu et al., 2019) model from huggingface model hub repository (Wolf et al., 2020). The reason we use RoBERTa instead of its other widely used counterparts such as BERT (Devlin et al., 2019), DistilBERT (Sanh et al., 2019) is that it has seen much more data during its pretraining compare to its predecessor which results in increased robustness for subpopulation as well distribution shift. We feed tokenized input to RoBERTa model and we use `CLS` token as input representation. It can be represented as :

$$h_{\text{rep}} = f_{\text{rep}}(x) \tag{1}$$

3.2 Linguistic Encapsulation

Work by (Gupta et al., 2021) shows that explicit encapsulation of linguistic information increases the performance of the model for claim detection task, inspired by the same we also explicitly encapsulate linguistic information in modelling of BloomNet. We use POS (Part-Of-Speech) and NER (Named Entity Recognition) information coming from a trained model for POS and NER tasks respectively. We freeze the POS and NER model during training so that it's weights do not change and hence it

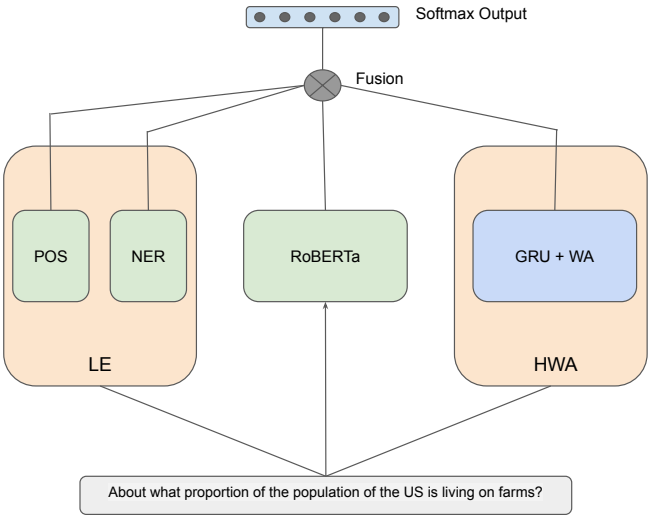


Figure 1: A high level architectrue digram of the proposed model BloomNet . POS (Part of Speech Module. NER(Named Entity Recognition) Module. HWA (Hierarchical Word Attention) Module. LE (Linguistic Encapsulation).

carries linguistic information. We use `CLS` token representation of the model and we define it as as follows:

$$h_{\text{POS}} = f_{\text{POS}}(x) \tag{2}$$

$$h_{\text{NER}} = f_{\text{NER}}(x) \tag{3}$$

3.3 Hierarchical Word Attention and Classification

Inspired by (Yang et al., 2016b) we use word level attention to get the better dense representation of the input. For this we use GRU Cho et al. (2014) and apply word level attention on its output. As result we get a vector from this module as input representation and we use this along with other information for classification. We denote this as follows :

$$h_{\text{HWA}} = f_{\text{HWA}}(x) \tag{4}$$

Finally, we get four different representation coming from different components and we fuse these information using concatenation and feed this to a linear classification model. We write the concatenation as:

$$H = h_{\text{Rep}} \oplus h_{\text{POS}} \oplus h_{\text{NER}} \oplus h_{\text{HWA}} \tag{5}$$

Classification module can be represented as:

$$C = f_{\text{c}}(H) \tag{6}$$

4 Experiments and Results

4.1 Dataset

We use two open domain datasets to evaluate the proposed approach. First dataset was proposed in (Yahya et al., 2012) which comprises 600 open-ended questions. The second dataset was compiled from a variety of websites, publications, and previous research (Haris and Omar, 2015). It contains 141 open-ended questions. The datasets are annotated and classified into six categories (Knowledge, Comprehension, Application, Analysis, Synthesis, and Evaluation). Table 1 illustrates the label distribution for both datasets. The questions in these two datasets come from a variety of fields of study, including chemical, literature, biological, artistic, and computer science, among others.

4.2 Baselines

In this section, we describe the various baseline models that we used for comparative analysis. These models are arranged in the order of their performance.

4.2.1 VDCNN

Very Deep CNN (VDCNN) (Schwenk et al., 2017) learns a hierarchical representation of a sentence with the help of a deep stack of convolutions and max-pooling of size 3 and by operating at the char-

Cognitive Level	Dataset 1	Dataset 2
Knowledge Level	100	26
Comprehension Level	100	23
Application Level	100	15
Analysis Level	100	23
Synthesis Level	100	30
Evaluation Level	100	24
Total	**600**	**141**

Table 1: Number of questions in each cognitive level

acter level representation of the text. VDCNNs are substantially deeper than convolutional neural networks published previously. This is the first CNN model to present the "advantage of depth" in the field of NLP.

4.2.2 LSTM

Text is viewed as a sequence of words in RNN-based models, which are designed to capture word dependencies and text structures for text classification. RNNs (Jain and Medsker, 1999) can memorise the local structure of a word sequence, but they struggle with long-range dependencies. Long-Short Term Memory (LSTM) (Sari et al., 2020) is the most popular variant of RNN, that is created to capture long term dependencies. Vanilla RNNs suffer from gradient vanishing problem and LSTMs resolve this issue by using a memory cell that remember values across arbitrary time periods.

4.2.3 HAN

Hierarchical Attention Networks (HAN) (Yang et al., 2016a) collects relevant tokens from sentences and aggregate their representation with the help of an attention mechanism. The same approach is used to retrieve relevant sentence vectors that is used in the classification task.

4.2.4 CNN

RNNs are taught to detect patterns over time, while CNNs (Kim, 2014) are taught to recognise patterns over space. RNNs work for NLP tasks like POS tagging or QA that need understanding of long-range semantics, but CNNs are good for recognising local and position-invariant patterns (LeCun et al., 1998). These patterns could be key phrases expressing a specific emotion or a topic. As a result, CNNs have become one of the most common text classification model.

4.2.5 RCNN

In contrast to CNN, Recurrent CNN (Girshick et al., 2014) comprises of bi-directional recurrent structure that captures greater contextual data from word representations. This is followed by a max pooling layer which is responsible for extracting key features for text classification.

4.2.6 Seq2Seq-Attention

Deep learning models known as sequence-to-sequence (Bahdanau et al., 2015) models have been deployed in tasks such as machine translation, text summarization, and image captioning. Seq2Seq comprises of encoder, decoder and attention layer where encoder is responsible for compiling data in the form of vector. Further this context is parsed to the decoder that produces desired output sequence. The primary idea behind the attention mechanism is to avoid learning a single vector representation for each sentence and instead be attentive to specific input vectors based on the attention weights.

4.2.7 Self-Attention

Self-attention is a type of attention that allows us to learn the relationship between words in a sentence. Various NLP tasks and Transformers (Vaswani et al., 2017) use self-attention. Despite the fact that CNNs are less sequential than RNNs, the computing cost of capturing relationships between words in a phrase increases with the length of the sentence, much like RNNs. Transformers get around this constraint by using self-attention to compute a "attention score" for each word in a sentence or document in parallel, modelling the influence each word has on the others.

4.2.8 TF-IDF Random Forest

Random Forest (RF) models (Xue and Li, 2015) are made up of a collection of decision trees that were trained on random feature subsets. This model's predictions are obtained via a majority vote of all forest tree projections. In addition, RF classifiers are simple to apply to text classification of high-dimensional noisy data. Furthermore, TF-IDF (Term Frequency Inverse Document Frequency) (Sammut and Webb, 2010) is a commonly used approach for converting text to a number representation that may be employed by a machine algorithm.

TfidfVectorizer weights word counts based on how frequently they appear in the sentence.

4.2.9 DistilRoBERTa

DistilRoBERTa has been distilled from RoBERTa-base model (Liu et al., 2019; Sanh et al., 2019) that contains around half number of parameters as BERT model. It is based on same training process as that of DistilBERT. Moreover, DistilRoBERTa maintains 95 percent of BERT's performance on the GLUE language understanding benchmark (Wang et al., 2018).

4.2.10 RoBERTa

RoBERTa (Robustly Optimized BERT Pretraining Accuracy) model (Liu et al., 2019) is a more robust version of BERT that is trained with a lot more data. It is based on fine-tuning the hyper-parameters that has improved the results and performance of the model significantly. To boost performance of BERT, RoBERTa also modified its training procedure and architecture. These modifications include removing next sentence prediction and dynamically changing the masking pattern during pre-training.

4.3 Experimental Setup

We use HuggingFace Transformers (Wolf et al., 2020), PyTorch (Paszke et al., 2019), PyTorch-Lightning (Falcon, 2019) and Scikit-learn (Pedregosa et al., 2011) for model implementation, training and evaluation. We train all of the models with KFold (k=5) cross validation and report `mean` and `std` values across the folds. We observe that the text in the dataset is relatively short, thus we use `maximum sequence lengths = 128`. For LSTM-like models, we employ `hidden size = 768`, `number of layers = 4`, and `dropout = 0.10` throughout the experiments. We use `Adam` (Kingma and Ba, 2015) as the optimizer and `cross-entropy` as the objective function. We use different `learning rates` for different models depending on how they are initialized, and for `BloomNet` we use `learning rate = 2e-5` and train all models for `50 epochs` with `batch size = 32` and `early stopping` to prevent overfitting. We do not change the shared hyper-parameters across the models so that the comparison is as fair as possible. We do not conduct comprehensive hyper-parameter searches due to computational constraints.

4.4 Results

We evaluate following baselines to compare the performance of our proposed model, BloomNet: VDCNN (Schwenk et al., 2017), LSTM (Sari et al., 2020), HAN(Yang et al., 2016a), CNN(Kim, 2014), RCNN (Girshick et al., 2014), Seq2Seq-Attention (Bahdanau et al., 2015), Self-Attention (Vaswani et al., 2017), Random Forest (Xue and Li, 2015), DistilRoBERTa (Liu et al., 2019), and RoBERTa (Liu et al., 2019). We find that BloomNet outperforms all the considered baselines on the two datasets, demonstrating its higher performance for text classification. Table 2 reports the performance of BloomNet and all the baselines.

The model is trained on Dataset1 and evaluated for both the datasets. Dataset1 is used for evaluating BloomNet's IID performance while Dataset2 is used to test BloomNet's generalisation capabilities (OOD performance) by assessing it on new distributions that our model does not encounter during the training process. We observe that in comparison to the baseline models, BloomNet is less vulnerable to distribution shift.

4.4.1 Comparative Analysis

As seen in Table 2 BloomNet outperforms the baseline models and achieves 87.50 ± 1.88 and 70.40 ± 2.52 and Macro-F1 score 87.23 ± 2.47 and 67.10 ± 2.43 on Dataset1(IID) and Dataset2(OOD) respectively.

In addition, we also made some very in-depth observations while evaluating the baselines. Surprisingly, the TF-IDF (Sammut and Webb, 2010) encoded text with random forest performs better than several strong baselines like LSTM, HAN, CNN, and RCNN. It is the third best performing baseline model that achieves 70.66 ± 2.52 and 62.12 ± 1.38 accuracy and Macro-F1 70.50 ± 2.75 and 58.04 ± 1.73 on Dataset1(IID) and Dataset2(OOD) respectively.

We also observe that Attention based models like Seq2Seq-Attention and Self-Attention show better classification performance than vanilla mod-

[1]Very Deep Convolutional Networks for Text Classification(VDCNN)
[2]Long Short-Term Memory (LSTM)
[3]Hierarchical Attention Networks (HAN)
[4]Convolutional Neural Network (CNN)
[5]Recurrent Convolutional Neural Network (RCNN)
[6]Sequential to Sequential Model with Attention
[7]Term Frequency - Inverse Document Frequency(TF-IDF)
[8]Distilled from RoBERTa model
[9]Robustly Optimized BERT Pre-training Approach

Model OV	Dataset1 (IID)		Dataset2(OOD)	
	Accuracy	Macro-F1	Accuracy	Macro-F1
VDCNN[1]	32.00 ± 6.78	31.70 ± 6.71	28.79 ± 3.82	26.54 ± 4.12
LSTM[2]	58.50 ± 3.99	59.27 ± 3.55	47.09 ± 4.05	45.47 ± 2.71
HAN[3]	59.64 ± 3.72	58.90 ± 4.16	54.69 ± 3.39	50.61 ± 3.12
CNN[4]	60.67 ± 1.11	60.57 ± 1.36	49.79 ± 2.17	48.17 ± 2.00
RCNN[5]	66.33 ± 3.01	65.90 ± 3.51	54.04 ± 3.57	51.05 ± 3.09
Seq2Seq-Attention[6]	64.00 ± 3.09	63.79 ± 3.50	52.91 ± 2.22	50.92 ± 2.11
Self-Attention	70.17 ± 3.55	69.92 ± 3.80	55.46 ± 2.07	52.75 ± 1.81
Random Forest TF-IDF[7]	70.66 ± 2.52	70.50 ± 2.75	62.12 ± 1.38	58.04 ± 1.73
DistilRoBERTa[8]	80.50 ± 3.23	80.21 ± 3.49	67.80 ± 1.59	63.94 ± 1.48
RoBERTa[9]	82.00 ± 2.01	81.67 ± 2.20	68.65 ± 2.74	65.65 ± 2.82
BloomNet	$\mathbf{87.50 \pm 1.88}$	$\mathbf{87.23 \pm 2.47}$	$\mathbf{70.40 \pm 2.52}$	$\mathbf{67.10 \pm 2.43}$

Table 2: Mean and Standard deviation of the results obtained over 5 folds. BloomNet performs significantly better ($p < 0.004$) than the RoBERTa. **Bold** shows best performance. All models are trained and evaluated on Dataset1 hence IID, and OOD evaluation is performed on Dataset2.

els (like VDCNN, CNN, LSTM, and RCNN). Further, we investigate BERT-based models DistilRoBERTa and RoBERTa (which are pre-trained Transformers) that achieve superior performanc over all the other considered baselines. RoBERTa is the best performing model with accuracy of 82.00 ± 2.01 and 68.65 ± 2.74 and Macro-F1 score 81.67 ± 2.20 and 65.65 ± 2.82 on Dataset1(IID) and Dataset2(OOD) respectively.

4.4.2 Out-of-distribution Generalisation

We evaluate models on new data which is not seen during training to evaluate the OOD robustness. We observe that OOD and IID performance is linearly correlated. The models that do not perform well on IID data such as VDCNN, LSTM, etc also perform poor on OOD data. Pretrained transformers have been proven robust to distribution shift (Hendrycks et al., 2020; Ramesh Kashyap et al., 2021) but in our case we notice significant performance drop (20%) between IID and OOD data across all the pretrained transformer based models in our experiment which is same for other models as well. We hypothesise that this might be caused by large discrepancy between IID and OOD data.

4.5 Ablation Study

Our proposed model BloomNet has three main component: 1. Representation model or Language Encoder 2. Linguistic Encapsulation Module and 3. Hierarchical Word Attention Model. We conduct an ablation study to assess the contribution of different components in our model. First we remove the word attention module from BloomNet and train it like other models with same configuration. We observe the BloomNet without word attention yields ≈ 84 and ≈ 65 accuracy for IID and OOD data respectively. Then we remove the linguistic encapsulation block and train the model like previously. BloomNet without linguistic encapsulation yeilds similar IID performance (≈ 84 accuracy) but performs better on OOD data. If we remove the both components word attention as well as linguistic encapsulation BloomNet is same as RoBERTa (Liu et al., 2019) baseline. The result of ablation is stated in the table 3.

Component	Accuracy	Macro-F1
RoBERTa	82.00	81.67
+ WA	84.11	84.10
+ POS-NER	84.64	84.48
+ WA + POS-NER	**87.50**	**87.23**

Table 3: Ablation results for BloomNet . Mean of IID Accuracy and Macro-F1 is reported. Linguistic Encapsulation along with Word level attention yields significantly better ($p < 0.004$) results. WA: Word Attention. POS: Part-of-speech. NER: Named-Entity Recognition.

5 Discussion

Limitations: We propose a novel transformer based model named BloomNet which has three language encoder (we use RoBERTa), two for linguistic encoding named as POS Encoder and

NER Encoder, and one generic encoder. Due to three large transformer based language encoder proposed model is compute and memory heavy hence it becomes very cumbersome to deploy it in production. To asses the generalization capability of models we evaluate them on a different distribution which they do not see during training. We do not quantify the shift between IID and OOD and we restrict ourselves to only evaluation as investing the cause of performance drop on OOD data is beyond scope of this study. The datasets used in our work is relatively small having 600 and 141 samples respectively in both Dataset1 and Dataset2. Although we do cross validation and report mean and standard-deviation but we expect change in performance on bigger dataset. For same reason we do not train models on Dataset2.

Ethical Considerations: We are well aware of the societal implication of deploying large language models it could have unintended bias against marginalized groups and model itself plays significant role in amplifying those biases. We do not see any immediate misuse of our work, but more research in this area could lead to the development of systems such as automated scoring, which can have a disproportionately detrimental impact on marginalized groups.

6 Conclusion and Future Work

We propose a novel transformer-based model, BloomNet, that captures the linguistic and semantic information to classify the course learning outcomes according to the different cognitive domains of Bloom's Taxonomy. BloomNet outperforms the considered baseline models analyzed in this study in terms of performance and generalization capability. Interestingly, we observe that carefully processed text with TF-IDF encoding outperforms numerous strong baselines like CNN, RNN, and attention based models. We also observe that pretrained Transformers generalize to OOD examples surprisingly well. Overall, we use a state-of-the-art Natural Language Processing (NLP) model for a relatively new task, and we believe it opens up new research directions for NLP in the education domain. We believe that, similar to previous domain-oriented NLP studies, such as NLP4Health, NLP4Programming, LegalNLP, and so on, NLP4Education has the potential to improve existing systems for the mutual benefit of the community and society in general. This is a novel task employing the state-of-the-art Natural Language Processing(NLP) system into education which is relatively new and we believe that it will open a new direction for NLP research.

References

D. Abduljabbar and N. Omar. 2015. Exam questions classification based on bloom's taxonomy cognitive level using classifiers combination. Journal of theoretical and applied information technology, 78:447–455.

F. Adesoji. 2018. Bloom taxonomy of educational objectives and the modification of cognitive levels. Advances in Social Sciences Research Journal, 5.

L. Anderson, D. Krathwohl, and B. Bloom. 2000. A taxonomy for learning, teaching, and assessing: A revision of bloom's taxonomy of educational objectives.

A. S. Attia. 2021. Bloom's taxonomy as a tool to optimize course learning outcomes and assessments in architecture programs.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. CoRR, abs/1409.0473.

Zheng Cai, Lifu Tu, and Kevin Gimpel. 2017. Pay attention to the ending: Strong neural baselines for the roc story cloze task. In ACL.

Wen-Chih Chang and Ming-Shun Chung. 2009. Automatic applying bloom's taxonomy to classify and analysis the cognition level of english question items. 2009 Joint Conferences on Pervasive Computing (JCPC), pages 727–734.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1724–1734, Doha, Qatar. 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.

et al. Falcon, WA. 2019. Pytorch lightning. GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning, 3.

Ross B. Girshick, Jeff Donahue, Trevor Darrell, and J. Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. 2014 IEEE Conference on Computer Vision and Pattern Recognition, pages 580–587.

Shreya Gupta, Parantak Singh, Megha Sundriyal, Md. Shad Akhtar, and Tanmoy Chakraborty. 2021. LESA: Linguistic encapsulation and semantic amalgamation based generalised claim detection from online content. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pages 3178–3188, Online. Association for Computational Linguistics.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In NAACL-HLT.

S. S. Haris and N. Omar. 2012. A rule-based approach in bloom's taxonomy question classification through natural language processing. 2012 7th International Conference on Computing and Convergence Technology (ICCCT), pages 410–414.

S. S. Haris and N. Omar. 2015. Bloom's taxonomy question categorization using rules and n-gram approach. Journal of theoretical and applied information technology, 76:401–407.

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. 2020. Pretrained transformers improve out-of-distribution robustness. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 2744–2751, Online. Association for Computational Linguistics.

L. C. Jain and L. R. Medsker. 1999. Recurrent Neural Networks: Design and Applications, 1st edition. CRC Press, Inc., USA.

K. Jayakodi, M. Bandara, and I. Perera. 2015. An automatic classifier for exam questions in engineering: A process for bloom's taxonomy. 2015 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE), pages 195–202.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In EMNLP.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. CoRR, abs/1412.6980.

Kamran Kowsari, K. Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura E. Barnes, and D. Brown. 2019. Text classification algorithms: A survey. Inf., 10:150.

Adidah Lajis, H. Nasir, and N. A. Aziz. 2018. Proposed assessment framework based on bloom taxonomy cognitive competency: Introduction to programming. Proceedings of the 2018 7th International Conference on Software and Computer Applications.

Y. LeCun, L. Bottou, Yoshua Bengio, and P. Haffner. 1998. Gradient-based learning applied to document recognition.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. ArXiv, abs/1907.11692.

Shervin Minaee, Nal Kalchbrenner, E. Cambria, Narjes Nikzad, M. Chenaghlu, and Jianfeng Gao. 2020. Deep learning–based text classification. ACM Computing Surveys (CSUR), 54:1 – 40.

Manal Mohammed and N. Omar. 2018. Question classification based on bloom's taxonomy using enhanced tf-idf. International Journal on Advanced Science, Engineering and Information Technology, 8:1679–1685.

Manal Mohammed and N. Omar. 2020. Question classification based on bloom's taxonomy cognitive domain using modified tf-idf and word2vec. PLoS ONE, 15.

K. A. Osadi, Mgnas Fernando, and W. V. Welgama. 2017. Ensemble classifier based approach for classification of examination questions into bloom's taxonomy cognitive levels. International Journal of Computer Applications, 162:1–6.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary 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. In Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in python. Journal of Machine Learning Research, 12(85):2825–2830.

Jeffrey Pennington, R. Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In EMNLP.

Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil Lawrence. 2009. Dataset shift in machine learning.

Abhinav Ramesh Kashyap, Laiba Mehnaz, Bhavitvya Malik, Abdul Waheed, Devamanyu Hazarika, Min-Yen Kan, and Rajiv Ratn Shah. 2021. Analyzing the domain robustness of pretrained language models, layer by layer. In Proceedings of the Second Workshop on Domain Adaptation for NLP, pages 222–244, Kyiv, Ukraine. Association for Computational Linguistics.

Claude Sammut and Geoffrey I. Webb, editors. 2010. TF–IDF, pages 986–987. Springer US, Boston, MA.

A. Sangodiah, Rohiza Ahmad, and W. Ahmad. 2017. Taxonomy based features in question classification using support vector machine 1.

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.

Winda Kurnia Sari, Dian Palupi Rini, and R. F. Malik. 2020. Text classification using long short-term memory with glove features.

Holger Schwenk, Loïc Barrault, Alexis Conneau, and Y. LeCun. 2017. Very deep convolutional networks for text classification. In EACL.

A. Swart and M. Daneti. 2019. Analyzing learning outcomes for electronic fundamentals using bloom's taxonomy. 2019 IEEE Global Engineering Education Conference (EDUCON), pages 39–44.

Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. ArXiv, abs/1706.03762.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In BlackboxNLP@EMNLP.

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.

Dashen Xue and Fengxin Li. 2015. Research of text categorization model based on random forests. 2015 IEEE International Conference on Computational Intelligence & Communication Technology, pages 173–176.

Anwar Ali Yahya. 2019. Swarm intelligence-based approach for educational data classification. J. King Saud Univ. Comput. Inf. Sci., 31:35–51.

Anwar Ali Yahya, Z. Toukal, and A. Osman. 2012. Bloom's taxonomy-based classification for item bank questions using support vector machines. In Modern Advances in Intelligent Systems and Tools.

Zichao Yang, Diyi Yang, Chris Dyer, X. He, Alex Smola, and E. Hovy. 2016a. Hierarchical attention networks for document classification. In HLT-NAACL.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016b. 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, San Diego, California. Association for Computational Linguistics.

S. L. Zorluoğlu, Kübra Elif Bağrıyanık, and Ayşe Şahintürk. 2019. Analyze of the science and technology course teog questions based on the revised bloom taxonomy and their relation between the learning outcomes of the curriculum. The International Journal of Progressive Education, 15:104–117.

Indic Languages Automatic Speech Recognition using Meta-Learning Approach

Anugunj Naman
IIIT Guwahati, India
anugunj.naman@iiitg.ac.in

Kumari Deepshikha
LOWE's, Bengaluru
kumari.deepshikha@lowes.com

Abstract

Recently Conformer-based models have shown promising leads to Automatic Speech Recognition (ASR), outperforming transformer-based networks while meta-learning has been extremely useful in modeling deep learning networks with a scarcity of abundant data. In this work, we use Conformers to model both global and local dependencies of an audio sequence in a very parameter-efficient way and meta-learn the initialization parameters from several languages during training to attain fast adaptation on the unseen target languages, using model-agnostic meta-learning algorithm (MAML). We analyse and evaluate the proposed approach for seven different Indic languages. Preliminary results showed that the proposed method, MAML-ASR, comes significantly closer to state-of-the-art monolingual Automatic Speech Recognition for all seven different Indic languages in terms of character error rate.

1 Introduction

"Ok, Google. Hi Alexa. Hey Siri." have featured an enormous boom of smart speakers in recent years, unveiling a trend towards ubiquitous and ambient computing (AI) for better daily lives. As the communication bridge between humans and machines, multilingual ASR is of central importance. India is a country with an enormous amount of languages and catering to those languages is difficult without having a large amount of label training corpora. Pretraining on other language sources as the initialization, then fine-tuning on target language is the main approach for such low-resource setting, also referred to as multilingual transfer learning pretraining (Multi-ASR) (Vu et al., 2014) (Tong et al., 2017). Multi-ASR models are designed to learn using an encoder to extract language-independent representations to build a better acoustic model from many source languages. The success of language independent features to improve ASR performance compared to monolingual training has been shown in many recent works (Dalmia et al., 2018) (Cho et al., 2018)(Tong et al., 2018). However, there performance have been lacklustre compared to model trained directly using target language, i.e., training for single language only.

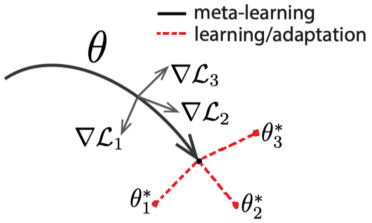


Figure 1: The MAML algorithm learns a good parameter initializer θ by training across various meta-tasks such that it can adapt quickly to new tasks.

In this paper, we follow on the concept of multilingual pretraining – Meta-learning. Meta-learning, or learning-to-learn, has recently received considerable interest within the machine learning community. The goal of meta-learning is to resolve the matter of fast adaptation on unseen data, which is aligned with our low-resource setting for different Indic languages. We use model-agnostic meta-learning algorithm (MAML) (Finn et al., 2017) in this work. As its name suggests and seen in figure 1, MAML can be applied to any neural network architecture since it only modifies the optimization process following a meta-learning training method. It doesn't introduce any additional modules like adversarial training or requires phoneme level annotation like hierarchical approaches (Hsu et al., 2019).

In recent times, the Transformer architecture based on self-attention (Zhang et al.,

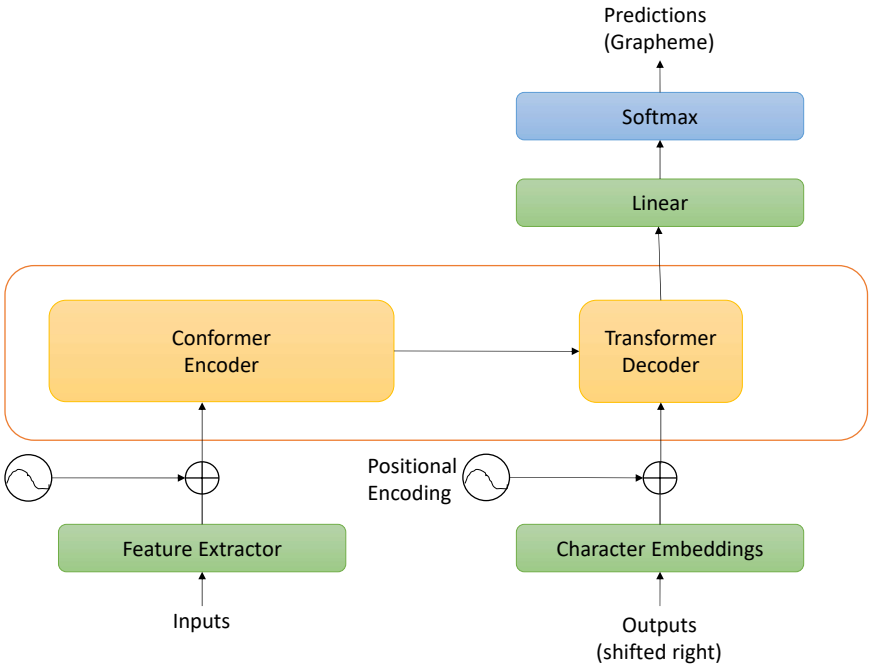


Figure 2: Transformer ASR model architecture.

2020)(Vaswani et al., 2017) has shown widespread adoption for modeling sequences due to its ability to capture long-distance interactions and the high training efficiency. Alternatively, convolutions have also been successful for speech recognition (Li et al., 2019) (Kriman et al., 2019)(Han et al., 2020)(Sainath et al., 2013)(Abdel-Hamid et al., 2014), that capture local context progressively using local receptive field layer by layer.

However, models with convolutions or self-attention each have their own limitations. While Transformers are good at modeling long-range global context, they are not very capable to extract fine-grained local feature patterns. Convolution networks, on the other hand, exploit local information and are used as the common computational block in vision. They learn shared position-based kernels over a local window which maintains translation equivariance and can capture features like edges and shapes. One limitation of using local connectivity is that you need several layers or parameters to capture global information. To tackle this issue, contemporary work ContextNet (Han et al., 2020) adopts the squeeze-and-excitation module (Hu et al., 2018) in each residual block to capture longer context. However, the model is still limited in capturing dynamic global context because it only applies a global averaging over the entire sequence.

Recently, combining convolution and self-attention has shown significant improvement in automatic speech recognition model as they can learn both position-wise local features and use content-based global interactions. We have used Conformers (Gulati et al., 2020) in this work. Conformers are the combination of self-attention and convolution sandwiched between a pair of feed-forward modules that achieves the best of both worlds i.e., self-attention learns the global interaction whilst the convolutions coherently captures the relative offset-based local correlations.

We evaluated the effectiveness of the proposed model of several Indic languages. Our experiments show that our model comes close to monolingual models.

2 Proposed Method

In this section, we present the architecture of our conformer-based speech recognition model and the proposed meta-learning method for fast adaptation to the multilingual speech recognition task.

2.1 Conformer Speech Recognition Model

As shown in Figure 2, we build our model using a Conformers to learn to predict graphemes from the

speech input. Our model extracts learnable features from audio inputs using a feature extractor module to generate input embeddings. The encoder process the input embeddings generated from the feature extractor module using conformer blocks. Mathematically, this means, for input x_i to a Conformer block i, the output z_i of the block is:

$$\begin{aligned}\tilde{x_i} &= x_i + \frac{1}{2}\text{FFN}(x_i)\\ x'_i &= \tilde{x_i} + \text{MHSA}(\tilde{x_i})\\ x''_i &= x'_i + \text{Conv}(x'_i)\\ z_i &= \text{Layernorm}(x''_i + \frac{1}{2}\text{FFN}(x''_i))\end{aligned} \tag{1}$$

where FFN refers to the Feedforward module, MHSA refers to the Multi-Head Self-Attention module, and Conv refers to the Convolution module as described in the preceding sections (Gulati et al., 2020).

Then the decoder receives the encoder outputs from conformer blocks and applies multi-head attention to its input to finally compute the logits of the outputs. To generate the probability of the outputs, we then compute the value of logits using a softmax function. We also apply a mask in the attention layer to avoid any possible information flow from future tokens. We then train our model by optimizing the next-step prediction on the previous characters and by maximizing the log probability shown below:

$$\max_{\theta} \sum_{i} \log P(y_i|z, y'_{<i}; \theta), \tag{2}$$

where z is the character inputs, y_i is the next predicted character, and $y'_{<i}$ is the ground truth of the previous characters. uring inference, we generate the output sequence using a beam-search method in an auto-regressive manner. Then we maximize the following objective function:

$$\eta \sum_{i} \log P(y_i|z, \hat{y}_{<i}; \theta) + \gamma \sqrt{wc(\hat{y}_{<i})}, \tag{3}$$

where η is the parameter to control the decoding probability from the decoder, and γ is the parameter to control the effect of the word count $wc(\hat{y}_{<i})$ as suggested in (Winata et al., 2019) and (Winata et al., 2020).

2.2 Fast Adaptation via Meta-Learning

Model-agnostic meta-learning (MAML) (Finn et al., 2017) learns to quickly adapt to a new task from a number of different tasks using a gradient descent method. In this paper, we apply MAML to effectively learn from a set of languages and quickly adapt to a new language in the few-shot setting. We denote our Conformer based ASR as f_θ parameterized by θ. Our dataset is consist a set of languages $\mathcal{A} = \{A_1, A_2, \cdots, A_n\}$, and for each language i, we split the data into A_i^{tra} and A_i^{val}, then update θ into θ' by computing gradient descent updates on A_i^{tra}:

$$\theta'_i = \theta - \alpha \nabla_\theta \mathcal{L}_{A_i^{tra}}(f_\theta), \tag{4}$$

where α is the fast adaptation learning rate. During the training, the model parameters are trained to optimize the performance of the adapted model $f(\theta'_i)$ on unseen A_i^{val}. The meta-objective is defined as follows:

$$\min_{\theta} \sum_{A_i \sim p(\mathcal{A})} \mathcal{L}_{A_i^{val}}(f_{\theta'_i}) = \sum_{A_i \sim p(\mathcal{A})} \mathcal{L}_{A_i^{val}}(f_{\theta - \alpha \nabla_\theta \mathcal{L}_{A_i^{tra}}(f_\theta)}). \tag{5}$$

where $\mathcal{L}_{A_i^{val}}(f_{\theta'_i})$ is the loss evaluated on A_i^{val}. We collect the loss $\mathcal{L}_{A_i^{val}}(f_{\theta'_i})$ from a batch of languages and perform the meta-optimization as follows:

$$\theta \leftarrow \theta - \beta \sum_{A_i \sim p(\mathcal{A})} \nabla_\theta \mathcal{L}_{A_i^{val}}(f_{\theta'_i}), \tag{6}$$

where β is the meta step size and $f_{\theta'_i}$ is the adapted network on language A_i. The meta-gradient update step is performed to achieve a good initialization for our conformer based ASR model, then we can optimize our model with few number of samples on target languages in the fine-tuning step. In this

Table 1: Statistics of Indic Language Speech Data.

Language	# Samples
Assamese **(as)**	36,000
Bengali **(be)**	232,537
Hindi **(hi)**	80,000
Marathi **(ma)**	44,500
Nepali **(ne)**	157,905
Sinhala **(sh)**	185,293
Tamil **(ta)**	62,000
Total	798,235

Table 2: Average Character Error Rate (% CER) comparison with single training.

Languages	**MAML**						**Single Training**
-	10%-shot	25%-shot	50%-shot	75%-shot	all-shot		-
Assamese	61.29	50.87	41.80	25.44	13.44	**(+1.86)**	11.58
Bengali	57.48	47.60	38.19	26.47	10.77	**(+2.04)**	8.73
Hindi	55.49	43.81	35.78	23.43	10.19	**(+2.92)**	7.27
Marathi	56.78	45.30	36.68	23.56	10.04	**(+2.91)**	7.13
Nepali	57.33	47.33	35.27	22.85	10.32	**(+3.46)**	6.86
Sinhala	54.36	45.22	35.15	24.36	11.69	**(+4.36)**	7.33
Tamil	60.38	48.70	39.89	27.41	19.74	**(+4.21)**	15.53

Table 3: Mean Human Evaluation score(0-5) for Indic Languages

Language	**MAML**		**Single Training**	
-	**Mean Correct**	**Mean Fluency**	**Mean Correct**	**Mean Fluency**
Assamese **(as)**	4.1	4.0	4.4	4.5
Bengali **(be)**	4.0	4.0	4.4	4.4
Hindi **(hi)**	4.2	4.1	4.4	4.5
Marathi **(ma)**	4.0	4.1	4.5	4.5
Nepali **(ne)**	4.1	4.0	4.6	4.5
Sinhala **(sh)**	4.1	4.1	4.5	4.4
Tamil **(ta)**	3.9	4.1	4.2	4.2

work, we use first order approximation MAML (Gu et al., 2018) and (Finn et al., 2018), thus Equation 6 is further rewritten as:

$$\theta \leftarrow \theta - \beta \sum_{A_i \sim p(\mathcal{A})} \nabla_{\theta_i'} \mathcal{L}_{A_i^{val}}(f_{\theta_i'}). \quad (7)$$

3 Experiments

3.1 Dataset

We use Assamese, Tamil, and Marathi datasets from Government of India DeitY-TDIL and Bengali, Sinhala and Nepali datasets (Kjartansson et al., 2018) from Open-SLR. The statistics of the dataset are shown in Table-1. The dataset is imbalanced with languages with a large number of training samples.

3.2 Experimental Details

We preprocess the raw audio inputs into a spectrogram before we fetch it into our conformer based model. Our model utilizes a VGG model (Simonyan and Zisserman, 2015), a 6-layer CNN architecture, as the feature extractor. Our speech recognition model consists of sixteen conformer encoders and three transformer decoder layers with eight heads for multi-head attention. The conformer consists of a $dim_{encoder}$ of 512. In total, our model has around 14.9M parameters. For both the MAML and single training models (training model on target language directly), we end the training process after 3M iterations and 1M iteration respectively. During the fine-tuning step for MAML, we run 15 iterations for each sample. We evaluate our model using a beam search with $\eta = 1$, $\gamma = 0.1$, and a beam size of 5. In the single-training setting as well as MAML based training setting, we downsample the speech data to a 16 kHz audio sample rate. The code can be found at here

We train and evaluate the effectiveness of MAML for Indic languages by comparing its performance with the stand-alone conformer model trained on a single language i.e., single-training setting. For each language in MAML taken as target language during experiment, every other languages are used in training. During testing we fine-tune the MAML with target language and then We evaluate the model performance using the character error rate (CER) and run experiments ten times using different test folds. We report the average and standard error of all folds in the 10%-shot, 25%-shot, 50%-shot, 75%-shot and all-shot settings, where q-shot setting means only q% data is used in training from training set.

LANGUAGE: HINDI

ORIGINAL: मुझे इससे कोई फर्क नहीं पड़ता कि रन कहां बने हैं क्योंकि टेस्ट मैचों में रन तो रन होते हैं
SINGLE: मुझे इसे कोई फर्क नहीं पड़त कि रन कहा बने हैं क्योकि टेस्ट मैचो में रन तो रन होते है
MAML: मुझे इसे कोई फर्क नहीं पड़त कि रन कह बने ह क्योकि टेस्ट मैचो मे रन त रन होते ह

ORIGINAL: बजट तैयार करने में अहम भूमिका होती है आइए जानते हैं बजट तैयार करने वाली टीम के बारे में
SINGLE: बजट तैयार करने में अम भूमिका होती है आए जानते ह बजट तैयार करने वाली टीम के बारे मे
MAML: बजट तयार करने में अम भूमिका होती ह आए जानते बजट तैयार करने वाली टीम के बारे म

LANGUAGE: BENGALI

ORIGINAL: এটি ভারতে অনুষ্ঠিত সর্বকালের বৃহত্তম নির্বাচন।
SINGLE: এটি ভারতে অনুষ্ঠিত সর্বকালের বৃত্তম নির্বচন
MAML: এটি ভারতে অনুষ্ঠত সর্বকালে বৃহত্তম নির্বচন

LANGUAGE: TAMIL

ORIGINAL: இந்தியாவில் இதுவரை நடந்த மிகப்பெரிய தேர்தல் இதுவாகும்.
SINGLE: இந்தியாவில் இதுவர நடந்த மிகப்பெரிய தேர்தல் இதுவாகும்.
MAML: இந்தியாவில் இதுவர நடத மிகப்பெரிய தேரதல் இதுவாகும.

Figure 3: Output of some samples in Hindi, Bengali and Tamil Language for MAML and Single language trained model.

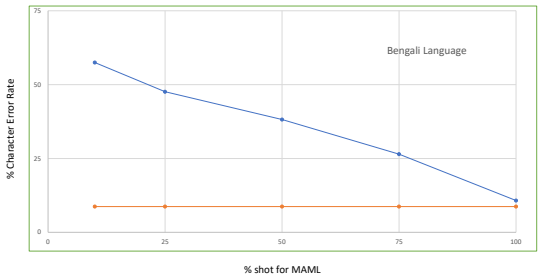


Figure 4: Few-shot results on Bengali Language using MAML vs Single Training.

4 Results and Discussion

4.1 Quantitative Analysis

As shown in Table 2, MAML performance is very close to the model when trained completely on a single language. We have used character error rate (CER) as evaluation criteria because Indic languages contain lot of vowel diacritic which sound similar but are different hence using word error rate (WER) to evaluate will not give correct information on performance of model. Our approach yields up to a 2-4% CER margin in the all-shot MAML and single training. This difference is attributed to low precision in prediction of vowel diacritic for MAML compared to single training. In Figure-3, you can see that sentence generated by MAML is readable and sensible but not entirely correct since there are few missing vowel diacritic.

4.2 Qualitative Analysis

We also evaluate the outputs produced by the model for both MAML based method and the single training method. We evaluate them using a mean human evaluation score that is averaged over 1000 samples for each language. This score is based on the correctness of output and fluency. The scoring is range 0-5 where 0 is for worst performance and 5 for best performance. The evaluation were done by five independent native speakers of each languages. The Table-3 shows the result of the mean human evaluation score for all the languages experimented with.

Few examples generated by both MAML based model and single-language trained model are given in Figure-3.

4.3 Efficacy of Few-Shot Fine-tuning

We investigate the number of samples required to observe performance improvement after fine-tuning the model. We start by training the model with a very small number of samples, i.e., 10%-25% of training data, where each sample approximately consists of 3-4 seconds of audio. We observe that the model cannot adapt to the target language with a such minuscule amount of data. We attribute this to the fact that our model is unable to capture the information from small audio samples due to a large amount acoustic variation in the data. Therefore, we increase the minimum threshold to 10% of the training data, and the model starts to adapt to the target language accordingly. We do this process until the threshold is set to 100%. Figure-4 shows the adaption and constant decrease in CER with an increase in fine-tuning data for the Bengali language.

5 Conclusion

In this paper, we analyse and evaluate the performance of our proposed method for automatic speech recognition in multilingual scenario for seven different low-resource Indic languages. We apply a fast adaptation method on Conformers using model-agnostic meta-learning (MAML) approach to learn a robust automatic speech recognition model to rapidly adapt to unseen languages. Based on the empirical results, MAML consistently comes close to single trained model using target unseen language with a margin of 2-4% CER in all such low-resource multilingual scenarios for Indic languages.

6 Acknowledgement

The work is done in collaboration with NVIDIA. We thank the support from Nvidia, India for providing the computing power and compute infrastructure requirements along with the software stack for the project. We would also like to thank our colleagues at IIIT Guwahati and ISI Kolkata for their valuable input during manual domain language analysis.

7 Note

The work was done when Kumari Deepshikha was at NVIDIA.

References

Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. 2014. Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on audio, speech, and language processing*, 22(10):1533–1545.

J. Cho, M. K. Baskar, R. Li, M. Wiesner, S. H. Mallidi, N. Yalta, M. Karafiát, S. Watanabe, and T. Hori. 2018. Multilingual sequence-to-sequence speech recognition: Architecture, transfer learning, and language modeling. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 521–527.

S. Dalmia, R. Sanabria, F. Metze, and A. W. Black. 2018. Sequence-based multi-lingual low resource speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4909–4913.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135, International Convention Centre, Sydney, Australia. PMLR.

Chelsea Finn, Kelvin Xu, and Sergey Levine. 2018. Probabilistic model-agnostic meta-learning. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Jiatao Gu, Yong Wang, Yun Chen, Victor O. K. Li, and Kyunghyun Cho. 2018. Meta-learning for low-resource neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3622–3631, Brussels, Belgium. Association for Computational Linguistics.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. 2020. Conformer: Convolution-augmented Transformer for Speech Recognition. In *Proc. Interspeech 2020*, pages 5036–5040.

Wei Han, Zhengdong Zhang, Yu Zhang, Jiahui Yu, Chung-Cheng Chiu, James Qin, Anmol Gulati, Ruoming Pang, and Yonghui Wu. 2020. Contextnet: Improving convolutional neural networks for automatic speech recognition with global context. *arXiv preprint arXiv:2005.03191*.

Jui-Yang Hsu, Yuan-Jui Chen, and Hung yi Lee. 2019. Meta learning for end-to-end low-resource speech recognition.

Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141.

Oddur Kjartansson, Supheakmungkol Sarin, Knot Pipatsrisawat, Martin Jansche, and Linne Ha. 2018. Crowd-Sourced Speech Corpora for Javanese, Sundanese, Sinhala, Nepali, and Bangladeshi Bengali. In *Proc. The 6th Intl. Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU)*, pages 52–55, Gurugram, India.

Samuel Kriman, Stanislav Beliaev, Boris Ginsburg, Jocelyn Huang, Oleksii Kuchaiev, Vitaly Lavrukhin, Ryan Leary, Jason Li, and Yang Zhang. 2019. Quartznet: Deep automatic speech recognition with 1d time-channel separable convolutions. *arXiv preprint arXiv:1910.10261*.

Jason Li, Vitaly Lavrukhin, Boris Ginsburg, Ryan Leary, Oleksii Kuchaiev, Jonathan M Cohen, Huyen Nguyen, and Ravi Teja Gadde. 2019. Jasper: An end-to-end convolutional neural acoustic model. *arXiv preprint arXiv:1904.03288*.

Tara N Sainath, Abdel-rahman Mohamed, Brian Kingsbury, and Bhuvana Ramabhadran. 2013. Deep convolutional neural networks for lvcsr. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 8614–8618. IEEE.

Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition.

Sibo Tong, Philip N. Garner, and Hervé Bourlard. 2017. An investigation of deep neural networks for multilingual speech recognition training and adaptation. In *Proceedings of Interspeech*, pages 714–718, Stockholm, Sweden.

Sibo Tong, Philip N. Garner, and Hervé Bourlard. 2018. Multilingual training and cross-lingual adaptation on ctc-based acoustic model.

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

N. T. Vu, D. Imseng, D. Povey, P. Motlicek, T. Schultz, and H. Bourlard. 2014. Multilingual deep neural network based acoustic modeling for rapid language adaptation. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7639–7643.

G. I. Winata, S. Cahyawijaya, Z. Lin, Z. Liu, and P. Fung. 2020. Lightweight and efficient end-to-end speech recognition using low-rank transformer. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6144–6148.

Genta Indra Winata, Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. 2019. Code-switched language models using neural based synthetic data from parallel sentences. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 271–280.

Qian Zhang, Han Lu, Hasim Sak, Anshuman Tripathi, Erik McDermott, Stephen Koo, and Shankar Kumar. 2020. Transformer transducer: A streamable speech recognition model with transformer encoders and rnn-t loss.

TPT: An Empirical Term Selection for Arabic Text Categorization

Mourad Abbas
High Council of Arabic Language
Algiers, Algeria
abb.mourad@gmail.com

Mohamed Lichouri
Algiers, Algeria
medlichouri@gmail.com

Abstract

In this paper, we will investigate an empirical term selection method for text categorization, namely Transition Point (TP) technique, and we will compare it to two other widely used methods: Term Frequency (TF) and Document Frequency (DF). For evaluation, we have used the well-known TFIDF technique. Experiments have been conducted by using the Arabic corpus Khaleej-2004 which is composed of 4 categories. The results obtained from this study show that performance is almost the same for the three techniques. However, we should note that TP is advantageous since it uses a vocabulary much smaller than the ones used in TF and DF.

1 Introduction

Up to now, all studies about text categorization that have been carried out by using different approaches and algorithms led to relatively satisfactory results, but did not lead to a hopeful and significant improvement, and so did many systems which are based on machine learning and statistical approaches. This is due in part to term selection methods which are not powerful enough. More studies and experiences must be done intensively to achieve this goal. In this regard, we focus our interest on comparing a relatively recent technique namely Transition Point to the two well-known Term Frequency and Document Frequency methods. These last two are used as a reference in this work since they are ranked, in many studies, among the most efficient feature selection methods. The TP technique has been used in different works like: text categorization (Moyotl Hernández and Jiménez Salazar, 2004), (Moyotl-Hernández and Jiménez-Salazar, 2005), summarization (Bueno et al., 2005), clustering of short texts (Jimenez et al., 2005), keyphrases extraction (Tovar et al., 2005), and weighting models for information retrieval systems (Cabrera et al., 2005). After using this feature selection technique in the aforementioned works, we believe that using it as a term selection process for text categorization could yield good performance. The focus in this paper is mainly evaluating the TP technique by using an Arabic corpus and comparing it to TF and DF. Section 2 is dedicated to related work. Section 3 describes succinctly the TP technique. Section 4 presents the experiments and the results. Finally, we conclude in section 5.

2 Related Work

Many works on feature selection for text categorization, particularly Term Frequency and Document Frequency had been achieved. In fact, Term frequency had been used in (Yang and Pedersen, 1997), (Abbas et al., 2010), (Abbas et al., 2011) and yielded good results. The selected words are those of high frequency of occurrence. Indeed, the stop words are not included in this selection. For Arabic, Term Frequency had been tested using Khaleej-2004 corpus (Abbas and Smaili, 2005) and the corresponding results are shown in Figure 3. For Document Frequency, the basic idea is that rare terms are non-informative for category prediction. Hence, terms whose document frequency are less than a predetermined threshold are removed. Yang and Pedersen show in (Yang and Pedersen, 1997) that reducing the training corpus by a factor of 10 did not affect performance, and caused only a slight degradation when reducing it by a factor of 100. They stated that the performance achieved by using DF is good and approximates that of Information Gain. In (Brun, 2003) it had been shown that the performance of the TFIDF technique using two vocabularies [1] obtained from TF and DF were about 74.3% and 83.1%, respectively[2]. However, only few works are dedicated to Transition Point. Indeed, Pinto et al. proposed a procedure to cluster abstracts of scientific texts by applying the

[1] The size of the 2 vocabularies are 30000 distinct words.
[2] Here, performance is in terms of Recall.

transition point technique during the term selection process (Pinto et al., 2006). They found that TP outperforms TF and TS (Term Strength). From the research conducted by Moyotl and Jimenez (Moyotl Hernández and Jiménez Salazar, 2004) in which they tested DF, Information Gain and χ^2 in combination with TP, it shows that the DF-TP pair gives the best result. They concluded also that selecting terms lesser than the transition point discarded noise terms with maintaining the performance of categorization. We consider these preliminary encouraging results as the main motivation for testing TP. In the following, we will describe the TP method.

3 Transition Point Technique

TP technique is based on the Zipf Law (Zipf, 2016) and also on the studies of Booth (Booth, 1967). In these works, it has been shown that terms of medium frequency are narrowly related to the content of a document (Pinto et al., 2006). This is the motivation for using the terms whose frequency is closer to TP as indexes of a document. It should be noted that TP is a frequency that splits the vocabulary of a document into two groups of terms, with respectively high and low frequency. It can be calculated by using the formula (1):

$$TP_V = \frac{\sqrt{8I_1 + 1}. - 1}{2} \quad (1)$$

I_1 stands with the number of words occurring once in the text T. According to Booth's law (Booth, 1967), TP_V can be determined by identifying the lowest frequency, among the highest frequencies, that is not repeated. Hence, the first task to realize is to extract a list of terms with their corresponding frequencies, from the text T. The result is a frequency-sorted vocabulary given by: $V = [(t_1, f_1), ..., (t_n, f_n)]$, with $f_k \geq f_{k-1}$, then $TP_V = f_{k-1}$ if $f_k = f_{k+1}$. After identifying TP, the most important words would be those that frequencies are the closest to TP value (Pinto et al., 2006). These words are presented by the expression:

$$V_{TP} = \{t_k | (t_k, f_k) \in V, U_1 \leq f_k \leq U_2\}, \quad (2)$$

U_1 and U_2 are respectively lower and upper threshold, they can be calculated by using the formulas:

$$U_1 = (1 - NTP).TP_V \quad (NTP \in [0, 1]) \quad (3)$$

$$U_2 = (1 + NTP).TP_V \quad (NTP \in [0, 1]) \quad (4)$$

4 Experiments and Results

Our experiments are carried out by using the well-known TFIDF method. This type of technique is based on the relevance feedback algorithm proposed by Rocchio (Rocchio, 1971). The idea of the TFIDF algorithm is to represent each document d by a vector D = (d_1, d_2,..., d_v) in a vector space. The vector elements are calculated as the combination of the term frequency $TF(w, d)$, which is the occurrence number of the word w in the document d, and the inverse document frequency $IDF(w)$ (Salton, 1991; Rosenfeld and Huang, 1992). $DF(w)$ is the number of documents in which the word w occurs at least once. The value d_i is called the weight of word w_i in the document d, and is given by : $d_i = TF(w_i, d) * IDF(w_i)$ with $IDF(w_i) = log(DF(w_i)/N)$, where N is the total number of documents. In order to calculate the similarity between a document D_i and the category D_j we used the equation 5. A document is assigned to the category which gives the highest similarity.

$$Sim(D_j, D_i) = \frac{\sum_{k=1}^{|V|} d_{jk} d_{ik}}{\sqrt{\sum_{k=1}^{|V|} (d_{jk})^2 \sum_{k=1}^{|V|} (d_{ik})^2}} \quad (5)$$

4.1 Khaleej-2004 corpus

We built Khaleej-2004 corpus[3] by downloading thousands of articles from an online arabic newspaper. The corpus is divided into four categories, namely:*Sports*, *International news*, *Local news* and *Economy*. Table 1 shows the number of documents for each category. We carried out the usual operations for data preprocessing such as removing all signs of punctuation and stop words. An overview on the size of the corpus before and after removing stop words is presented in Table 2. The size of the resulted corpus becomes 2.172.000 words, i.e reduced by 23.90%.

4.2 Terms Representativity

High frequencies of words usually indicate that they are more informative (except stop words).

[3]Khaleej-2004 corpus had been released in 2010, it can be downloaded from:
(http://sites.google.com/site/mouradabbas9/corpora).
(http://sourceforge.net/projects/arabiccorpus/files).

Table 1: Khaleej-2004 corpus

Topic	Documents	Words
Economy	909	578.000
Int.news	953	754.000
Loc.news	2398	893.000
Sports	1430	628.000
Total number	5690	2.853.000

However, some words of high frequency of occurrence are not informative at all since they belong to more than one category and their frequencies are not very different from each other. For example, the word *year* \' *A m*\, which is considered among the most frequent words in *Economy* category, is not representative because its frequency is also high in the other categories. The 11 most frequent words in each category are extracted from their related corpora and presented in Table 3, written in International Phonetic Alphabet (IPA) and translated to English. Of course, choosing this limited number of words presents simply an illustration to give an overview on the advantages of term frequencies and their limits of being informative in the representation of categories. Other words, in contrast, represent faithfully and rigorously their categories. For example, as presented in Figure 1, the word *Match* \m b A r A t\ which is the most frequent word in the *Sports* category is very rare to find in other categories. It is the same for the word *American* \' m r I k y h\ that we found only in the *International news* category [4], because America is frequently present on the international scene.

In Figure 1, subfigures (a), (b), (c) and (d) present the distribution of the top selected terms extracted from the categories *Local news*, *Sports*, *International News* and *Economy*, respectively. We define this distribution as relative frequencies (RF) of terms, given by the values $RF = F_w/N_c$, where F_w stands for the frequency of the word w in the category C and N_c represents the total number of the category C.

In each of these subfigures, four curves are presented. One curve concerns the distribution of the words of the category in question, and the three other ones deal with the distribution of the same words over the remaining categories.

[4]Within the eleven extracted words.

4.3 Experiments on Transition Point Technique

Booth presented interesting ideas about occurrences of words (Booth, 1967) and tried to extract a law which purpose is to explain and illustrate the case of words of very low frequency of occurrence. For instance, he studied the ratios I_1/D, i.e. the ratio of the number of words occurring once to the number of different words for each of the texts. He equally investigated *the remarkable constancy* of this ratio. All the experiments realized by Booth have used English texts [5]. However, he stated that *there is no reason to suppose that the rather arbitrary assumption used to deduce* I_1 *would be equally valid in languages other than English.*
This is another motivation for us to test and evaluate TP technique on Arabic corpora. Terms of high frequency of occurrence are known to be more representative, and then allow to have good results for text categorization tasks. However, The statistics of terms' representativity presented in subsection 4.2 show that some terms are of high frequency of occurrence in all the 4 categories, which means that they are not representative even they are highly occurring. -see Figure 1-.
Relying on Booth assumptions and on the results presented in subsection 4.2, the TP could be viewed as an idea which allow to extract efficient features. Indeed, the idea to find a value TP_V and then extract the terms localized around it seems to be efficient and outperforms the methods based only on high frequencies of terms.
We obtained different vocabularies by using different values of NTP. Figure 2 plots recall and precision values given many NTP values. As can be seen in this figure, it shows that performance curves related to the four categories increase with NTP. Global values of Recall and Precision for $NTP = 0.9$ are equal to 90.75% and 90.5% respectively. The best result was for DF, indeed Recall was about 91.75% and Precision 91.50%. While TF outperforms TP very slightly (Recall=90.80% and Precision=91.20%). The most important point to be mentioned is that the result achieved by TP is obtained by using a vocabulary size of about 2500 distinct words, which is a very small size in comparison with the vocabularies obtained from TF and DF which sizes attain

[5]Three texts (Western Reserve University), and the sampling of newspaper English published by Eldridge (Eldridge, 1911).

Table 2: The corpus before and after stop words removing.

Topic	Economy	Int. news	Loc. news	Sports
Corpus before	578.000	754.000	893.000	628.000
Corpus after	440.000	567.000	680.000	485.000

Table 3: Most frequent words for each category

Sports		Int. News	
English	IPA	English	IPA
Match	\m b A r A t\	President	\r ’ I s\
Team	\f r I q\	Iraq	\’ i r A q\
Center	\m r k z\	Year	\’ A m\
Championship	\b ṭ U l h\	United	\m t ḥ d h\
Team	\m n t kh b\	Forces	\q w A t\
Year	\’ A m\	Government	\ḥ k U m h\
First	\’ w l\	American	\’ m r I k y h\
Olympic	\’ U l m b y h\	Past	\m A ḍ I\
second	\th A n I\	States	\w l A y A t\
Tournament	\d w r h\	Council	\m zh l s\
Ball	\k r h\	Elections	\’ n t kh A b A t\
Loc. News		Economy	
English	IPA	English	IPA
Year	\’ A m\	Year	\’ A m\
Work	\’ m l\	Countries	\d w l\
Ministry	\w z A r h\	Work	\’ m l\
Festival	\m h r zh A n\	Turf	\q ṭ A ’\
Activities	\f ’ A l y A t\	Oil	\n f ṭ\
General	\’ A m h \	Million	\m l y U n\
Health	\s ḥ y h\	Market	\s U q\
Instruction	\t ’ l Y m\	Company	\sh r k h\
Center	\m r k z\	Companies	\sh r k A t\
Number	\’ d d\	Council	\m zh l s\
Zone	\m n ṭ q h\	Trade	\t zh A r h\

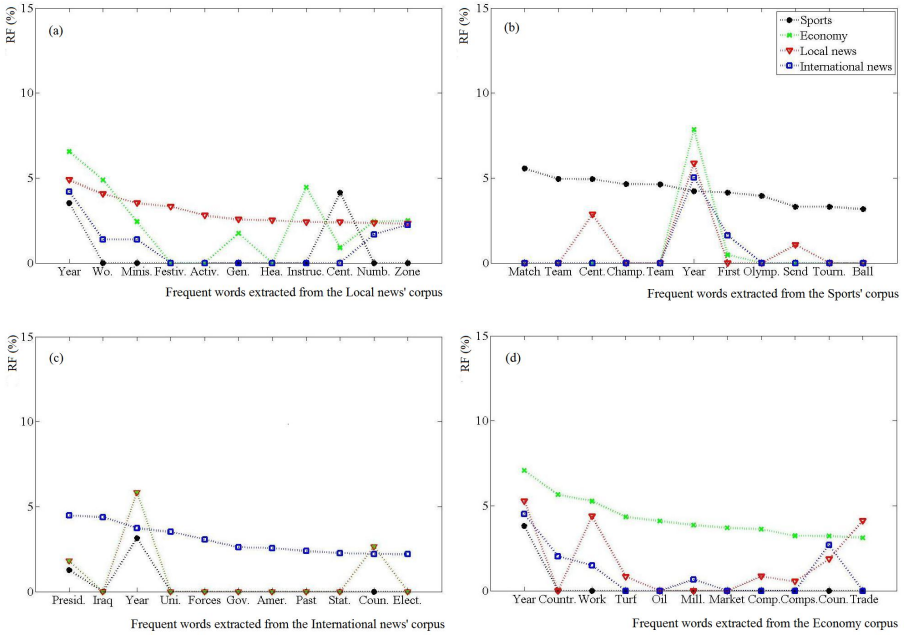


Figure 1: Behavior of RF values of each set of words - presented in Table 3-

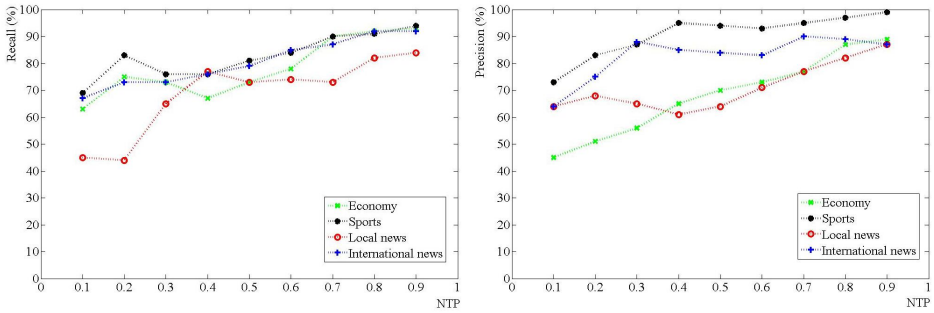


Figure 2: Recall and Precision versus NTP values

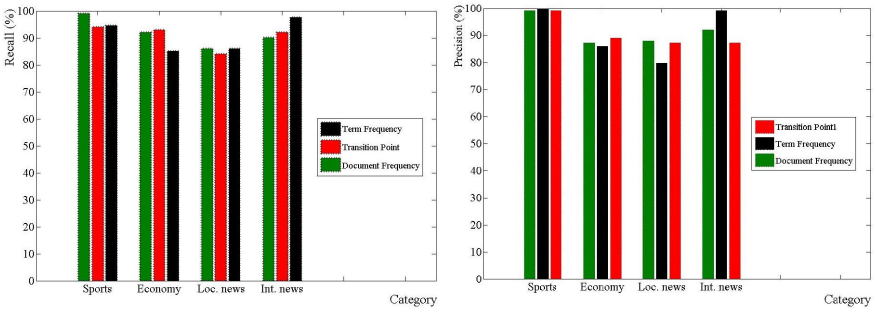


Figure 3: Performance of TP, TF and DF for each category

40000 words. Figure 3 presents the performance of TP, TF and DF related to each category.

5 Conclusion

The work presented in this paper is a contribution for the evaluation of the TP technique by using an Arabic corpus. Based on the findings, the obtained results seem to be consistent with other research (Moyotl Hernández and Jiménez Salazar, 2004), (Pinto et al., 2006). The strong point of TP is that we achieved good performance by using a very small corpus. TF and DF are used widely for extracting features. Indeed, in the case of TF, the selected ones are those of high frequency of occurrence. However we presented in section 2 some examples of non-informative words though they are highly frequent. This, we believe that TP could be a good feature selection method for the reasons that we mentioned above. Nevertheless, many other experiments on TP should be realized by using different text collections in order to prove its efficiency. In addition, more efforts must be carried out to find the best method which allows to extract TP because, up to now, it is computed empirically.

References

M Abbas, K Smaili, and D Berkani. 2011. Evaluation of topic identification methods for arabic texts and their combination by using a corpus extracted from the omani newspaper alwatan. *Arab Gulf Journal of Scientific Research*, 29(3-4):183–191.

Mourad Abbas and Kamel Smaili. 2005. Comparison of topic identification methods for arabic language. In *Proceedings of International Conference on Recent Advances in Natural Language Processing, RANLP*, pages 14–17.

Mourad Abbas, Kamel Smaïli, and Daoud Berkani. 2010. Efficiency of tr-classifier versus tfidf. In *2010 First International Conference on Integrated Intelligent Computing*, pages 233–237. IEEE.

Andrew D Booth. 1967. A "law of occurrences for words of low frequency. *Information and control*, 10(4):386–393.

Armelle Brun. 2003. *Détection de thème et adaptation des modèles de langage pour la reconnaissance automatique de la parole*. Ph.D. thesis, Université Henri Poincaré-Nancy 1.

Claudia Bueno, David Pinto, and Héctor Jiménez. 2005. El párrafo virtual en la generación de extractos. *Research on Computing Science*, 13:83–90.

Rubı Cabrera, David Pinto, H Jimenez, and D Vilarino. 2005. Una nueva ponderación para el modelo de espacio vectorial de recuperación de información. *Research on Computing Science*, 13:75–81.

RC Eldridge. 1911. *Six Thousand Common English Words: Their Comparative Frequency and What Can Be Done with Them*. Clement Press.

Héctor Jimenez, David Pinto, and Paolo Rosso. 2005. Selección de términos no supervisada para agrupamiento de resúmenes. In *proceedings of Workshop on Human Language, ENC05*, pages 86–91.

Edgar Moyotl Hernández and Héctor Jiménez Salazar. 2004. An analysis on frequency of terms for text categorization. *Procesamiento del lenguaje natural, nº 33 (septiembre 2004); pp. 141-146*.

Edgar Moyotl-Hernández and Héctor Jiménez-Salazar. 2005. Enhancement of dtp feature selection method for text categorization. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 719–722. Springer.

David Pinto, Héctor Jiménez-Salazar, and Paolo Rosso. 2006. Clustering abstracts of scientific texts using the transition point technique. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 536–546. Springer.

Joseph Rocchio. 1971. Relevance feedback in information retrieval. *The Smart retrieval system-experiments in automatic document processing*, pages 313–323.

Ronald Rosenfeld and Xuedong Huang. 1992. Improvements in stochastic language modeling. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*.

Gerard Salton. 1991. Developments in automatic text retrieval. *science*, 253(5023):974–980.

Mireya Tovar, Maya Carrillo, David Pinto, and H Jimenez. 2005. Combining keyword identification techniques. *Research on Computing Science*, 14:157–162.

Yiming Yang and Jan O Pedersen. 1997. A comparative study on feature selection in text categorization. In *Icml*, volume 97, page 35. Nashville, TN, USA.

George Kingsley Zipf. 2016. *Human behavior and the principle of least effort: An introduction to human ecology*. Ravenio Books.

From local hesitations to global impressions of the listener

Tanvi Dinkar
LTCI, Télécom Paris
IP Paris, France
tanvi.dinkar@telecom-paris.fr

Beatrice Biancardi
LTCI, Télécom Paris
IP Paris, France
beatrice.biancardi@telecom-paris.fr

Chloé Clavel
LTCI, Télécom Paris
IP Paris, France
chloe.clavel@telecom-paris.fr

Abstract

The listener's interpretation of a speaker's utterance includes estimates about the speaker's commitment to what they are saying. Previous works have shown that fillers (e.g. "um") are linked to both the speaker's metacognitive state, and the listener's impression of a speaker's state. However, these results are limited to contexts that may not apply to spontaneous speech. Additionally, there is a lack of hierarchical analysis of the discourse; i.e. how a speaker's local use of fillers could lead to a listener's overall impression. In this work, we address these limitations by studying how does a speaker's use of fillers relate to the incoming message, and consequently, what is the resulting impression formed by the listener. We do so by analysing a dataset of English monologue movie reviews, where the speakers voluntarily and naturally recorded themselves giving a movie review. Our findings show that speakers tend to stylistically use fillers in the incoming message before introducing new information related to the review, and that listeners may not associate this specific use of fillers with their estimate of the speaker's expressed confidence. Our results highlight that there are potentially different metacognitive effects from the speaker's use of fillers on the listener.

1 Introduction

There is a complex relationship between what a speaker says versus the way that a speaker says it; and consequently, what is the resulting impression left on the listener. Consider the following example, taken from Brennan and Williams (1995):

A: Can I borrow that book?
B: . . . {F um} . . . all right.

In the above example, speaker **B** used a filler {F...} (Clark and Fox Tree, 2002), which is a sound filling a pause in an utterance or a conversation. We see that the filler causes **A** to note that **B** might have had a different intention compared to if **B** answered "all right" immediately. While **B** in essence says "yes" to lending the book, the way **B** said this *implicitly* indicates some uncertainty or hesitation.

The aim of this work is to empirically study on a real-life dataset, whether the utterance level use of fillers can help in understanding/ interpreting the perception of the speaker that was formed by a listener. The present work is based on the following observations: According to Brennan and Williams (1995), the listener's interpretation of the speaker's utterance includes estimates about the speaker's commitment to/ expressed confidence in what they are saying. Flavell (1979) termed these processes (of the speaker) as **metacognitive** ones, that is cognition about cognitive phenomena, or more simply "thinking about thinking". While the field of metacognition was initially in the context of children's development and education; the idea of metacognitive states is applicable a wide variety of communicative scenarios. When considering the comprehension of disfluent speech for e.g., research has linked fillers to the listener's assessment of a speaker's metacognitive state (Brennan and Williams, 1995). However, these results may not apply to spontaneous speech datasets collected in real-life contexts, or non-QA datasets. Additionally, the focus of analysis tends to be on utterances as if they occur in isolation, rather than part of an overall discourse.

Thus existing studies do not focus on the connection between the hierarchical levels of discourse; i.e. how a speaker's local use of fillers could lead to a listener's overall (global) impression of the speaker. In this work, we study how does a speaker's use of fillers relate to the incoming message from the speaker, and consequently, how does that relate to a listener's perception of the speaker.

We do so by studying a dataset of publicly available English monologue movie reviews, where the speakers voluntarily and naturally recorded themselves giving a movie review. These video reviews were collected from a social media platform that was created for the purpose of enabling speakers to upload their unbiased opinions towards products (in this case, a movie) to a large, but unseen audience. Annotators (listeners) were asked to label the reviews for attributes such as "confidence"; without explicitly being told to pay attention to the speaker's use of fillers. Our findings suggest that speakers stylistically do tend to use fillers in the incoming message, when introducing a new entity (to indicate new information), rather than an entity already introduced into the discourse. Our results also suggest that the occurrence of fillers before new entities may not have an effect on the listener's perception of the speaker's expressed confidence, despite previous works that suggest the link between fillers and expressed confidence. This does not discount other possible metacognitive aspects, such as the listener may expect a speaker to use fillers typically when the speaker is introducing new information in the incoming message. The rest of the paper is organised as follows: in section 2, we overview the theoretical foundations and research questions of our study, section 3 describes the dataset, section 4, the methodology, section 5, the results and discussions of the work, and section 6, the conclusion.

2 Background and Research Questions

2.1 Metacognition and the listener's perspective

When a speaker says an utterance, this articulation process includes an estimation of their commitment/ certainty about what they are saying. Research suggests that fillers and prosodic cues are linked to a speaker's metacognitive state, specifically; their *Feeling of Knowing (FOK)* or **expressed confidence** — a speaker's certainty or commitment to a statement (Smith and Clark, 1993). A speaker may *encode* meaning into their utterance using fillers, but the onus is on the listener to *decode* this information; making the interpretation of fillers contextual and dependent on the listener. Brennan and Williams (1995) observed that fillers and prosodic cues contribute to the listener's perception of the speaker's metacognitive state; which they refer to as the *Feeling of Another's Knowing (FOAK)*.

Other studies also focus on the comprehension of disfluent speech, i.e. taking into account the listener's understanding of the speaker's disfluencies (Corley and Stewart, 2008), and not on why the disfluency itself was produced (Nicholson, 2007). For example, Vasilescu et al. (2010) observe that the French "euh" has both *disfluent* (signalling production difficulties of the speaker) and *fluent* (as a discourse marker – to bracket lexical units that may aid in listener comprehension) properties. Related to metacognition, research suggests that following fillers, listeners may expect a speaker to shift topics, as they carry information about larger topical units (Swerts, 1998), that the use of fillers biases listeners towards new referents rather than ones already introduced into the discourse (Arnold et al., 2004), relax listener's expectations when hearing an unpredictable word (Corley et al., 2007), and that listeners expect the speaker to refer to something new following the filler "um", compared to noise of the same duration (such as a cough or sniffle) (Barr and Seyfeddinipur, 2010). In the present paper, we focus on the listener's comprehension of disfluencies. As Corley and Stewart (2008) state, "it is hard to determine the reason that a speaker is disfluent, especially if the investigation is carried out after the fact from a corpus of recorded speech". We analyse the speaker's use of fillers from the incoming message from a corpus of previously recorded speech, and then observe what effect this may have on the listener's perception.

Drawbacks of current works Corley and Stewart (2008) illustrate that the results observed in Brennan and Williams (1995) that link fillers to FOAK, could have been influenced by the listener's being asked explicitly to rate speaker confidence/certainty on the speaker's short answer to a question (which may have included a filler). While this effect has been observed in other scenarios, for e.g. in human-machine interaction (Wollermann et al., 2013), it was still based on single utterance responses. These studies were appropriately targeted towards a QA setting. In a similar line of reasoning, Schrank and Schuppler (2015) show the drawbacks in research on automatic uncertainty detection[1], due to the narrow range of question-answering (QA) datasets commonly utilised. In general, this shows that when listener's are asked

[1]Which among other features, can use prosodic cues and the presence of fillers

to evaluate a speaker's certainty on shorter utterances, it could direct the listener towards paying attention to the fillers used by the speaker. Moreover, perceived uncertainty of the speaker in *local* utterances could still lead to a different *global* impression. Thus, there is a lack of evidence to support this effect on more spontaneous speech datasets.

Recently, Dinkar et al. (2020a) found that in an unsupervised manner, fillers can indeed be a discriminative feature in the automatic prediction of a listener's impression of a speaker's confidence. These results empirically solidified an effect that was often assumed to be true (and indeed, fillers are sometimes interchangeably used with the term "hesitations" in certain works (Pickett, 2018; Corley and Stewart, 2008)). However, the study simply focused on the overall impression the listener had of the speaker, i.e. the global, and did not account for more fine-grained information shared by the speaker.

2.2 Research Questions and Hypothesis

While work such as in Dinkar et al. (2020a) is important as preliminary analysis, they do not account for how fillers locally interact with the rest of the message in a holistic way. Clark (1996); Clark and Fox Tree (2002) proposed that speakers are able to utilise fillers as *collateral signals* in communication, in addition to the *primary signal* of the message. We colloquially refer to the primary signal of the message as *what* was said (in essence) and the collateral signal as *how* it was said. In Spoken Language Understanding (SLU), a similar phenomenon occurs of separating these two signals. However, in this context, reducing an input utterance into its primary signal (or *what* was said in essence) is standard practice (e.g. as seen in Tur and De Mori (2011), chapter 13. Speech Summarization). Indeed, in dialogue systems, the output transcripts of automatic speech recognisers are often cleaned of disfluencies such as fillers in post-processing, despite work relevant to the area that shows for e.g. the link between fillers and opinions (Le Grezause, 2017; Levow et al., 2014; Dinkar et al., 2020a), or the rich linguistic literature to suggest otherwise (Clark and Fox Tree, 2002). And yet, even recent work such as Barr and Seyfeddinipur (2010) support the collateral signal account, specifically that the listener is able to process fillers as a collateral signal (even if unclear whether the speaker (un)intentionally used them as such). This is an important finding, as it shows that perhaps the listener's attention is drawn to the cognitive state of the speaker. The problem then, as stated in Clark and Fox Tree (2002), remains about how to merge the two signals. Given the rapid advancements of dialogue systems, and growing interest in SLU, there is a need to move towards an automatic but holistic analysis of both together; if we hope to move towards better models and understanding of spontaneous speech. Thus the research questions are as follows:

RQ1: (Local effect of fillers): How does a speaker's use of fillers relate to the incoming message from the speaker? From the findings of Barr and Seyfeddinipur (2010); Arnold et al. (2004) as discussed in section 2, we would like to empirically analyse the role fillers play in a dataset of spontaneous speech, specifically related to new information from the incoming message of the speaker. Since the dataset we choose to study is a dataset of English monologue movie review videos (please refer to section 3), we consider the speaker's mention of terms related to the movie annotated from metadata, such as actors and directors.

- **H1** Fillers are more likely to occur before the introduction of new and upcoming information in the review.

RQ2: (Global effect of fillers): How does the speaker's use of fillers relate to a listener's perception of the speaker? We would like to empirically analyse whether the speaker's use of fillers has an impact on the listener's overall impression of the speaker.

- **H2** From H1, the speaker's use of fillers preceding new information in the incoming message contributes to the listener's perception of the speaker's confidence.

  Specifically, we hypothesise that when fillers are predominantly used in the context of preceding new information, listener's may judge the expressed confidence of the speaker as high, and listeners may only notice when fillers are used in other contexts (for e.g. as seen in Tottie (2014), listeners notice fillers when they are overused or used in the wrong context) which consequently will decrease the expressed confidence rating.

3 Materials

Persuasive Opinion mining (POM) dataset For this work, we choose the POM dataset (Park et al., 2014), a dataset of 1000 (American) English monologue movie review videos. Speakers recorded themselves (video and audio) giving a movie review, which they rated from 1 star (most negative) to 5 stars (most positive). The movie review videos are freely available on ExpoTV.com, and are completely in the wild; speakers were simply reviewing a movie without the knowledge that their review would eventually be annotated for such a context. 3 annotators (or listeners) per video were then asked to label the movie reviews for high level attributes, such as confidence. We think this dataset is particularly relevant for the following reasons: 1. Since this is a dataset of monologues, it allows us to focus uniquely on the role of fillers (Swerts, 1998). This is because the speaker is conscious of an *unseen* listener, but is not interrupted by the listener with other dialogue related disfluencies, such as backchannels ("Uh-huh"). This also minimises some turn-taking properties of fillers, such as when they are used to hold the speaker turn. Additionally, the annotators were never asked to pay special attention to the speaker's use of fillers. 2. Filler annotations of "uh" and "um" have been manually transcribed. Each transcription of a movie review video was reviewed by experienced transcribers for accuracy after being transcribed via Amazon Mechanical Turk (AMT) (Park et al., 2014). The experience of the transcriber is important, as Zayats et al. (2019) shows that transcribers tend to misperceive disfluencies and indeed, this can affect the transcription of fillers (Le Grezause, 2017). The filler count of this dataset is high (roughly 4% of the transcriptions, for comparison, the Switchboard (Godfrey et al., 1992) dataset of human-human dialogues, consists of $\approx$ 1.6% of fillers (Shriberg, 2001)). Sentence markers have been manually transcribed, with the practice of the filler being annotated sentence-initially, if the filler occurs between sentences (in this dataset, utterance segmentation is not available, and is interchangeable with sentence). 3. The inter-annotator agreement for several attributes is high; with confidence (which we use to denote the *FOAK*, or the listener's perception of the speaker's expressed confidence) (Krippendorff's alpha = 0.73), (Park et al., 2014). For confidence annotators were asked "How confident was the reviewer", and had to rate the speaker on a Likert scale of 1-7 with given labels: 1 (not confident), 3 (a little confident), 5 (confident) and 7 (very confident). Additional details can be found in Park et al. (2014). Summary statistics, that have been taken from Dinkar et al. (2020b), are given in Table 1.

Description	Value
Reviews that contain fillers	**792**
Total number of review used	892
Total *um* fillers in the corpus	4969
Total *uh* fillers in the corpus	4967
Total fillers in the corpus	**9936**
Number of tokens in the corpus	230462
% of tokens that are fillers	**4.31**
Average length (in tokens) of a review	255.9

Table 1: Details about the POM dataset.

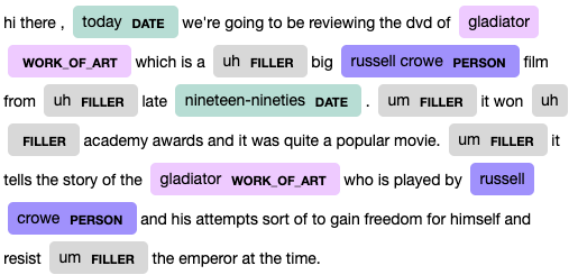


Figure 1: An example transcript that has annotated entities (in colour) using the EntityRuler. As shown, patterns from the metadata (e.g. "russell crowe") are added to the existing set (e.g. "nineteen-nineties"). Fillers are marked in grey. The first mention of "russell crowe" would be considered a new entity mentioned, while the second, an old one. Note, while the entity annotation is fairly reliable given the metadata, it is not exact. For e.g. the EntityRuler sometimes mislabels entities (the second mention of the word "gladiator".)

4 Methodology

4.1 RQ1 How does a speaker's use of fillers relate to the incoming message from the speaker?

H1 Fillers are more likely to occur before the introduction of new information in the review.

We consider the speaker's mention of entities related to the movie, that we extract from metadata files[2]. These entities could be categorised into actor, director or title of the movie. We then add these custom entities to SpaCy's EntityRuler, a rule

[2]The complete code and processed data will be made available online for reproducibility here https://github.com/tdinkar/fillers_in_POM.git

based named entity recogniser[3]. We prepossess the files (e.g. so that the filler annotations match the fillers in the existing model's vocabulary). We map the entities to match the existing patterns in the EntityRuler, for e.g "actor" is converted to "PERSON", by adding to the already existing entity patterns (please refer to Figure 1). The tagging of entities follows the *BIO* format (beginning, inside and outside of an entity).

To investigate H1, we inspect for each transcript, the distribution of filler positions, in relation to the automatically annotated entities in the discourse (denoted by Ent). We split these entities into Ent_new; i.e. entities newly introduced in the discourse, to indicate new information in the incoming message, and Ent_old to indicate entities already introduced in the discourse. We specifically note the order of the tokens in the transcripts for the filler positions and the first token of the 1. Ent_new (the first occurrence of the Ent) and 2. Ent_old (the second and following occurrences of each Ent), using the *B* tag of the Ent. Then, we check whether the distributions of filler positions (by its token position in the transcript) are significantly different compared to the distributions of 1. Ent_new and 2. Ent_old positions (by its first token's position), by utilising a Kruskal-Wallis H test[4] and use the Benjamini-Hochberg procedure for multiple testing correction. We then estimate the effect size by computing Cliff's Delta δ[5]. Lastly, we compare the δ distributions of the two experiments, i.e. fillers with Ent_new versus fillers with Ent_old using a Wilcoxon signed-rank test, to see if they significantly differ.

4.2 RQ2 How does the speaker's use of fillers relate to a listener's perception of the speaker and review?

H2 From H1, the speaker's use of fillers preceding new information contributes to the listener's perception of the speaker's confidence.

To investigate H2, we take the mean of the three confidence labels provided by the three annotators as the final rating of the speaker giving the review. We then consider reviews that are categorised as high-confidence (HC) and low-confidence (LC). Since confidence ratings are positively skewed[6] we take ratings of 3 (a little confident) and below to denote LC speakers, and 6 and above to denote HC speakers. The resulting size of the categories are 130 HC and 116 LC speakers. To calculate the percentage of fillers preceding new information (denoted by a new entity), we first consider the Ent_new labels that were automatically annotated in H1. We then count the number of fillers in the review that occur before (but not after) an Ent_new, constrained to a maximum distance of 1 token in between the filler and Ent_new. We normalise by dividing this count by the total number of fillers used in the review. From this, we obtain the percentage of fillers that occur before an Ent_new versus the percentage of fillers used in any other context that is not Ent_new. We then sum these two values for all HC and LC reviews, to get a cumulative percentage (please see Table 2).

Table 2: OR contingency table, where NE stands for the cumulative percentage of fillers that occur preceding an Ent_new for all HC (a) / LC (b) reviews, and OC the remaining cumulative percentage of fillers used in other contexts ((c) and (d) respectively).

		Outcome	
		HC	LC
Exposure	NE	a	b
	OC	c	d

We compute Odds Ratios (ORs) in order to investigate whether the use of fillers around new entities is associated with confidence. Odds ratios are an association measure that represents the odds that an outcome will occur given a particular exposure, compared to the odds that the outcome will occur in the absence of that exposure. Here, the odds denote the outcome of HC or LC, given the occurrence of fillers before new entities, compared to the occurrence of fillers that do not occur before new entities. We expect that the more fillers are used in the context of preceding new entities, the greater the odds of HC.

$$OR = \frac{odds_{HC}}{odds_{LC}}$$

where $odds_{HC} = a/c$ and similarly $odds_{LC} =$

[3] https://spacy.io/api/entityruler

[4] We utilise this method according to the guidelines given in the scipy software (https://scipy.org/) where the test is only run if the samples for each category ≥ 5. We calculate Cliff's delta regardless of this criteria.

[5] Utilising effect size tools from https://github.com/ACCLAB/DABEST-python/blob/master/dabest/_stats_tools/effsize.py

[6] This is shown both in the annotation guidelines as discussed in section 3, and the ratings itself, as annotator's may have hesitated to rate the speaker 1 (not confident). and preferred instead to use the label 3 (a little confident).

Table 3: Results of the Kruskal-Wallis H test, to compare the distributions of filler positions (by its token position in the transcript) compared to Ent_new/Ent_old positions, where "corrected" indicates the p-value after the Benjamini-Hochberg procedure. Note: Each cell indicates the number of reviews

	$p > .05$	$p \leq .05$
Ent_new	322	59
Ent_new corrected	381	0
Ent_old	477	70
Ent_old corrected	547	0

b/d using Table 2 for reference.

5 Results and Discussion

5.1 RQ1 How does a speaker's use of fillers relate to the incoming message from the speaker?

H1 Fillers are more likely to occur before the introduction of new information in the review.

Results for H1 are given in Table 3 for the Kruskal-Wallis H test, to compare the distributions of filler positions compared to 1. Ent_new and 2. Ent_old positions. By Kruskal-Wallis H test the distributions are significantly different for $\approx 15-20\%$ of the reviews (where $p \leq .05$). However, after utilising the Benjamini-Hochberg procedure for multiple testing correction, the distributions using this method do not significantly differ. This test is calculated using the sum of the ranks of each distribution. Given that the average review length is short ($\approx$ 256 tokens), and considering the close average median of fillers, Ent_new and Ent_old as given in Table 4, on reflection, this test may not capture nuances of the positional effects of fillers. We further discuss the limitations in section 7.

While significance testing focuses on a dichotomous result (i.e. significant versus not), we utilise Cliff's Delta δ to gain further insight into the magnitude of the effect. To interpret the results, Cliff's Delta δ ranges from -1 to 1, where 0 would indicate that the group distributions overlap completely; whereas values of -1 and 1 indicate a complete absence of overlap with the groups. For e.g. in H1 Ent_new, -1 indicates that all fillers in the review occur before new entities, and 1 indicates that all fillers in the review occur after new entities. This means that the smaller the effect size (close to zero) the larger the overlap, and the larger the effect size, the smaller the overlap.

By computing δ to estimate effect sizes as given in Figure 2, we see that for most reviews, fillers do occur visibly before Ent_new (median = -0.30 , $SD = 0.41$), but not before Ent_old (median = 0.20, $SD = 0.37$, given in Table 4), where the distributions of the δ values significantly differ ($Z = 27578.0$, $p < .05$ using Wilcoxon signed rank test). We see further evidence for this in Table 5[7], where majority of the reviews (565) have fillers occurring before Ent_new (sum of "nLarge" to "nSmall" δ sizes), compared to 163 reviews that had negligible effect size, and 139 reviews that had positive effect size (reviews that had fillers occurring after the introduction of new entities). We see the opposite δ effect sizes for Ent_old, where most of the reviews have fillers occurring after entities already introduced in the discourse (with predominately positive δ values as shown in Table 5), but not before. Fillers occurring after Ent_old is entirely plausible given that new entities can occur throughout the review, and not just at the start of one (as shown in Table 4, where the average median of Ent_new is roughly the same as Ent_old). Given the larger group with negligible effect size (247) for Ent_old, this does show that speakers may sometimes use fillers when repeating entities already introduced into the discourse. Dinkar et al. (2020a) used a language model (LM) trained on spontaneous speech to observe the probability of a filler appearing at a certain position; and found that the learnt word distribution shows that the LM places fillers predominantly at the start of sentences. However, sentence boundary annotation is dependent on the perspective of the transcriber, which in turn is certainly based on the presence of prosodic cues and fillers itself. Our findings suggest that there is more nuance to the way speakers utilise fillers (and indeed, our methodology is agnostic to sentence boundaries) in spontaneous speech. Therefore, regarding H1, stylistically speakers do tend to use fillers in the incoming message when introducing a new entity rather than one already introduced[8] (whether intentionally or not remains an open question), and the positions of fillers with

[7] The magnitude of Cliff's Delta δ can be interpreted by using the thresholds from Romano et al. (2006), i.e. $|\delta| < 0.147$ "negligible", $|\delta| < 0.33$ "small", $|\delta| < 0.474$ "medium", and otherwise "large".

[8] and indeed, this is the case for a dataset of spontaneous speech.

Table 4: Average median and SD for Ent_new, Ent_old (by first token position) and Fillers, and median and SD for effect size of the two δ distributions respectively.

	Avg. Median	Avg. SD
Ent_new	66.32	88.21
Ent_old	67.84	156.91
Fillers	66.05	125.95
δEnt_new	-0.30	0.41
δEnt_old	0.20	0.37

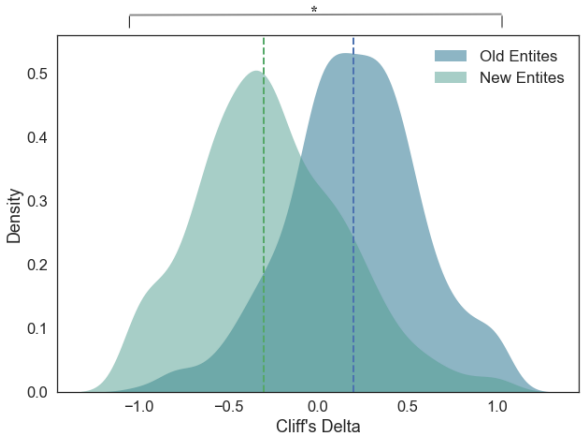


Figure 2: Distribution of Cliff's delta δ for fillers with Ent_new (New Entities) and fillers with Ent_old (Old Entites). Wilcoxon signed rank test has been performed to test whether the distributions significantly differ, with p $<$ .05 given by $*$. The dotted line denotes the median (given in Table 4).

respect to Ent_new significantly differ from positions of fillers with respect to Ent_old.

5.2 RQ2 How does the speaker's use of fillers relate to a listener's perception of the speaker and review?

H2 From H1, the speaker's use of fillers preceding new information contributes to the listener's perception of the speaker's confidence.

To investigate the presence of fillers occurring before new information among confidence ratings, we computed ORs. To interpret the results, when $OR = 1$, the presence of the percentage of fillers that occur before new entities (exposure) does not affect the odds of neither HC nor LC (i.e. no association of the expo-sure with outcome). When $OR > 1$, the presence of the exposure is associated with higher odds of HC (positive association). When $OR < 1$, the presence of the exposure is associated with higher odds of LC (positive association with decrease of HC).

Table 5: Counts of Cliff's delta δ for fillers with Ent_new and fillers with Ent_old for all reviews, where the "n" or "p" before each row value indicates negative or positive values respectively.

	Ent_new	Ent_old
nLarge	277	36
nMedium	142	36
nSmall	146	66
Negligible	163	247
pSmall	62	156
pMedium	35	138
pLarge	42	189

The results of the test show $OR = 0.72$ ($p <$ .001, 95% CI : 0.6-0.8)[9]. While $OR < 1$ in this case, indicating that the presence of fillers occurring before new entities gives a higher odds of LC, it is closer to 1, showing that the presence of the stimulus on the outcome is small. Interestingly, these findings are the opposite of what was hypothesised, which was that the speaker's use of fillers preceding new information contributes to the listener's perception of confidence; i.e. the more fillers are used in this way, the greater the odds of HC. According to the results of the *OR*s test, fillers occurring before new entities do not have a great effect on the odds of HC (only 28% lower given the presence of new entities) of the rating that the listener gives the speaker. This is consistent with the existing psycholinguistic literature on fillers as discussed in section 2. Arnold et al. (2004) for e.g. showed that fillers bias listeners towards new referents rather than ones already introduced into the discourse. In a study of the two fillers "um" and "uh" in American English, Tottie (2014) found that in natural conversation, listener's are not aware of the use of fillers, unless overused or used in the wrong context. Barr and Seyfeddinipur (2010) found that listener's expect the speaker to refer to something new following a filler (although they also found this to be specific to what was new for the speaker, and not only the listener), showing that listeners interpret fillers as delay signals, and infer plausible reasons for the delay by taking the speaker's perspective. While we cannot account for whether the annotator had rated the same speaker

[9]Risk Ratio $RR = 0.826$ with $p = .001$, 95% CI : 0.7-0.9

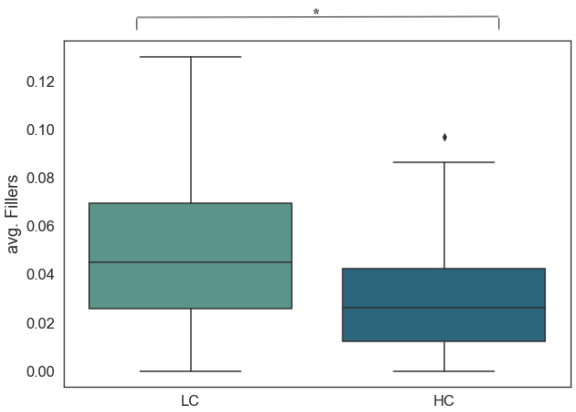


Figure 3: The speaker's average use of fillers (given by the percentage of fillers used compared to tokens in the review) with the categories of confidence using the divisions given in section 4 RQ2, $*$ denotes $p < .05$.

in multiple reviews, the annotator thus may expect the speaker to use fillers before new entities, or generally, before new expressions. This may not be considered usage in the "wrong context", and indeed, could simply indicate an increase in the number of entities in the review.

Looking at Figure 3, to show the average rate of fillers in the review (given by the percentage of fillers used compared to tokens in the review), it is clear that the use of fillers differs between HC and LC rated speakers (median filler rate of 0.026 and 0.045 respectively, with $U = 3873.0$ and $p < .05$ by Mann-Whitney U test). These results do not contradict Brennan and Williams (1995), i.e. there could be impressions formed by the listener about the speaker's expressed confidence based on fillers in spontaneous speech (as found in Dinkar et al. (2020a)). However, these results would suggest that the effect may not be from fillers used in the context of introducing new entities. This is an interesting finding; as fillers in these contexts may still have a metacognitive function as discussed above, but not necessarily related to FOAK. We cannot reject the null hypothesis, because there isn't sufficient evidence using our methodology to suggest that the occurrence of fillers before new entities has an effect on confidence (neither HC nor LC). Thus, these results suggest that fillers used in the context of introducing new entities in the discourse has little effect on the listener's rating of confidence that they attribute to the speaker.

6 Conclusion

The aim of this study was to empirically study on a real-life dataset, whether the utterance level use of fillers can help in understanding/ interpreting the perception of the speaker that was formed by a listener. We do so by studying a dataset of publicly available English monologue movie reviews, where the speakers voluntarily and naturally recorded themselves giving a movie review. Our findings show that speakers generally do tend to use fillers in the incoming message when introducing a new entity, rather than an entity already introduced into the discourse. Our results also suggest that the occurrence of fillers before new entities may not have an effect on the listener's perception of the speaker's expressed confidence, despite previous research to suggest otherwise (although these findings were validated in a different QA context). **Thus, local hesitations need not always lead to global impressions of uncertainty.** To the best of our knowledge, we are the first to contribute an in depth study of fillers accounting for hierarchical levels of analysis, i.e the sentence level and discourse level on real life data. In the *perspective taking account* of language comprehension as discussed in Barr and Seyfeddinipur (2010); the listener might be drawn to the mind of the speaker and infer possible reasons for delays in speech. Our analysis shows the possibility of different metacognitive functions in this perspective taking account that are brought about by the use of fillers on the listener. We hope that by using real-life data (reviews are available on ExpoTV.com, a social media platform where speakers can directly upload to an (unseen) audience videos of themselves giving an unbiased review), this study will both contribute to and encourage research on fillers in SLU.

7 Limitations

Our study is constrained to a dataset of monologues as mentioned in section 3. However, fillers can be used differently by the speaker (and consequently, processed differently by the listener) in dialogues. Furthermore, when considering the use of fillers, an important aspect is the acoustic information – as fillers are ubiquitous to spontaneous speech. While our measures focus on the transcripts and use ranking, it loses this temporal information, for e.g. distances in time, durations of fillers etc. However, it is difficult to calculate H1 in terms of time (rather than position), due to the poor results of the

forced alignment algorithms on this dataset. Since speaker's recorded themselves voluntarily and naturally using their own equipment, it is hardly surprising that the audio data is noisy. However, considering that SLU is often done on the output transcripts of ASR without considering acoustic information (except for the purposes of speech recognition), we consider these results as a preliminary analysis towards integrating fillers for SLU tasks.

References

Jennifer E Arnold, Michael K Tanenhaus, Rebecca J Altmann, and Maria Fagnano. 2004. The old and thee, uh, new: Disfluency and reference resolution. *Psychological science*, 15(9):578–582.

Dale J Barr and Mandana Seyfeddinipur. 2010. The role of fillers in listener attributions for speaker disfluency. *Language and Cognitive Processes*, 25(4):441–455.

Susan E Brennan and Maurice Williams. 1995. The feeling of another's knowing: Prosody and filled pauses as cues to listeners about the metacognitive states of speakers. *Journal of memory and language*, 34(3):383–398.

Herbert H Clark. 1996. *Using language*. Cambridge university press.

Herbert H. Clark and Jean E. Fox Tree. 2002. Using uh and um in Spontaneous Speaking. *Cognition*, 84(1):73 – 111.

Martin Corley, Lucy J MacGregor, and David I Donaldson. 2007. It's the way that you, er, say it: Hesitations in speech affect language comprehension. *Cognition*, 105(3):658–668.

Martin Corley and Oliver W Stewart. 2008. Hesitation disfluencies in spontaneous speech: The meaning of um. *Language and Linguistics Compass*, 2(4):589–602.

Tanvi Dinkar, Pierre Colombo, Matthieu Labeau, and Chloé Clavel. 2020a. The importance of fillers for text representations of speech transcripts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7985–7993, Online. Association for Computational Linguistics.

Tanvi Dinkar, Ioana Vasilescu, Catherine Pelachaud, and Chloé Clavel. 2020b. How confident are you? exploring the role of fillers in the automatic prediction of a speaker's confidence. In *Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8104–8108. IEEE.

John H Flavell. 1979. Metacognition and cognitive monitoring: A new area of cognitive–developmental inquiry. *American psychologist*, 34(10):906.

John J Godfrey, Edward C Holliman, and Jane McDaniel. 1992. SWITCHBOARD: Telephone Speech Corpus for Research and Development. In *Proceedings of the 1992 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages 517–520. IEEE.

Esther Le Grezause. 2017. *Um and Uh, and the expression of stance in conversational speech*. Ph.D. thesis.

Gina-Anne Levow, Valerie Freeman, Alena Hrynkevich, Mari Ostendorf, Richard Wright, Julian Chan, Yi Luan, and Trang Tran. 2014. Recognition of stance strength and polarity in spontaneous speech. In *2014 IEEE Spoken Language Technology Workshop (SLT)*, pages 236–241. IEEE.

Hannele Buffy Marie Nicholson. 2007. Disfluency in dialogue: attention, structure and function.

Sunghyun Park, Han Suk Shim, Moitreya Chatterjee, Kenji Sagae, and Louis-Philippe Morency. 2014. Computational Analysis of Persuasiveness in Social Multimedia: A Novel Dataset and Multimodal Prediction Approach. In *Proceedings of the 16th International Conference on Multimodal Interaction*, ICMI 2014, page 50–57, New York, NY, USA. Association for Computing Machinery.

Joseph P Pickett. 2018. *The American heritage dictionary of the English language*. Houghton Mifflin Harcourt.

Jeanine Romano, Jeffrey D Kromrey, Jesse Coraggio, and Jeff Skowronek. 2006. Appropriate statistics for ordinal level data: Should we really be using t-test and cohen's d for evaluating group differences on the nsse and other surveys? In *annual meeting of the Florida Association of Institutional Research*, volume 177.

Tobias Schrank and Barbara Schuppler. 2015. Automatic detection of uncertainty in spontaneous german dialogue. In *Sixteenth Annual Conference of the International Speech Communication Association*.

Elizabeth Shriberg. 2001. To 'errrr' is Human: Ecology and Acoustics of Speech Disfluencies. *Journal of the International Phonetic Association*, 31(1):153–169.

Vicki L Smith and Herbert H Clark. 1993. On the course of answering questions. *Journal of memory and language*, 32(1):25–38.

Marc Swerts. 1998. Filled Pauses as Markers of Discourse Structure. *Journal of Pragmatics*, 30(4):485 – 496.

Gunnel Tottie. 2014. On the use of uh and um in american english. *Functions of Language*, 21(1):6–29.

Gokhan Tur and Renato De Mori. 2011. *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons.

Ioana Vasilescu, Sophie Rosset, and Martine Adda-Decker. 2010. On the functions of the vocalic hesitation euh in interactive man-machine question answering dialogs in french. In *DiSS-LPSS Joint Workshop 2010*.

Charlotte Wollermann, Eva Lasarcyk, Ulrich Schade, and Bernhard Schröder. 2013. Disfluencies and Uncertainty Perception-Evidence from a Human-Machine Scenario. In *Sixth Workshop on Disfluency in Spontaneous Speech (DISS)*.

Vicky Zayats, Trang Tran, Richard Wright, Courtney Mansfield, and Mari Ostendorf. 2019. Disfluencies and human speech transcription errors. *arXiv preprint arXiv:1904.04398*.

Task2Dial: A Novel Task and Dataset for Commonsense-enhanced Task-based Dialogue Grounded in Documents

Carl Strathearn and Dimitra Gkatzia
Edinburgh Napier University
{c.strathearn,d.gkatzia}@napier.ac.uk

Abstract

This paper proposes a novel task on commonsense-enhanced task-based dialogue grounded in documents and describes the Task2Dial dataset, a novel dataset of document-grounded task-based dialogues, where an Information Giver (IG) provides instructions (by consulting a document) to an Information Follower (IF), so that the latter can successfully complete the task. In this unique setting, the IF can ask clarification questions which may not be grounded in the underlying document and require commonsense knowledge to be answered. The Task2Dial dataset poses new challenges: (1) its human reference texts show more lexical richness and variation than other document-grounded dialogue datasets; (2) generating from this set requires paraphrasing as instructional responses might have been modified from the underlying document; (3) requires commonsense knowledge, since questions might not necessarily be grounded in the document; (4) generating requires planning based on context, as task steps need to be provided in order. The Task2Dial dataset contains dialogues with an average 18.15 number of turns and 19.79 tokens per turn, as compared to 12.94 and 12 respectively in existing datasets. As such, learning from this dataset promises more natural, varied and less template-like system utterances.

1 Introduction

Goal and task oriented dialogue systems enable users to complete tasks, such as restaurant reservations and travel booking, through conversation (Chen et al., 2017). Traditionally, goal-oriented dialogue is based on domain-specific database schemas (Shah et al., 2018), however, encoding all domain information can be prohibitive since most domain knowledge exists in some unstructured format, such as documents (Feng et al., 2020), grounding dialogue in documents is a promising direction for several tasks. Here, we propose a new task for document-grounded dialogue, `Task2Dial`, which aims at generating instructions grounded in a document so that the receiver of the instructions can complete a task. This task requires following steps in a pre-specified order, invoking every day communication characteristics, such as asking for clarification, questions or advice, which might require the use of commonsense knowledge. The proposed task is different to existing document-grounded tasks such as CoQA (Reddy et al., 2019) in the sense that it goes beyond question answering grounded in a document, as answers might require commonsense knowledge and the underlying information might not be present in the document. At the same time, this challenging task aims to accommodate task-based dialogue, where the information follower has to comprehend (and confirm) all steps for completing the task.

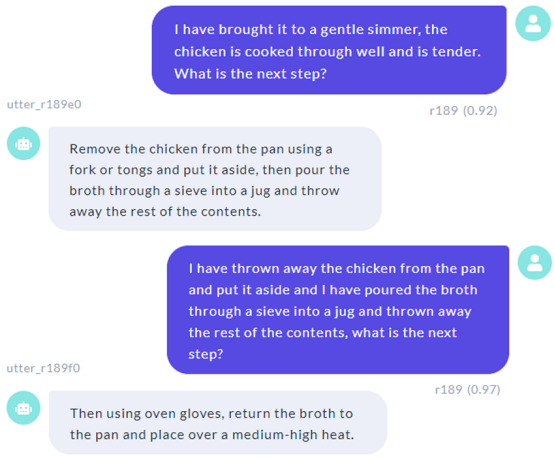


Figure 1: Excerpt from dialogue showing the commonsense handling of hot objects in the Task2Dial dataset.

Inspired by previous work on document-grounded dialogue (Feng et al., 2020; Hu et al., 2016; Stoyanchev and Piwek, 2010),

commonsense-enhanced natural language generation (NLG) (Lin et al., 2020; Clinciu et al., 2021), referring expressions generation (Panagiaris et al., 2021), concept acquisition (Gkatzia and Belvedere, 2021), and task-based/instructional dialogue (Gargett et al., 2010), we aim to capture two different types of knowledge: (1) document-level procedural context, i.e. what is the next step; (2) commonsense, i.e. answering questions that are not available in the document, as demonstrated in Figure 1. The task is designed as an instruction-following scenario with an information giver (IG) and an information follower (IF), inspired partly by the GIVE challenge (Gargett et al., 2010). The IG has access to the recipe and gives instructions to the IF. The IG might choose to omit irrelevant information, simplify the wording or provide it as is. The IF will either 'follow' the task by providing confirmation that they have understood the instruction or ask for further information. The IG might have to rely on information outside the given document, in other words the IG will rely on their common sense to enhance understanding and success of the task.

Task Description The proposed task considers the recipe-following scenario with an information giver (IG) and an information follower (IF), where the IG has access to the recipe and gives instructions to the IF. The IG might choose to omit irrelevant information, simplify the wording in the recipe or provide it as is. The IF will either follow the task or ask for further information. The IG might have to rely on information outside the given document (i.e. commonsense) to enhance understanding and success of the task. In addition, the IG decides on how to present the recipe steps, i.e. split them into sub-steps or merge them together, often diverting from the original number of recipe steps. The task is regarded successful when the IG has successfully followed/understood the recipe. Hence, other dialogue-focused metrics, such as the number of turns, are not appropriate here. Formally, *Task2Dial* can be defined as follows: Given a recipe R_i from $R = R_1, R_2, R_3, ..., R_n$, an ontology or ontologies $O_i = O_1 1, O_2, ..., O_n$ of cooking related concepts, a history of the conversation h, predict the response r of the IG.

This paper follows a theoretical framework which combines a background literature review with the design, development and challenges of the Task2Dial dataset (§2). The proceeding sections cover the data curation methodology (§3), present an analysis of the Task2Dial dataset and a comparison to related datasets (§4), discuss the related work (§5), and finally discuss the implications and challenges for the development of instruction-giving dialogue systems.

2 Theoretical Framework

The proposed task and associated dataset have connections to several lines of research in task and goal oriented dialogue, dialogue tracking and planning, document-grounded dialogue and commonsense reasoning. We next review related work in these areas while grounding our work.

2.1 Task and Goal-oriented dialogue

In dialogue management, task-oriented approaches focus on the successful completion of the individual stages of a task, towards achieving an end goal (Hosseini-Asl et al., 2020). Comparatively, goal-oriented approaches focus on comparing the outcome or overall performance against a gold standard (Ham et al., 2020). Task and goal oriented dialogue systems are common in domains such as booking and reservation systems for businesses (Zhang et al., 2020). However, business models are typically goal-oriented as the instructions are minimal and the focus is on the outcome (Ilievski et al., 2018). Instead, the Task2Dial task is formulated as a task-oriented dialogue paradigm to imitate real-world practical scenarios that can vary in complexity and require adaptability, additional information, clarification and natural conversation in order to enhance understanding and success.

2.2 Dialogue State Tracking and Planning

Task-based dialogue systems require the user and artificial agent to work synergistically by following and reciting instructions to achieve a goal. Zamanirad et al. (2020) define these methods in human-bot conversational models as:

- **Single intent and single turn policy:** relies solely on question and answer pairs assuming that the user provides all slot values in a single utterance. This type of task does not require dialogue state tracking.
- **Single intent and multi-turn policy:** Extends the previous conversational model, however this model can include multiple turns, to fill in missing information. Historic information is then extracted from all turns and used to structure data.

- **Multi-intent and multi-turn policy:** the intents can change depending on the context.

Instruction-giving scenarios follow the *multi-intent multi-turn* conversational framework, since they must accommodate knowledge and variability outside of a linear deterministic model as practical tasks can vary in complexity and the conversation can vary based on the interlocutors prior knowledge. In addition, there is no restriction on the amount of variability introduced into a task, such as introducing alternate methods, commonsense knowledge and concepts that change the structure and information within the dialogue. Variability is often reduced in human-machine scenarios as systems are limited in knowledge and their ability to respond to questions not seen in training (Shum et al., 2018), which can result in shortened responses and fewer questions asked on aspects of the task (Byrne et al., 2019). This hinders the system's ability to ensure that the IF has understood the IGs directions, which may produce irregular outcomes or result in an incomplete task. Therefore, capturing and emulating natural variability within the dialogue is crucial for creating robust and reliable conversational systems for instruction-giving scenarios.

Similarly to existing datasets such as Multi-Domain Wizard-of-Oz (MultiWOZ) (Budzianowski et al., 2018), Taskmaster-1 (Byrne et al., 2019), Doc2dial (Feng et al., 2020) and the Action-Based Conversations Dataset (ABCD) (Chen et al., 2021), Task2Dial also addresses the task of completing a process by following a sequence of steps. However, in addition to grounded information in documents, Task2Dial aims to accommodate questions and clarifications on different aspects of the task that might not be grounded in the document. In previous work, the user is limited to the path of the subroutine, however in Task2Dial, the IF can ask the IG questions at any stage of the task, regardless of the position within a given sequence and then return to that position after the question is fulfilled. For example, in a cooking scenario the IF may ask the IG how to use a certain kitchen utensil. The IG would need to answer this question, then return to the correct stage in the recipe in order to continue the sequence. This introduces additional challenges for state-tracking. The conversational agent must not only generate appropriate sequential instructions based on a document, it must also be able to request confirmation that the user has understood the task, and be able to answer questions outside its pre-defined script. Using document-grounded subroutines to capture intents that change the direction of a task broadens the interaction between the IG and IF (Chen et al., 2021), introducing new challenges for dialogue state-tracking.

2.3 Document-grounded dialogue

Document-grounded dialogue systems (DGDS) classify unstructured, semi-structured and structured information in documents to aid understanding human knowledge and interactions, creating greater naturalistic human-computer interactions (HCI) (Zhou et al., 2018). The aim of DGDS is to formulate a mode of conversation from the information (utterances, turns, context, clarification) provided in a document(s) (Ma et al., 2020). DGDS are particularly useful in task-oriented and goal-oriented scenarios as they emulate the natural dialogue flow between the IG and IF. A recent example of DGDS and closest to our work is Doc2Dial, a multi-domain DGDS dataset for goal-oriented dialogue modelled on hypothetical dialogue flows and dialogue scenes to simulate realistic interactions between a user and machine agent in information seeking settings (Feng et al., 2020). Here, we follow a similar setup, however in our proposed task, we further allow users to ask clarification questions, the answers to which are are not necessarily grounded in the document. This consideration is vital in the development of instruction giving conversational agents as the dialogue pipeline needs to be more flexible, as discussed earlier.

2.4 Commonsense-enhanced Dialogue

Commonsense reasoning is the innate understanding of our surroundings, situations and objects, which is essential for many AI applications (Ilievski et al., 2021). Simulating these perceptual processes in task and goal oriented DGDS generates greater context and grounding for more human-like comprehension. An example of commonsense dialogue in a practical task-based scenario is understanding the common storage locations of objects, or the safe handling and use of objects from their common attributes i.e. a handle, knob or grip. Commonsense dialogue is highly contextual: In Question Answering in Context (QuAC) (Choi et al., 2018), dialogues are constructed from Wikipedia articles interpreted by a teacher. A student is given

the title of the article and asks the teacher questions on the subject from prior knowledge, the teacher responds to the students' questions using the information in the document. This mode of question answering (Q&A) development is more naturalistic and grounded than previous methods as the challenges of understanding the information is ingrained in the dialogue from the underlying context. Similarly, the Conversational Question Answering Challenge (CoQA) dataset (Reddy et al., 2019) is formulated on a rationale, scenario and conversation topic, and the Q&As pairs are extracted from this data. This methodology is used in the Task2Dial dataset as it provides greater coreference and pragmatic reasoning within the dialogue for enhanced comprehension as shown in Figure 1.

In human-human IG/IF tasks, the IG may have prior knowledge of appropriate alternative methods, components and tools that can be used in a task that are not mentioned in the instructions. This information is vital if the IF has missing components or requires clarification on aspects of the task that are not clearly represented in the document. Variability is problematic to capture in DGDS alone as hypothetical scenarios in documents cannot account for all the potential issues in practice (Li et al., 2019). Thus, the ability to ask questions that are not available in the document is crucial when conducting real-world tasks due to the changeable conditions, complexity of the task and availability of components. This is particularly important in cooking tasks (as well as other instruction giving tasks) as the user may not have all the ingredients stated in a recipe, but may have access to alternative items that can be used instead. This approach can also be used in other domains such as maintenance or construction tasks if the user does not have a specific tool but has access to a suitable alternative tool without knowing it. This inevitably introduces new challenges for dialogue systems as commonsense-related intends and actions needs to be introduced in the dialogue system. Task2Dial moves away from the closed knowledge base/s in DGDS into incorporating multiple sources of information to broaden the adaptability and application of DGDS. This is achieved by developing additional resources that listed alternative ingredients to those mentioned in the metadata from the original recipes as well as instructions on how to use cookery tools. Appropriate alternative ingredients were collected and verified using certified online cooking resources that provide food alternatives.

3 Task2Dial

The `Task2Dial` dataset includes (1) a set of recipe documents; and (2) conversations between an IG and an IF, which are grounded in the associated recipe documents. Figure 2 presents sample utterances from a dialogue along with the associated recipe. It demonstrates some important features of the dataset, such as mentioning entities not present in the recipe document; re-composition of the original text to focus on the important steps; and the break down of the recipe into manageable and appropriate steps. Following recent efforts in the field to standardise NLG research (Gehrmann et al., 2021), we have made the dataset freely available[1].

3.1 Data Collection Methodology

The overall data collection methodology is shown in Figure 3 and is described in detail below.

Pilot Data Collection Prior to data collection, we performed three pilot studies. In the first, two participants assumed the roles of IG and IF respectively, where the IG had access to a recipe and provided recipe instructions to the IF (who did not have access to the recipe) over the phone, recording the session and then transcribing it. Next, we repeated the process with text-based dialogue through an online platform following a similar setup, however, the interaction was solely chat-based. The final study used *self-dialogue* (Byrne et al., 2019), where one member of the team wrote entire dialogues assuming both the IF and IG roles. We found that self-dialogue results were proximal to the results of two person studies. However, time and cost was higher for producing two person dialogues, with additional time needed for transcribing and correction, thus, we opted to use self-dialogue.

Creation of a recipe dataset Three open-source and creative commons licensed cookery websites[2] were identified for data extraction, which permit any use or non-commercial use of data for research purposes (Bień et al., 2020; Marin et al., 2019). As content submission to the cooking websites was unrestricted, data appropriateness was ratified by

[1] `www.huggingface.co/datasets/cstrathe435/Task2Dial`

[2] (a) `www.makebetterfood.com`
(b) `www.cookeatshare.com`
(c) `www.bbcgoodfood.com`

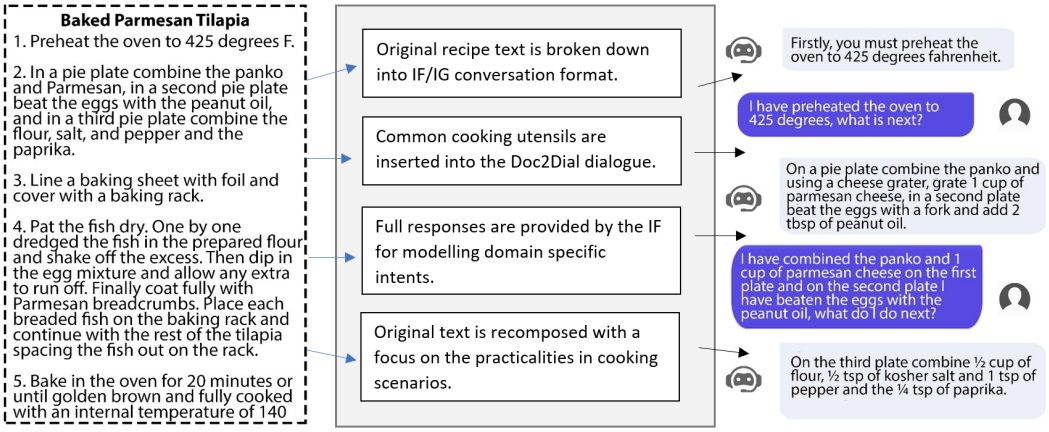


Figure 2: Original recipe text converted to Task2Dial dialogue

the ratings and reviews given to each recipe by the public, highly rated recipes with positive feedback were given preference over recipes with low scores and poor reviews (Wang and Kim, 2021). From this, a list of 353 recipes was compiled and divided amongst the annotators for the data collection. As mentioned earlier, annotators were asked to take on the roles of both IF and IG, rather than a multi-turn WoZ approach, to allow flexibility in the utterances. This approach allowed the annotators additional time to formulate detailed and concise responses.

Participants Research assistants (RAs) from the School of Computing were employed on temporary contracts to construct and format the dataset. After an initial meeting to discuss the job role and determine suitability, the RAs were asked to complete a paid trial, this was evaluated and further advice was given on how to write dialogues and format the data to ensure high quality. After the successful completion of the trial, the RAs were permitted to continue with the remainder of the data collection. To ensure high quality of the dataset, samples of the dialogues were often reviewed and further feedback was provided.

Instructions to annotators Each annotator was provided with a detailed list of instructions, an example dialogue and an IF/IG template (see Appendix A). The annotators were asked to read both the example dialogue and the original recipe to understand the text, context, composition, translation and annotation. The instructions included information handling and storage of data, text formatting, meta data and examples of high-quality and poor dialogues. An administrator was on hand throughout the data collection to support and guide the

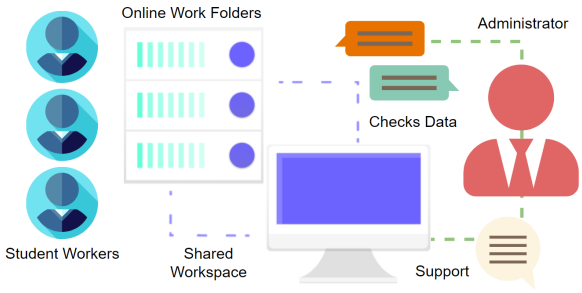


Figure 3: Overview of the Task2Dial Dataset Construction

annotators. This approach reduced the amount of low quality dialogues associated with large crowdsourcing platforms that are often discarded post evaluation, as demonstrated in the data collection of the Doc2Dial dataset (Feng et al., 2020).

Time Scale The data collection was scheduled over four weeks. This was to permit additional time for the annotators to conduct work and study outside of the project. Unlike crowdsourcing methods, the annotators were given the option to work on the project flexibly in their spare time and not commit to a specific work pattern or time schedule.

Ethics An ethics request was submitted for review by the board of ethics at our university. No personal or other data that may by used to identify an individual was collected in this study.

3.2 Task2Dial Long-form description

Unlike previous task and goal oriented DGDS, the Task2Dial corpus is unique as it is configured for practical IF/IG scenarios as demonstrated in Figure 2. Following (Bender and Friedman, 2018), we provide a long-form description of the Task2Dial cooking dataset here.

Curation Rationale Text selection was dependent on the quality of information provided in the existing recipes. Too little information and the transcription and interpretation of the text became diffused with missing or incorrect knowledge. Conversely, providing too much information in the text resulted in a lack of creativity and commonsense reasoning by the data curators. Thus, the goal of the curation was to identify text that contained all the relevant information to complete the cooking task (tools, ingredients, weights, timings, servings) but not in such detail that it subtracted from the creativity, commonsense and imagination of the annotators.

Language Variety The recipes selected for this dataset were either written in English or translated into English prior to data collection for ease of the annotators, language understanding and future training for language models. This made the dataset accessible to all contributors involved in the curation, support and administration framework.

Speaker Demographics The recipes are composed by people of different race / ethnicity, nationalities, socioeconomic status, abilities, age, gender and language with significant variation in pronunciations, structure, language and grammar. This provided the annotators with unique linguistic content for each recipe to interpret the data and configure the text into an IF/IG format. To help preserve sociolinguistic patterns in speech, the data curators retained the underlying language when paraphrasing, to intercede social and regional dialects with their own interpretation of the data to enhance lexical richness (Zampieri et al., 2020).

Annotator(s) Demographics Undergraduate research assistants were recruited through email. The participants were paid an hourly rate based on a university pay scale which is above the living wage and corresponds to the real living wage, following ethical guidelines for responsible innovation (Silberman et al., 2018). The annotation team was composed of two males and one female data curators, under the age of 25 of mixed ethnicity's with experience in AI and computing. This minimised the gender bias that is frequently observed in crowd sourcing platforms (Goodman et al., 2012).

Speech Situation The annotators were given equal workloads, although workloads were adjusted accordingly over time per annotator availability to maximise data collection. The linguistic modality of the dialogue is semi-structured, synchronous interactions as existing recipes were used to paraphrase the instructions for the IG. Following this, the IF responses where created spontaneously following the logical path of the recipe in the context of the task. The intended audience for the Task2Dial dataset is broad, catering for people of different ages and abilities. Thus, the dataset is written in plain English with no jargon or unnecessary commentary to maximise accessibility.

Text Characteristics The structural characteristics of the Task2Dial dataset is influenced by real-world cooking scenarios that provide genre, texture and structure to the dialogues. This provides two important classifications, utterances and intents that are universal for all task-based datasets and domain specific text that is only relevant for certain tasks. This data is used when training language models as non-domain specific sample utterances such as 'I have completed this step' can be used to speed up the development of future task-based DGDS.

Recording Quality As mentioned previously, the dialogues in Task2Dial are text-based.

4 Dataset Analysis

This section presents overall statistics of the Task2Dial dataset. We compare our dataset to the Doc2Dial dataset, although the latter focuses on a different domain. Employing research assistants to collect and annotate data rather than using crowdsourcing platforms meant that no dialogues were discounted from the dataset. However, a pre-evaluation check was performed on the dataset before statistical analysis to reduce spelling and grammatical errors that may affect the results of the lexical analysis.

Size Table 1 summarises the main descriptive statistics of Task2Dial and Doc2Dial. The dialogues in Task2Dial contain a significantly higher number of turns than Doc2Dial dialogues (18.15 as opposed to 12.94). In addition, Task2Dial utterances are significantly longer than in Doc2Dial, containing on average more than 7 tokens.

Lexical Richness & Variation We further report on the lexical richness and variation (Van Gijsel et al., 2005), following Novikova et al. (2017) and Perez-Beltrachini and Gardent (2017). We compute both Type-token ratio (TTR), i.e. the ratio of

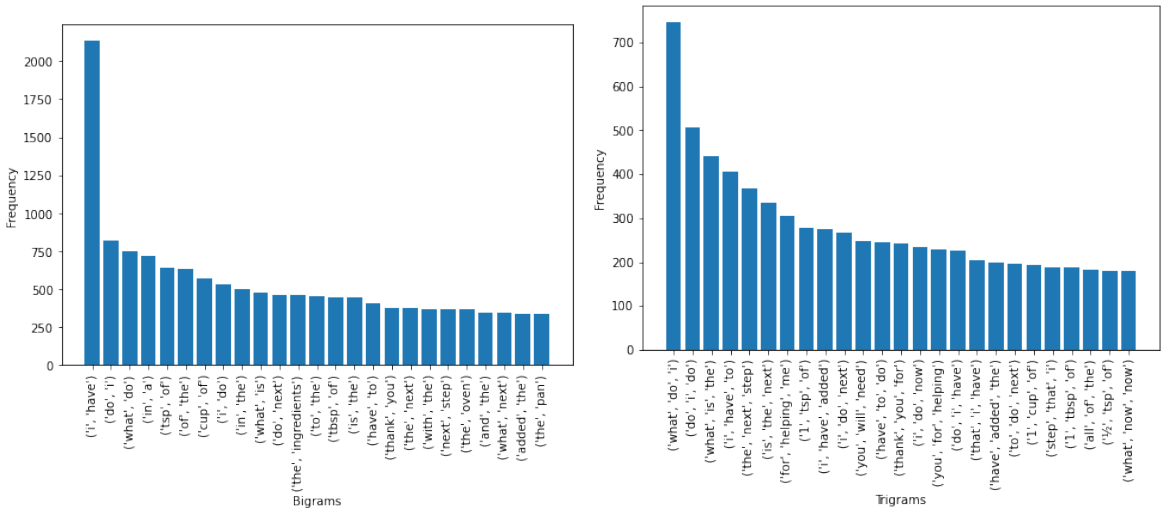


Figure 4: Distribution of the top 25 most frequent bigrams and trigrams in our dataset (left: most frequent bigrams, right: most frequent trigrams).

Dataset	#docs	#Turns	#Tkns/Turn	TTR	MSTTR
TASK2DIAL	353	**18.15**	**19.79**	**0.025**	0.84
DOC2DIAL	487	12.94	12	0.011	0.86

Table 1: Size and Lexical Richness of the dataset.

the number of word types to the number of words in a text, and the Mean segmental TTR (MSTTR), which is computed by dividing the corpus into successive segments of a given length and then calculating the average TTR of all segments to account for the fact the compared datasets are not of equal size[3]. All results are shown in Table 1. We further investigate the distribution of the top-25 most frequent bigrams and trigrams in our dataset as seen in Figure 4. The majority of both trigrams (75%) and bigrams (59%) is only used once in the dataset, which creates a challenge to efficiently train on this data. For comparison, in Doc2Dial's 54% of bigrams and 70% of trigrams are used only once. Infrequent words and phrases pose a challenge for the development of data-driven dialogue systems as handling out-of-vocabulary words is a bottleneck.

5 Related Work

This research considers the development of a DGDS for instruction-giving task-based dialogue. The work is inspired by previous research in DGDS: Doc2Dial (Ma et al., 2020) focuses on information seeking scenarios where the interaction between an assisting agent and a user is modelled as a a sequence of dialogue scenes. To enable document-grounded dialogue, each dialogue turn consists of a dialogue scene (dialogue act, a role such as user or agent and a piece of grounding content from a document). The sequence of dialogue scenes constitute the dialogue flow. DoQA (Campos et al., 2020) contains domain specific Q&A dialogues in three domains including cooking, where users can ask for recommendations/instructions regarding a specific task, although the task does not involve providing steps for completing a task as well. Finally, Task2Dial has drawn inspiration from crowd-sourced datasets such as MultiWoz (Budzianowski et al., 2018), Taskmaster-1 (Byrne et al., 2019) and ABCD (Chen et al., 2021) which demonstrate how DGDS can be configured in end-to-end pipelines for task-driven dialogue in virtual applications such as online booking systems. Commonsense enhanced dialogue datasets such as QuAC (Choi et al., 2018) and CoQA (Reddy et al., 2019) provided key information on infusing commonsense knowledge in dialogue and commonsense actions to instil greater human-like comprehension for artificial agents to operate more effectively in the real-world.

[3]TTR and MSTTR have been computed using https://github.com/LSYS/LexicalRichness.

6 Discussion & Conclusions

In this paper, we introduce the Task2Dial dataset of task-based document-grounded conversations with everyday speech characteristics, between an IG and IF during a cooking task. We further extend previous work in DGDS in order to emulate the unpredictability of human-human conversations in instruction giving that do not necessarily follow a tight schema of sequential instruction giving. Instead, other discourse and dialogue phenomena might take place such as clarification questions. We further considered the aforementioned challenges of modelling dialogue for instruction-giving tasks with a focus on state-tracking, task planning, and commonsense reasoning and proposed a new task and associated dataset.

Our proposed task aims to motivate research for modern dialogue systems that address the following challenges. Firstly, modern dialogue systems should be flexible and allow for "off-script" scenarios in order to emulate real-world phenomena, such as the ones present in human-human communication. This will require new ways of encoding user indents and new approaches to dialogue management in general. Secondly, as dialogue systems find different domain applications, the complexity of the dialogues might increase as well as the reliance of domain knowledge that can be encoded in structured or unstructured ways, such as documents, databases etc. Many applications, might require access to different domain knowledge sources in a course of a dialogue. Finally, as we design more complex dialogue systems, commonsense will play an essential part, with models required to perform reasoning with background commonsense knowledge, and generalise to tackle unseen concepts, similarly to (Lin et al., 2020). In the future, we aim to benchmark and evaluate a dialogue system based on the Task2Dial dataset and the Chefbot (Strathearn and Gkatzia, 2021), and extend this approach to a human-robot interaction (HRI) scenario.

Acknowledgements

The research is supported under the EPSRC projects CiViL (EP/T014598/1) and NLG for low-resource domains (EP/T024917/1).

References

Emily M. Bender and Batya Friedman. 2018. Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6:587–604.

Michał Bień, Michał Gilski, Martyna Maciejewska, Wojciech Taisner, Dawid Wisniewski, and Agnieszka Lawrynowicz. 2020. RecipeNLG: A cooking recipes dataset for semi-structured text generation. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 22–28, Dublin, Ireland. Association for Computational Linguistics.

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.

Bill Byrne, Karthik Krishnamoorthi, Chinnadhurai Sankar, Arvind Neelakantan, Daniel Duckworth, Semih Yavuz, Ben Goodrich, Amit Dubey, Andy Cedilnik, and Kyu-Young Kim. 2019. Taskmaster-1: Toward a realistic and diverse dialog dataset. *CoRR*, abs/1909.05358.

Jon Ander Campos, Arantxa Otegi, Aitor Soroa, Jan Deriu, Mark Cieliebak, and Eneko Agirre. 2020. Doqa – accessing domain-specific faqs via conversational qa.

Derek Chen, Howard Chen, Yi Yang, Alexander Lin, and Zhou Yu. 2021. Action-based conversations dataset: A corpus for building more in-depth task-oriented dialogue systems. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3002–3017, Online. Association for Computational Linguistics.

Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A survey on dialogue systems: Recent advances and new frontiers. *SIGKDD Explor. Newsl.*, 19(2):25–35.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wentau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. QuAC: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2174–2184, Brussels, Belgium. Association for Computational Linguistics.

Miruna-Adriana Clinciu, Dimitra Gkatzia, and Saad Mahamood. 2021. It's commonsense, isn't it? demystifying human evaluations in commonsense-enhanced NLG systems. In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 1–12, Online. Association for Computational Linguistics.

Song Feng, Hui Wan, Chulaka Gunasekara, Siva Patel, Sachindra Joshi, and Luis Lastras. 2020. doc2dial: A goal-oriented document-grounded dialogue dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8118–8128, Online. Association for Computational Linguistics.

Andrew Gargett, Konstantina Garoufi, Alexander Koller, and Kristina Striegnitz. 2010. The GIVE-2 corpus of giving instructions in virtual environments. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Sebastian Gehrmann, Tosin P. Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Aremu Anuoluwapo, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna-Adriana Clinciu, Dipanjan Das, Kaustubh D. Dhole, Wanyu Du, Esin Durmus, Ondrej Dusek, Chris Emezue, Varun Gangal, Cristina Garbacea, Tatsunori Hashimoto, Angelina McMillan-Major, Simon Mille, Emiel van Miltenburg, Moin Nadeem, Shashi Narayan, Vitaly Nikolaev, and Rubungo Andre Niyongabo. 2021. The GEM benchmark: Natural language generation, its evaluation and metrics. *CoRR*, abs/2102.01672.

Dimitra Gkatzia and Francesco Belvedere. 2021. "what's this?" comparing active learning strategies for concept acquisition in hri. New York, NY, USA. Association for Computing Machinery.

Joseph K. Goodman, Cynthia Cryder, and Amar Cheema. 2012. Data collection in a flat world: Strengths and weaknesses of mechanical turk samples. *Journal of Behavioral Decision Making, Forthcoming*.

Donghoon Ham, Jeong-Gwan Lee, Youngsoo Jang, and Kee-Eung Kim. 2020. End-to-end neural pipeline for goal-oriented dialogue systems using GPT-2. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 583–592, Online. Association for Computational Linguistics.

Ehsan Hosseini-Asl, Bryan McCann, Chien-Sheng Wu, Semih Yavuz, and Richard Socher. 2020. A simple language model for task-oriented dialogue. In *Advances in Neural Information Processing Systems*, volume 33, pages 20179–20191. Curran Associates, Inc.

Zhichao Hu, Michelle Dick, Chung-Ning Chang, Kevin Bowden, Michael Neff, Jean Fox Tree, and Marilyn Walker. 2016. A corpus of gesture-annotated dialogues for monologue-to-dialogue generation from personal narratives. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3447–3454, Portorož, Slovenia. European Language Resources Association (ELRA).

Filip Ilievski, Alessandro Oltramari, Kaixin Ma, Bin Zhang, Deborah L. McGuinness, and Pedro Szekely. 2021. Dimensions of commonsense knowledge.

Vladimir Ilievski, Claudiu Musat, Andreea Hossmann, and Michael Baeriswyl. 2018. Goal-oriented chatbot dialog management bootstrapping with transfer learning. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, page 4115–4121. AAAI Press.

Zekang Li, Cheng Niu, Fandong Meng, Yang Feng, Qian Li, and Jie Zhou. 2019. Incremental transformer with deliberation decoder for document grounded conversations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 12–21, Florence, Italy. Association for Computational Linguistics.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. CommonGen: A constrained text generation challenge for generative commonsense reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1823–1840, Online. Association for Computational Linguistics.

Longxuan Ma, Wei-Nan Zhang, Mingda Li, and Ting Liu. 2020. A survey of document grounded dialogue systems (DGDS). *CoRR*, abs/2004.13818.

Javier Marin, Aritro Biswas, Ferda Ofli, Nicholas Hynes, Amaia Salvador, Yusuf Aytar, Ingmar Weber, and Antonio Torralba. 2019. Recipe1m+: A dataset for learning cross-modal embeddings for cooking recipes and food images. *IEEE Trans. Pattern Anal. Mach. Intell.*

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2017. The E2E dataset: New challenges for end-to-end generation. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, pages 201–206, Saarbrücken, Germany. Association for Computational Linguistics.

Nikolaos Panagiaris, Emma Hart, and Dimitra Gkatzia. 2021. Generating unambiguous and diverse referring expressions. *Computer Speech Language*, 68:101184.

Laura Perez-Beltrachini and Claire Gardent. 2017. Analysing data-to-text generation benchmarks. In *Proceedings of the 10th International Conference on Natural Language Generation*, pages 238–242, Santiago de Compostela, Spain. Association for Computational Linguistics.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Pararth Shah, Dilek Hakkani-Tür, Bing Liu, and Gokhan Tür. 2018. Bootstrapping a neural conversational agent with dialogue self-play, crowdsourcing and on-line reinforcement learning. In *Proceedings*

of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers), pages 41–51, New Orleans - Louisiana. Association for Computational Linguistics.

Heung-Yeung Shum, Xiaodong He, and Di Li. 2018. From eliza to xiaoice: Challenges and opportunities with social chatbots.

M. S. Silberman, B. Tomlinson, R. LaPlante, J. Ross, L. Irani, and A. Zaldivar. 2018. Responsible research with crowds: Pay crowdworkers at least minimum wage. *Commun. ACM*, 61(3):39–41.

Svetlana Stoyanchev and Paul Piwek. 2010. Constructing the CODA corpus: A parallel corpus of monologues and expository dialogues. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Carl Strathearn and Dimitra Gkatzia. 2021. Chefbot: A novel framework for the generation of commonsense-enhanced responses for task-based dialogue systems. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 46–47, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Sofie Van Gijsel, Dirk Speelman, and Dirk Geeraerts. 2005. A variationist, corpus linguistic analysis of lexical richness. volume 1, pages 1–16.

Yiqi Wang and Jewoo Kim. 2021. Interconnectedness between online review valence, brand, and restaurant performance. *Journal of Hospitality and Tourism Management*, 48:138–145.

Shayan Zamanirad, Boualem Benatallah, Carlos Rodriguez, Mohammadali Yaghoubzadehfard, Sara Bouguelia, and Hayet Brabra. 2020. State machine based human-bot conversation model and services. In *Advanced Information Systems Engineering*, pages 199–214, Cham. Springer International Publishing.

Marcos Zampieri, Preslav Nakov, and Yves Scherrer. 2020. Natural language processing for similar languages, varieties, and dialects: A survey. *Natural Language Engineering*, 26(6):595–612.

Zheng Zhang, Ryuichi Takanobu, Minlie Huang, and Xiaoyan Zhu. 2020. Recent advances and challenges in task-oriented dialog system. *CoRR*, abs/2003.07490.

Kangyan Zhou, Shrimai Prabhumoye, and Alan W Black. 2018. A dataset for document grounded conversations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 708–713, Brussels, Belgium. Association for Computational Linguistics.

Domain and Task-Informed Sample Selection for Cross-Domain Target-based Sentiment Analysis

Kasturi Bhattacharjee[1], **Rashmi Gangadharaiah**[1], **Smaranda Muresan**[1, 2]

[1]AWS AI
[2]Columbia University
{kastb,rgangad,smaranm}@amazon.com

Abstract

A challenge for target-based sentiment analysis is that most datasets are domain-specific and thus building supervised models for a new target domain requires substantial annotation effort. Domain adaptation for this task has two dimensions: the nature of the targets (e.g., entity types, properties associated with entities, or arbitrary spans) and the opinion words used to describe the sentiment towards the target. We present a data sampling strategy informed by the difference between the target and source domains across these two dimensions (i.e., targets and opinion words) with the goal of selecting a small number of examples that would be hard to learn in the new target domain compared to the source domain, and thus good candidates for annotation. This obtains performance in the 86-100% range compared to the full supervised model using only ∼4-15% of the full training data.

1 Introduction

Target-based sentiment analysis aims to detect sentiments associated with specific targets in a given document. For instance, in Table 1, the targets ***service***, ***decor***, ***food***, ***portions*** have positive sentiment whereas ***operating system*** and ***kim kardashian*** have a negative sentiment. A key challenge for this task is that domain differences manifest themselves in terms of target types as well as the choice of opinion words used to express the sentiments towards those targets. Current datasets vary in their types of targets such as entities of various types (e.g., *Person*, *Location*, *Organization*, *Food*), predefined aspect/property categories (e.g., *quality* and *price*) or arbitrary spans that can denote an event ("The *opening night* was a success"). For instance, as shown in Table 1, for Restaurant reviews, one is likely to find target spans that are related to *food* (***food***, ***portions***), *ambience* (***decor***) or ***service***. Tweets might contain *celebrity* references (***kim kardashian***) as targets, while a Laptop review is likely to have references to *software* (***operating system***). Moreover, sentiment expressions vary from domain-to-domain as well. As shown in Table 1, we encounter sentiment expressions such as *delicious* for Restaurants domain, *older* for Laptops domain, and *famous* for Twitter that contains sentiment towards people.

Domain	Examples
Restaurants	The **service** is *excellent*, the **decor** is *great*, and the **food** is *delicious* and comes in *large* **portions**.
Laptops	I have had another Mac, but it got slow due to an *older* **operating system**.
Twitter	No, twitter, I don't want to follow **kim kardashian** - why is she *famous* btw or Chris Brown.

Table 1: Target spans (in **bold**) and sentiment expressions (*italicized*) from Restaurant review (Pontiki et al., 2016), Laptop review (Pontiki et al., 2014), and Twitter dataset (Dong et al., 2014).

Obtaining fine-grained sentiment annotations for specific spans of text is often time-consuming, expensive and requires domain expertise. Thus, we often encounter scenarios where we have labeled data from one or more domains (*source domains*) but none or very little labeled data from a new and *different* domain of interest (*target domain*). In this paper, we focus on a novel data sampling strategy for cross-domain target-based sentiment analysis that does not require sentiment labels but just the targets. It takes advantage of the two dimensions of domain differences for this task: targets and sentiment expressions. Our goal is complementary to work on transfer learning for domain adaptation for

this task (Rietzler et al., 2020).

Our proposed selection strategy aims to pick examples that are *informative* and *representative* of the target domain. To capture informativeness, a commonly used criteria in active learning settings (Settles and Craven, 2008; McCallum and Nigam, 1998), we use entropy-based sampling (Wang et al., 2017; Wang and Shang, 2014; Settles, 2009). This helps us sample examples that the model is most uncertain about in its sentiment predictions for given *targets*. Although entropy-based sampling is popular in active learning settings, to the best of our knowledge, it has not been applied to the task of sample selection for cross-domain targeted sentiment analysis. Further, we use Relative Salience (Mohammad, 2011) to pick examples containing *sentiment expressions* that are more *representative* of the target domain w.r.t the source domain. The efficacy of our data sampling strategy is tested by comparing the performance of the trained models on the sampled data against models trained on strong baselines such as entropy-based sampling (Section 3). Our proposed sampling strategy achieves performance in the 86-100% range compared to the full supervised model using only ∼4-15% of the full training data.

2 Datasets

We use three labeled datasets in English for target-based sentiment analysis that vary in domain - SemEval 2016 Task 5 (Pontiki et al., 2016) containing restaurant reviews **(R)**; SemEval 2014 Task 4 (Pontiki et al., 2014) containing laptop reviews **(L)** and a Twitter dataset **(T)** introduced by Dong et al., which contains tweets about celebrities (*Britney Spears*, *Lady Gaga*), products (*xbox*, *Windows 7*), and companies (*Google*). A *document* for R and L refers to a sentence of a review, with most documents containing a single target, and some containing multiple targets as well (30% of R-train, 38% of L-train). A tweet is a document for T, with each of them containing a single target. **R** and **T** contain Positive, Negative and Neutral sentiment labels for the target spans while **L** contains *Conflict* as a sentiment label. To maintain parity with **R** and **T**, we drop the conflict label from **L**. We retain the original train-test splits for all 3 datasets. Additionally, we sample 10% of the training data at random for a validation set.

Split	# Docs	# Pos, Neg, Neu spans
R-Train	1103	1107 397 61
R-Val	131	129 41 8
R-Test	420	468 114 30
L-Train	1320	884 786 434
L-Val	146	110 84 30
L-Test	411	341 128 169
T-Train	5588	1420 1392 2776
T-Val	659	141 168 350
T-Test	691	173 173 345

Table 2: Dataset stats. **R**=SemEval 2016 Restaurant Reviews, **L**=SemEval 2014 Laptop Reviews, **T**=Twitter. **Pos**=Positive, **Neg**=Negative and **Neu**=Neutral sentiments.

Setting	Highest RS scoring words
R→L	*easy*, *new*, *other*, *same*, *many*, *perfect*
L→R	*good*, *delicious*, *friendly*, *attentive*, *romantic*
L→T	*new*, *real*, *bad*, *last*, *famous*, *dead*

Table 3: Words with highest Relative Salience (RS) scores for each cross-domain setting.

3 Methodology

Entropy-based Sampling. In order to sample documents that contain hard-to-classify spans from the target domain, we use an uncertainty-based sampling method, that uses entropy (Shannon, 1948) to discover documents containing *targets* the model is uncertain about. Let D_s and D_t represent the training data for the *source* and *target* domains respectively. For each document in D_t, we predict the probability distribution over the 3 sentiment labels for *each target*, using a model trained on D_s, and compute the entropy per target prediction. The *average entropy* across all targets of the document indicates the *overall uncertainty* for the document. This aims to select documents based on informativeness.

Relative Salience (RS) based Sampling. We use Relative Salience (Mohammad, 2011) as a way to extract sentiment expressions that are more *representative* of the target domain when compared to the source domain. Based on the simplifying assumption that sentiment towards target spans are expressed through adjectives, we first extract all adjectives for each dataset using a Parts-of-Speech tagger. For each cross-domain experiment, we compute the RS of an adjective w as, $RS(w|D_s, D_t) = f_t/N_t - f_s/N_s$, where, f represents the frequency of occurrence of w in the training data, while N represents the total number of words in the training data. The subscripts s and

Setting	Sampling Strategy	Sample Documents Picked
L→R	Relative Salience	Be sure to try the seasonal, and always *delicious*, specials.
	Entropy	I had **Lobster Bisque** it has 2 oz. of Maine Lobster in it.
R→L	Relative Salience	I like how the Mac OS is so simple and *easy* to use.
	Entropy	pros: the macbook pro notebook has a large **battery life** and you wont have to worry to charge your laptop every five hours or so.
L→T	Relative Salience	"Sonny helped me grow, and become more aware of the media, and paparazzi, and the *famous* life. It makes me think twice." - demi lovato.
	Entropy	Gorbachev's 80th birthday was a huge success! among the guests were **arnold schwarzenegger** , Sharon Stone and Kevin Spacey. Exciting!

Table 4: Examples selected by RS-based and Entropy-based sampling for various cross-domain settings. *Italics* shows sentiment expressions used by RS, while **bold** shows the targets picked by the Entropy-based method.

t stand for *source* and *target* respectively. Note that labels are not considered for this, just the raw documents. Thus, RS score of a sentiment expression captures its importance in the target domain, w.r.t the source domain (see examples in Table 3). For each cross-domain scenario, we select documents from the target training set that contain any of the top 10 adjectives with the highest RS score.

RS+Entropy Sampling. Our proposed method of sampling involves selecting documents collected from both the Relative Salience and Entropy-based methods in different proportions for model training. Given the number of documents we wish to sample, the various combinations we experiment with include selecting 50%-50%, 30%-70% and 20%-80% from RS and entropy-based strategies, respectively. Depending on the combination, we first pick the top k documents ordered from highest to lowest entropy score, followed by the remaining number of documents picked from the RS set. In Table 4, we provide a few document samples picked by RS and Entropy. As expected, the RS method picks examples containing sentiment expressions that are more relevant to the target domain. With L (source) → R (target), we see sentiment expressions such as *friendly*, *delicious* and *romantic* that are more representative of the Restaurant domain (see Table 3). Meanwhile, the Entropy-based approach selects examples that the model is most uncertain about. For example, targets such as **Lobster bisque** are unlikely to be present in the Laptops domain and result in the model's uncertainty in predictions. A similar behavior is observed with R→L and L→T.

4 Model & Experimental Setup

The underlying model we use for target-based sentiment classification is a BERT model (Devlin et al., 2019). The model accepts as input the entire document and target spans with boundaries. The document is first encoded by BERT and span boundaries are used to pool tokens to form a span representation. Using span representation and the document as context, we perform multi-class classification to predict the sentiment for each span, by minimizing cross-entropy loss across sentiment labels.

Setting	Sampling Strategy	Pos F1	Neg F1	Neu F1
R→L	RS+Entropy	**85.03**	**72.30**	**52.08**
	Entropy	83.92	71.97	48.20
	Random	82.66	71.92	39.84
L→R	RS+Entropy	**94.34**	**77.64**	**28.00**
	Entropy	94.27	78.71	20.00
	Random	92.04	71.89	00.00
L→T	RS+Entropy	**58.39**	**62.91**	**71.37**
	Entropy	53.51	60.06	70.26
	Random	55.11	59.51	69.85

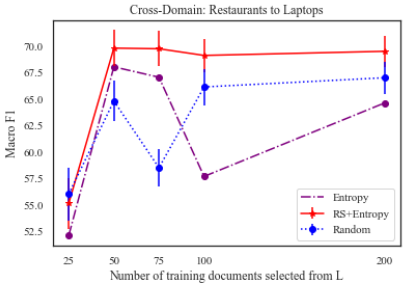
Table 5: F1 for each sentiment class obtained using various sampling strategies.

Experimental Setup & Baselines. SemEval datasets both consists of reviews in two different domains (restaurants and laptops). For our experiments, we explore both (**R→L**) and (**L→R**) as cross-domain settings. Further, we use the Twitter dataset that is different in genre to both L and R, and choose **L→T** as the cross-domain setting.

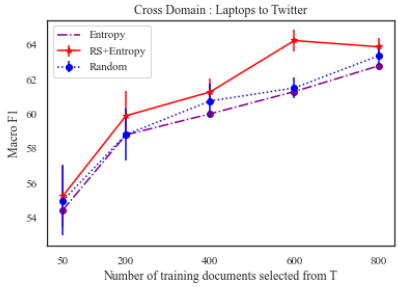
We first train the BERT model on *labeled* training data of the source domain. Documents from the target domain are then sampled using our proposed sampling method which is used to train the model. Model performance on target domain is reported using Macro F1. We experiment with a varying number n of sampled documents, starting with a small value (25 documents for Laptops and Restaurants, and 50 for Twitter) and going up to ∼15% of the training data for our experiments. Our baselines includes selecting a subset of n documents from the target domain at random as well as selecting the top n using entropy-based sampling only. For each experiment, we use the corresponding validation set for hyper-parameter optimization.

Setting	Samples	Entropy	RS+Entropy
R→L	**Price** was higher when purchased on MAC when compared to price showing on PC when I bought this product.	Neutral	Negative
L→R	Nice **ambience**, but highly overrated place.	Neutral	Positive
L→T	Quality night , amazing costumes but got ta say **lady gaga** was the best though.. poor gaga left shoes and phone in my car ha	Negative	Positive

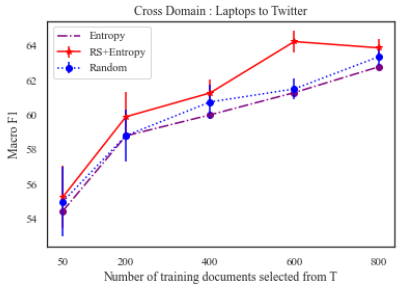
Table 6: **Targets** from test set that were *incorrectly* labeled by model trained using entropy-based sampled data, but were *correctly* predicted by model trained using the RS+Entropy sampled data.



(a)



(b)



(c)

Figure 1: F1 on the corresponding test sets (a) Laptops for R→L (b) Restaurants for L→R (c) Twitter for L→T.

5 Results

Figure 1 shows the mean Macro F1 scores (with standard deviation over 3 runs) for all three cross-domain settings with various sizes of sampled data. We find our proposed method to outperform both baselines for each cross-domain setting. In addition, Table 7 represents the amount of sampled data used by the model for training in these cross-domain settings and corresponding Macro F1 achieved as compared to a model trained with the full labeled training data. For R→L, we achieve 100% of Macro F1 as compared to the fully supervised case with only ∼4% of the training documents (4% of training instances). For L→T, we obtain 92.26% of the supervised setting with ∼11% of the training documents (∼11% of training instances). For L→R, our proposed method achieves within ∼86.68% of the fully supervised setting with ∼15% of the training documents (∼15% of training instances). Further, as shown in Table 5, RS+Entropy strategy outperforms both Entropy and Random baselines for each class, across all cross-domain settings.

Setting	% of Supervised Model Macro F1	%Train
R→L	100	∼4
L→T	92.26	∼11
L→R	86.68	∼15

Table 7: Comparison with fully supervised setting.

Error Analysis In Table 6, we show examples of targets for each cross-domain setting for which the model trained on Entropy-based sampled data makes errors in prediction, while model trained on RS+Entropy sampled data predicts correctly.

6 Conclusion

We propose a data sampling strategy for cross-domain target-based sentiment analysis that selects examples based on the two dimensions of domain differences for the task - targets and sentiment expressions. The proposed method combining Relative Salience and Entropy based sampling, when applied to three different cross-domain settings, is able to extract samples that are both informative and representative of the target domain. This helps the model achieve 86-100% of fully supervised performance using only 4-15% of the full training data, thus helping to reduce annotation cost. Further, it outperforms random and entropy-based baselines both in label-wise and overall model performance.

References

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.

Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. 2014. Adaptive recursive neural network for target-dependent Twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 49–54, Baltimore, Maryland. Association for Computational Linguistics.

A. McCallum and K. Nigam. 1998. Employing EM and Pool-Based Active Learning for Text Classification. In *ICML*.

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.

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, 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.

Alexander Rietzler, Sebastian Stabinger, Paul Opitz, and Stefan Engl. 2020. Adapt or get left behind: Domain adaptation through BERT language model finetuning for aspect-target sentiment classification. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4933–4941, Marseille, France. European Language Resources Association.

Burr Settles. 2009. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Burr Settles and Mark Craven. 2008. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 1070–1079, Honolulu, Hawaii. Association for Computational Linguistics.

C. E. Shannon. 1948. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423.

Dan Wang and Yi Shang. 2014. A new active labeling method for deep learning. In *2014 International Joint Conference on Neural Networks (IJCNN)*, pages 112–119.

Gaoang Wang, Jenq-Neng Hwang, Craig Rose, and Farron Wallace. 2017. Uncertainty sampling based active learning with diversity constraint by sparse selection. In *2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6.

Audio-Visual Recipe Guidance for Smart Kitchen Devices

Caroline Kendrick*
Technische Hochschule Ingolstadt
Research Institute AImotion Bavaria
Ingolstadt, Germany
cg_kendrick@outlook.com

Mariano Frohnmaier
Technische Hochschule Ingolstadt
Research Institute AImotion Bavaria
Ingolstadt, Germany
mariano.frohnmaier@thi.de

Munir Georges
Technische Hochschule Ingolstadt
Research Institute AImotion Bavaria
Ingolstadt, Germany
munir.georges@thi.de

Abstract

An important degree of accessibility, novelty, and ease of use is added to smart kitchen devices with the integration of multimodal interactions. We present the design and prototype implementation for one such interaction: guided cooking with a smart food processor, utilizing both voice and touch interface. The prototype's design is based on user research. A new speech corpus consisting of 2,793 user queries related to the guided cooking scenario was created. This annotated data set was used to train and test the neural-network-based natural language understanding (NLU) component. Our evaluation of this new in-domain NLU data set resulted in an intent detection accuracy of 97% with high reliability when tested. Our data and prototype (VoiceCookingAssistant, 2021) are open-sourced to enable further research in audio-visual interaction within the smart kitchen context.

1 Introduction

The importance of cooking in daily life makes the kitchen a natural focus for emerging technologies, as shown by Khot and Mueller in their analysis of human-food interaction (Khot and Mueller, 2019). Smart kitchen gadgets are part of this growing market (Research Private Ltd and Markets, 2020), including the all-in-one food processor, a countertop device which combines the expected blending utility with additional functionalities, such as weighing and cooking (Fries et al., 2018). This paper proposes a multimodal guided cooking experience for such a device, incorporating voice input and output, and touch interface.

The kitchen is a complex environment, and certain modes of interaction may be unavailable to a chef - for example, if their hands are oily, a touchscreen will be difficult to use. Offering multiple modes of interaction allows the user to interact with the device via the modality most natural to them in a context. Essential to natural interaction with any voice assistant is the Natural Language Understanding (NLU) component of the system, which greatly depends on both the quality of the model used and the amount of training data for the underlying domain. Although speech corpora for smart home and kitchen devices exist, to the best of our knowledge there is currently no public corpus with annotated voice commands for step-by-step guided cooking.

Our contributions are the design and implementation of a prototype for multimodal guided cooking with an all-in-one food processor. Moreover, we built an English-language NLU text corpus annotated with 39 intents and 19 entities for the guided cooking domain, and evaluated the quantity of training data. The prototype and data set are open source (VoiceCookingAssistant, 2021), and can be extended for further research in the area of smart cooking voice assistants.

2 Related Work

Existing guided recipe solutions require many sensors and 'smart' accessories to monitor a chef's progress. For example, the *Smart Kitchen* requires accessories such as radio frequency identification (RFID) tags to identify ingredients (Hashimoto et al., 2008), and *Shadow Cooking* requires a depth camera, projector, and digital scale (Sato et al., 2014). Both *KogniChef* (Neumann et al., 2017) and *Kochbot* (Alexandersson et al., 2015) make use of an entire smart kitchen system to monitor the user's actions and communicate recipe instructions, although the *Kochbot* may also be used alone as a mobile app providing recipe guidance via voice and screen (Schäfer et al., 2013). The research of Bouchard et. al (Bouchard et al., 2020) focuses

* See Acknowledgements on the very last page.

on the elderly and cognitively impaired, and their safety in the kitchen. They propose and prototype a smart range that monitors and guides cooking via a touch interface, but only monitors the stove, with limited recipes. These solutions and others successfully provide a multimodal approach to recipe guidance, and can also provide support for users with unique needs.

Blasco et al. (Blasco et al., 2014) discuss the importance of a voice interface in their smart kitchen proposal, which focuses on Assisted Living for the elderly and incorporates multiple kitchen and home appliances. They emphasize that voice commands are not only a natural interaction, but they also provide accessibility for users with visual or cognitive impairments. Although their research focuses on nutritional guidance, Angara et. al (Angara et al., 2017) outline another important benefit for conversational interaction in the kitchen: users prefer voice interaction when multi-tasking, and cooking is a complex and hands-on activity.

Our approach differs in that, for the majority of recipes, no other devices are needed to prepare or monitor a dish. Recipe guidance and cooking implements are contained in the same product, and the user must only interact with one device. It is also significant that, as recipe guidance is both visual and auditory, the chef does not *need* to look at the visual interface while cooking, and can focus on the meal being prepared.

3 Behind the Prototype and NLU Corpus

A detailed interaction flow (Section 3.1) supported by user interviews (Section 3.2) provided the basis for the prototype. The speech corpus (Section 3.3) was developed after the design & interaction flow were established. Section 3.4 briefly discusses the architecture and implementation of our prototype.

3.1 Interaction Flow for Guided Cooking

We began by evaluating guided cooking experiences, charting the potential interactions between the user and the system in an interaction flow. Examples of guided cooking experiences - such as instructional videos, product reviews, and recipe tutorials - were analyzed, from which we created an adaptive recipe framework, similar to that of Hashimoto et al. (Hashimoto et al., 2014). We found that recipe steps consist of the same or similar sub-steps, for which we created accompanying voice/touch interactions. A simplified example of

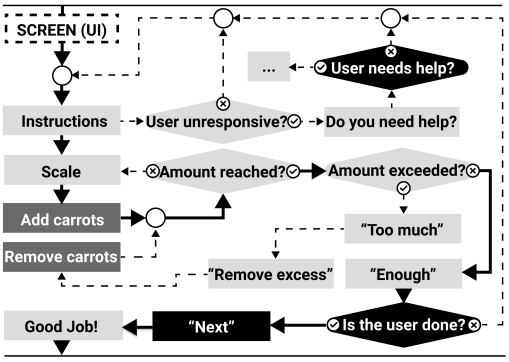


Figure 1: Excerpt from detailed interaction flowchart.

a recipe step may be seen in Figure 1. The diagram is based on Bollen's understanding of flow charts (Bollen, 2010).

This flow provides the basis of the prototype logic, affordances for the visual interface, and provides a general framework to adapt recipes for use with the food processor.

3.2 Visual & Voice Design based on User Studies

Two studies were conducted with potential users, which formed the basis of the NLU corpus and provided requirements for the visual design. Both qualitative and quantitative methods were employed, and the studies focused on identifying user needs and potential device interactions. The *first study* was a usability test which simulated a guided cooking scenario. While a participant used an all-in-one food processor to cook a recipe, the researcher dictated the recipe steps. The *second study* was a semi-structured interview, without using the food processor, in which 12 participants imagined a specific guided cooking scenario and were asked how they would give commands. For example, how would the participant increase the number of people a recipe would feed if they were preparing for a dinner party later that day?

Using affinity diagramming (Lucero, 2015) to analyze the interviews provided design insights, which informed the user expectations for the visual interface and user experience of the device. Figure 2 displays a sample of the visual interface, showing the overview of a selected recipe.

3.3 Generating the NLU Data

In order for our prototype to understand the user's intentions, we had to create a sufficiently large basis of annotated user queries (Allen, 1988). The language of the queries should sound as *natural* as

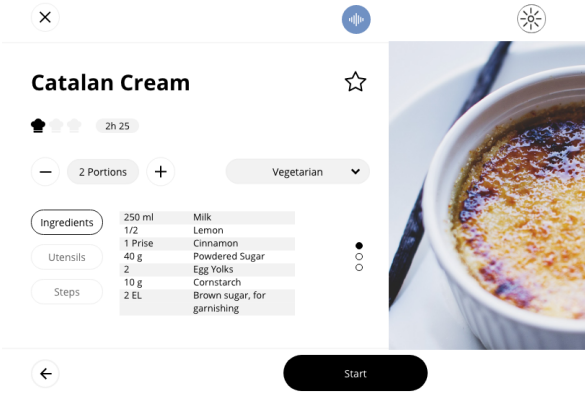


Figure 2: Visual Screen-Design: Recipe Overview.

possible. Therefore, we noted the specific language used by the participants from the second study in Section 3.2. The resulting 93 sample commands served as a starting point for several iterations to generate more commands and define the intents and entities.

Another condition of the definition of our domain-specific intents and entities was to define them in such a way that every interaction possible via the visual (or touch) interface can also be realized via speech. A detailed study of all of our prototype screens, like the one in Figure 2, and the interaction flow (see Figure 1) allowed us to fine-tune the definition of intents and entities. Following this strategy, we generated a set of 2803 text-based user queries with an in-domain vocabulary size of 436 and 15231 running words. Each sentence of this query base, denoted by $\mathcal{C}$, carries an intent label, but must not necessarily be labeled with an entity.

The resulting in-domain corpus $\mathcal{C}$ enables a user to navigate through step-by-step recipes, search for specific recipes, and add recipes to favourites. Also, it is possible to set device parameters like temperature, process duration, or blender speed. Overall there are 39 intents and 19 entities (VoiceCookingAssistant, 2021).

3.4 Architecture of the Prototype

As depicted in Figure 3, the prototype architecture is comprised of three components:

1. The **State Machine Frontend (SMF)** controls the user interface using high-fidelity graphics and a voice & touch interaction layer. It streams audio continuously to the
2. **Middleware Backend (MB)** which connects the SMF with the
3. **Logical Backend (LB)** using the MQTT protocol (Light, 2017). The LB processes the voice signal and provides the classified user intent in a JSON structure. (Pezoa et al., 2016).

Any component can be executed locally on the potential device without an internet connection as motivated by (Stemmer et al., 2017), or with partial internet access as proposed by (Georges et al., 2014). The heart of the LB uses *Rhasspy* (Hansen, 2021), an open-source collection of offline voice assistant services. It contains all subsystems necessary to processing a spoken query uttered by a user in a guided cooking scenario.

The prototype waits for a query that starts with a wake word. As soon as the wake word is recognized, the query is transcribed using the automatic speech recognition (ASR) system Kaldi (Povey et al., 2011). The recognized text is then forwarded to Rhasspy's intent recognition system in order to determine the user's intention. Here, the developer can choose the state-of-the-art NLU capabilities (Bocklisch et al., 2017).

4 Evaluation of the NLU Corpus

Rasa (Rasa Technologies, 2021) was used to evaluate our NLU dataset $\mathcal{C}$, as it can be selected in the Rhasspy pipeline. In addition, Rasa allowed us to easily use the **D**ual **I**ntent and **E**ntity **T**ransformer (DIET) architecture, a powerful state-of-the-art system for joint intent classification and entity recognition (Bunk et al., 2020).

4.1 First Analysis of the Speech Corpus

From the original NLU corpus $\mathcal{C}$ from Section 3.3, we carefully selected 839 sentences as our global test data set, denoted by $\mathcal{T}$, with 4507 running words. The remaining 1964 queries formed our training data set, denoted by $\mathcal{D} := \mathcal{C} \setminus \mathcal{T}$, with a total of 10724 running words. We repeatedly trained[1] the DIET model using the training data $\mathcal{D}$ and tested each model using the test set $\mathcal{T}$ ($n = 10$). The resulting averaged evaluation metrics (see Table 1) promise a good NLU performance. However, it is more interesting to know if we have collected *enough* user queries. The absolute numbers above do not allow us to make a statement about this.

[1]The configuration file which we used to specify Rasa's NLU training pipeline can be found on our GitHub repository.

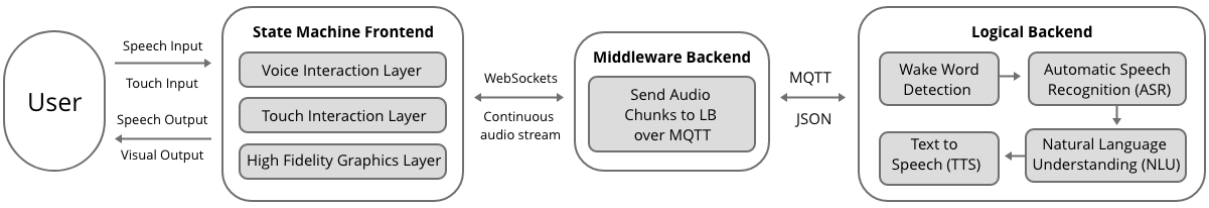


Figure 3: Architecture diagram for the high fidelity prototype. Touch input is processed by the same pipeline.

Table 1: Evaluating using 1964 user queries for training & 839 for testing, respectively.

Recognition	Precision	Recall	f1-score
Intent	0,975	0,973	0,971
Entity	0,948	0,958	0,944

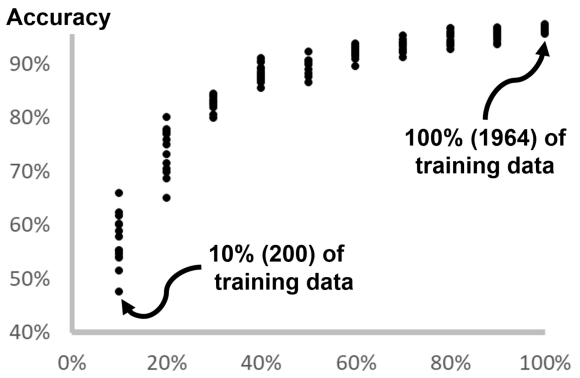


Figure 4: Evaluation using different amounts of training data $\mathcal{D}$.

4.2 Amount of Training Data vs. Intent Recognition Accuracy

We addressed this problem by running 200 experiments with different numbers of user queries $\mathcal{D}^{p_i}$ in each training phase. The process of successive enrichment of training data was simulated by starting with one-tenth of the original training data, denoted by $\mathcal{D}^{1/10}$, and repeatedly sampling about 200 new training examples until we collected all queries $\mathcal{D}$. This process yielded ten reduced training data sets $\mathcal{D}^{p_1} \subset \ldots \subset \mathcal{D}^{p_{10}} = \mathcal{D}$, where $p_i = i/10$, $i = 1, \ldots, 10$, denotes the ratio of the original amount of training data.

We simulated the above collection process 20 times, each time starting with a different randomly sampled set $\mathcal{D}^{1/10}$ resulting in different reduced training sets in each iteration. We used the same training parameters for the DIET models and the same test queries $\mathcal{T}$ as in Section 4.1. Evaluation started with ratio $p_1 = 1/10$ of the available training data and ended with the complete training data $\mathcal{D}$. Figure 4 shows the intent accuracy evaluation in more detail. When using 10% of the training data, the accuracy varied between 47% and 62% depending on the query selection. This means one may be lucky getting suitable queries, but a small number of user queries to train a NLU model is not reliable. The more training data is used, the higher the accuracy with decreasing dispersion.

5 Discussion and Future Work

This prototype is still a work in progress and was developed to test the recipe guidance, therefore it is not yet integrated into a device. The dataset is generic to multimodal assistants in the kitchen context, and may provide a basis for further research. To prove that the approach is generic to the kitchen context, similar work with emphasis on representative studies with larger data sets is needed. At the time of writing this paper, we recorded a subset of the NLU corpus in an anechoic chamber together with kitchen and device noises. The recordings will be published to enable academic and industrial research in this challenging speech domain.

6 Conclusion

The need for creating audio-visual interfaces is growing with the increasing availability of voice assistants in the home environment. Any new device is expected to provide a natural way of interaction. This short paper proposes a novel guided cooking experience, consisting of an audio-visual interface for a multi-functional food processor, in addition to an accompanying prototype with limited functionality.

Moreover, we built a speech corpus based on the proposed prototype design and user studies. Both the prototype and NLU dataset are freely accessible (VoiceCookingAssistant, 2021). This work provides a starting point for further research in the area of Natural and Spoken Language Understanding in the smart cooking domain.

Acknowledgements

This research was conducted as part of the User Experience Design Master Program at Technische Hochschule Ingolstadt. We would like to thank additional contributors **Malik Ali, Sadia Butt, Laura Forster, Nadine Kupitza, Viktoria Langeder, Alina Megos, Liia Mytareva, Subha Nair, Niklas Pachaly, Eunji Park, Daniel Peters, Andreas Riedel, Gülsüm Sanverdi**, and **Christian Sutter** for their work in designing and executing both research and prototype, as well as earlier versions of this paper. In addition, we thank **Manuel Kirschner** and **Tobias Hauser** for their support and domain expertise.

References

Jan Alexandersson, Ulrich Schäfer, Jochen Britz, Maurice Rekrut, Frederik Arnold, and Saskia Reifers. 2015. Kochbot in the intelligent kitchen–speech-enabled assistance and cooking control in a smart home. In *8. Deutscher AAL-Kongress (AAL)*, volume 8, pages 396–405.

James Allen. 1988. *Natural language understanding*. Benjamin-Cummings Publishing Co., Inc.

Prashanti Angara, Miguel Jiménez, Kirti Agarwal, Harshit Jain, Roshni Jain, Ulrike Stege, Sudhakar Ganti, Hausi A. Müller, and Joanna W. Ng. 2017. Foodie fooderson a conversational agent for the smart kitchen. In *Proceedings of the 27th Annual International Conference on Computer Science and Software Engineering*, page 247–253, USA.

Rubén Blasco, Álvaro Marco, Roberto Casas, Diego Cirujano, and Richard Picking. 2014. A smart kitchen for ambient assisted living. *Sensors*, 14(1):1629–1653.

Tom Bocklisch, Joey Faulkner, Nick Pawlowski, and Alan Nichol. 2017. Rasa: Open source language understanding and dialogue management.

P.W.L. Bollen. 2010. BPMN: A meta model for the happy path.

Bruno Bouchard, Kevin Bouchard, and Abdenour Bouzouane. 2020. A smart cooking device for assisting cognitively impaired users.

Tanja Bunk, Daksh Varshneya, Vladimir Vlasov, and Alan Nichol. 2020. Diet: Lightweight language understanding for dialogue systems.

Andreas Fries, Anne-Wiebke Bergmeister, and Marie Spindler. 2018. *Thermomix by Vorwerk – A New Way of Cooking*, pages 73–90. Springer Fachmedien Wiesbaden, Wiesbaden.

Munir Georges, Stephan Kanthak, and Dietrich Klakow. 2014. Accurate client-server based speech recognition keeping personal data on the client. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3271–3275, Florence, Italy. IEEE.

Michael Hansen. 2021. Rhasspy the docs. Accessed: 2021-08-12.

Atsushi Hashimoto, Jin Inoue, Takuya Funatomi, and Michihiko Minoh. 2014. How does user's access to object make hci smooth in recipe guidance? In *Cross-Cultural Design*, pages 150–161, Cham. Springer International Publishing.

Atsushi Hashimoto, Naoyuki Mori, Takuya Funatomi, Yoko Yamakata, Koh Kakusho, and Michihiko Minoh. 2008. Smart kitchen: A user centric cooking support system.

Rohit Ashok Khot and Florian Mueller. 2019. Human-food interaction. *Foundations and Trends® in Human–Computer Interaction*, 12(4):238–415.

Roger A Light. 2017. Mosquitto: server and client implementation of the mqtt protocol. *Journal of Open Source Software*, 2(13):265.

Andrés Lucero. 2015. Using affinity diagrams to evaluate interactive prototypes. In *Human-Computer Interaction – INTERACT 2015*, pages 231–248. Springer International Publishing, Cham.

Alexander Neumann, Christof Elbrechter, and Nadine et al. Pfeiffer-Leßmann. 2017. "kognichef": A cognitive cooking assistant. *KI - Künstliche Intelligenz*, 31(3):273–281.

Felipe Pezoa, Juan L. Reutter, Fernando Suarez, Martín Ugarte, and Domagoj Vrgoč. 2016. Foundations of json schema. In *Proceedings of the 25th International Conference on World Wide Web*.

Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al. 2011. The kaldi speech recognition toolkit. In *IEEE 2011 workshop on automatic speech recognition and understanding*.

Inc Rasa Technologies. 2021. Open source conversational ai. Accessed: 2021-08-11.

Markets Research Private Ltd and Markets. 2020. Smart home market. Accessed: 2021-06-21.

Ayaka Sato, Keita Watanabe, and Jun Rekimoto. 2014. Shadow cooking: situated guidance for a fluid cooking experience.

Ulrich Schäfer, Frederik Arnold, Simon Ostermann, and Saskia Reifers. 2013. Ingredients and recipe for a robust mobile speech-enabled cooking assistant for german. In *KI 2013: Advances in Artificial Intelligence*, volume 8077 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages 212–223. Springer.

G. Stemmer, Munir Georges, and J. Hofer et al. 2017. Speech recognition and understanding on hardware-accelerated dsp. In *INTERSPEECH*, pages 2036–2037, Stockholm, Sweden. ISCA.

VoiceCookingAssistant. 2021. Audio-visual-cooking-assistant.

Arabic Named Entity Recognition Using Transformer-based-CRF Model

Muhammad Al-Qurishi
Research Department / Elm Company
Riyadh, 12382-4182, Saudi Arabia

Riad Souissi
Research Department / Elm Company
Riyadh, 12382-4182, Saudi Arabia

Abstract

Named Entity Recognition is an essential component of Natural Language Progressing with countless practical applications. Several different directions of research are currently being pursued, exploring the possibilities in various ways. One particularly fruitful approach is the localization of multilingual deep learning tools based on the BERT architecture, with AraBERT and ARBERT/MARBERT serving as good examples. In this paper, we propose a simple but effective model for Arabic named entity recognition. The architecture of this model consists of three layers, as follows: a transformer-based language model layer, a fully connected layer, and the last layer is a conditional random field(CRF). Our proposed model shows promising results compared to the current state-of-art Arabic-NER models. For example, the F1-macro of the test data scores approximately 89.6% on the ANERCorp and 88.5% on the AQMAR datasets.

1 Introduction

Named Entity Recognition (NER) is an essential task that has numerous practical applications and enables successful performance of other NLP tasks. In the Arabic language, NER is additionally complicated by unique morphology and grammar, requiring specific methods to accurately label parts of text as entities from one of several possible classes.

Since Arabic is a widely spoken language with numerous dialects, there is an apparent need for automated linguistic tools that would serve the huge population of its speakers. At present time, the available Arabic NER tools are very scarce, only partially successful, and insufficiently tested, so new research in this field is urgently needed. Some of the notable NER tools for Arabic language based on machine learning include MADAMIRA by (Pasha et al., 2014), FARASA by (Abdelali et al., 2016), as well as the CAMeL tools suggested by (Obeid et al., 2020).

Discovering the most promising methodology that meets those requirements and allows for consistently accurate entity recognition would enable easier processing of the huge amount of information in Arabic that is already accumulated in various databases. This would in turn provide a boost in numerous research areas and practical applications of new technology, from semantic interpretation of multimedia content to searching for specific data in large data silos or the internet.

The idea of using automated algorithms for Natural Language Processing is several decades old, and has been successfully applied to many different problems. In particular, NER task has been the subject of many studies (Goyal et al., 2018; Li et al., 2020). Some of the earliest works of this kind are based on simple machine learning strategies, for example (Benajiba and Rosso, 2008; Oudah and Shaalan, 2012; Benajiba et al., 2010), while other proposed methods were based on grammatical rules and collections of named entities, i.e. (Khalil et al., 2020). More recently, models based on deep learning are emerging and achieving unprecedented success with a range of linguistic tasks including NER. In particular, BERT – Bidirectional Encoders Representations from Transformers model developed by (Devlin et al., 2018) has been extremely successful and inspired a number of regional variations that deal with specific languages. Some of the notable NER tools for Arabic language based on the BERT architecture include AraBERT and ARBERT/MARBERT serving as good examples.

In this paper we propose a simple but effective model for tagging Arabic named entities using transformer-based language model. Our solution architecture consists of three layers as follows. A transformer-based language model layer. In this layer we fine-tuned a pretrained language model Arabert which is the Arabic version of BERT. We also have fine-tuned and evaluated other models include AraElactra and XLM-Roberta. The second

layer is a fully connected linear layer which helps adjusting the output dimensions and initializing the inputs to the last conditional random field(CRF) layer. With CRF algorithm, tags are assigned based on the tag associated with the previous word in a linear fashion. Starting from the features extracted from the words, CRF estimates the probabilities that the word in question is a named entity or not. The problem is thus reduced to a simple probabilistic decision, while the inclusion of the previous word provides possibilities for capturing contextual clues. Due to its unique combination of simplicity and effectiveness, CRF is one of the most commonly used mathematical procedures used for NER tagging. Finally, we trained our model on two publicly available datasets- ANERCorp and AQMAR. The results of the conducted experiments were very promising and our proposed solution outperforms the current state-of-art in both datasets.

Our research paper organized as follows. We discuss the related studies in Section 2. In Section 3, we describe our solution in details. The experiments description is detailed in Section 4 and the results discussion is presented in Section 5. Finally, we conclude this work in Section 6.

2 Related Works

There are several different approaches to constructing NER tools, all of which are well represented in the modern literature and can be viewed as potentially viable in practice. The first group of solutions is based on simple machine learning strategies such as Support Vector Machines, Conditional Random Fields, or Random Forest, where the algorithm is trained on a labeled dataset and tasked with classifying new tokens based on statistical trends (Muhammad et al., 2020; Hudhud et al., 2021; Alshammari and Alanazi, 2020).

Deep learning methods employ a similar principle, but introduce vastly more complex architecture with far more sensitive capacity for capturing latent trends. Models based on bidirectional transformer stack architecture (such as BERT and its derivatives (Antoun et al., 2020; Abdul-Mageed et al., 2020)) have proven to be very promising, but a number of other designs including Pooled-GRU and CNN deserve to be examined. While deep learning systems are more powerful and accurate, they tend to have higher computational requirements and very long training times, which limits the extent of their practical usefulness (Helwe and Elbassuoni, 2019; Alkhatib and Shaalan, 2020; Al-Smadi et al., 2020).

Another group of solutions leverages certain grammatical rules to recognize named entities, for example using genitive rules to discover multi-word entities. It's also possible to construct knowledge bases that include gazetteers of named entities and semantic interpretation frameworks (ontologies) and use those resources to improve the recognition process (Elgamal et al., 2020; Khalil et al., 2020; Hudhud et al., 2021).

Since Arabic language presents additional challenges for NER algorithms due to its often ambiguous morphology and complex syntax rules, many works focused on this language introduce pre-processing operations and other modifications in order to optimize their tools to the nature of the task at hand, for example by standardizing the writing form for some words and removing diacritical marks (Pasha et al., 2014; Balla and Delany, 2020).

While most tools are trained with samples in Modern Standard Arabic, some of the works attempt to account for the diversity of local variations of spoken and written Arabic and improve performance with content that deviates from the formal language used in mainstream media and academic research. Incorporating data from social media (i.e. Twitter) is an ongoing research trend, contributing to diversification of training material and ultimately an increased capacity for recognizing named entities regardless of the form of writing. Typically, language-specific NER tools are expected to differentiate between several classes of entities, such as persons, organizations, and locations, although it is conceivable to have a much finer granularity and include a large number of classes. They should also be able to operate effectively with large quantities of unstructured data, which is characteristic for the modern online environment (Benali et al., 2021; Gridach, 2016; Zirikly and Diab, 2015).

All of the NER tools suggested in the reviewed works were empirically evaluated, and they typically displayed good overall performance during those tests. In general, the reported results are in the 85-95% range for accuracy and 60-83 for F1-macro score. They feature a limited amount of false positives and false negatives, with some fluctuation from one entity class to another. There is a trend that Arabic language tests yielded lower performance, mostly due to aforementioned complexities inherent in this language, but we have tried in this

paper to present an effective solution that can narrow this gap considerably. Since virtually no NER tools are completely error-free at this time, the differences between methods must be seen as the key to raising the expected level of performance. That's why even marginal improvements over state-of-the-art methods achieved through innovations are seen as very encouraging, providing some indications that with additional optimizations such innovations could lead to more tangible gains in the future.

3 Transformer-based-CRF Model

BERT and all of the derivative models feature an identical architecture which was directly inspired by Transformer (Vaswani et al., 2017). In all cases, there are two-layer stacks – decoder stack and encoder stack, each of which must contain one attention layer at the bottom. Self-attention mechanism in the decoder stack encodes the semantic relationships from the input sequence as attention scores and passes their normalized values to a series of forward-propagating layers. Conversely, in the encoder stack the representations are gradually refined with each new layer, until the correct output sequence can be produced. The number of layers and self-attention heads in the model can be variable, while the model is capable of processing semantic information in both directions, thus treating the entire sequence as a single connected unit.

This simple setup is now broadly accepted as the basis for advanced linguistic tasks, but it can be greatly improved through pre-training on certain supervised benchmark tasks, as well as fine-tuning of relevant model hyperparameters.

Our proposed architecture consists three layers, as shown in Figure 1. In the first layer, we fine-tuned three pretrained models, including AraBERT (Antoun et al., 2020), AraELECTRA (Antoun et al., 2021) and XLM-Roberta (Conneau et al., 2019) for Arabic Named Entity Recognition task.The second layer is a fully connected linear layer that receives the output of the pretrained model and reshapes it to be an input for the third layer. The final layer is the conditional random field (CRF), which is the tagging algorithm. The CRF makes sure that the objective of the model training is to return the most accurate combination of outgoing tags. We applied a dropout procedure in order to keep the training procedure well-balanced between entity classes.

3.1 Proposed Model Architecture

Figure 1 illustrates the proposed model architecture. If we have an input sentence x, it has to be tokenized before it is entered into the BERT model. In this process, the sentence x is padded to reach the maximum length of the sequence. Tokenized x with the corresponding attention mask is passed to the model, which outputs contextual embeddings of x. Transformer models including BERT use numerous separate attention mechanisms in each layer, in case of BERT base model, a total of 12 x 12 attention heads. In practice, this means that each token can be connected with 12 distinct features of any other token in the sequence.

There are two major aspects of BERT output crucial for accurate classification – prediction scores and hidden states. The prediction scores are obtained as the output of the last layer of the model, and thus represent the result of all attention heads in all layers and are relative to parameters such as batch size, hidden states size, and sequence length. Meanwhile, hidden states are the outputs of the individual layers of the model, and their total number is equal to the number of layers + 1 (12+1 for BERT). The output of one layer is immediately used as input for the next layer, which contextualizes its content further using its own attention heads. Thus, the prediction score basically represents a hidden state created by the final layer in the BERT model.

This output of the model can be understood in different ways. Intuitively, it seems logical that the last layer should contain all the information gained through different stages of learning, so its output should be seen as relevant regardless of the way the input vectors changed as they passed through the layers. On the other hand, it is quite possible that some of the vector modifications accidentally eliminated bits of useful information that could have contributed to and accurate prediction. To compensate for it, the input vectors can be partially or completely concatenated, or their sum could be used. We noticed that this procedure provides significant gains in terms of accuracy improvement, which is why we used this technique in our experiments.The model also includes a linear layer that helps reshape the output of BERT into 3 dimensions (batch size, number of tags, sequence length) and then pass it to a CRF layer tasked with making a prediction regarding the probabilities for each of the tags.

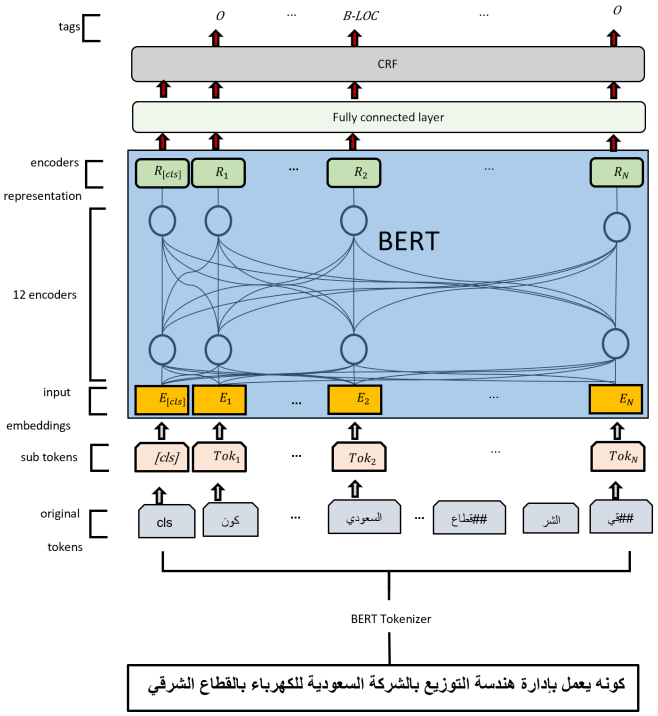


Figure 1: Basic layout of the model. Words are sent into a BERT-CRF model. Tokens are used to contextualize the word, to build the final representation

3.2 CRF Tagging Algorithm

Using transformer-based models, we get full distribution over the potential labels of the actual input possibilities. From this distribution, the classifier can predict the most likely class that the input could belong to. Named Entity Recognition meets this description, as the need to apply rules for semantic interpretation stretches it to a breaking point. That's because, I-PER can't match B-LOC under such conditions, effectively preventing the algorithm from maintaining its independence. Because of this, the selection of tags is performed using the conditional random field (CRF) method.

CRF method was originally publicized in the early 2000's (Lafferty et al., 2001), and has since been much discussed and broadly transferred to a number of different fields. This machine learning approach has proven to be useful in applications as diverse as genetic sequencing and linguistics. In recent years, models of this type are frequently used alongside LSTM-Long Short Term Memory networks to obtain the highest performance possible. This is especially noticeable in the natural language processing field, where this combination is increasingly becoming a standard due to its ability to raise the accuracy level for all tasks where tagging of token sequences is required. Classification of token sequences has the ultimate goal to determine the likelihood that a particular sequence belongs to the class Y from incoming data in vector form X. This section will present a simple type of conditional random field known as the linear chain conditional random field (Larochelle, 2021). We motivate the use of conditional random fields in the context of named entity classification where we want to be able to jointly model the full sequence of labels y associated with a sequence of inputs X as follows: For a given training set $\{(X, y)\}$, where $X = [x_1, ..., x_k]$ is a set of inputs and $y = [y_1, ..., y_k]$ is a target sequences, we calculate the conditional probability $p(y|X)$ as follows:

$$
\begin{aligned}
p(y|X) &= \prod_{i=1}^{k} p(y_i|x_i) \\
&= \frac{\prod_{i=1}^{k} \exp(E(x_i, y_i))}{Z(x_i)} \\
&= \frac{\sum_{i=1}^{k} \exp(E(x_i, y_i))}{\prod_{i=1}^{k} Z(x_i)}
\end{aligned}
\tag{1}
$$

Form equation 1, for a pair (X, y), the normal classification task can be resolved by calculating $P(y|X)$ in the way of multiplication of individual

probabilities for all tokens in the sequence up to the position i, where the value of i is greater than one but lower than the total length of the sequence k. Normalization with exponential function is used in this model, rather than the softmax function that is commonly deployed in the same role in most deep learning models.

Two central variables in this equation of the model are marked by E and Z, with the following definitions:

$E(x, y)$ denotes the emission score and represents the score assigned for the class y based on the vector x after i iterations. In other words, it represents the output of a BERT model after i steps are completed. While the input vector can contain practically any type of information, it is typically populated by a combination of nearby tokens, i.e. words or sentence representations. Each score is assigned its relative weight based on the results of training BERT model.

$Z(x)$ refers to the partition function, and can be viewed as a specific type of normalization function as it serves to discover a distribution of probabilities. All probabilities for different classes must always add up to 1. In this sense, it is a component of the softmax activation and can be calculated as follows.

$$Z(X) = \sum_{y_1^{'}} \sum_{y_2^{'}} ... \sum_{y_i^{'}} ... \sum_{y_k^{'}} \exp(\sum_{i=1}^{k} E(x_i, y_i^{'}) + \sum_{i=1}^{k-1} V(y_i^{'}, y_{i+1}^{'})) \quad (2)$$

Due to numerous loops in the model, calculating the value of $Z(X)$ is not simple. This requires considering all iterations of the input for each step in the model, necessitating k repetitions of all calculations to get the value for the entire set.

While the complexity of this procedure is very high $O(|y|^k)$, it's feasible to use the recurrent properties of the model and decrease the computational requirements. This can be accomplished with the forward/backward algorithm, which is capable of processing the sequence in either direction. Once this parameter is determined, the CRF implementation can be undertaken.

Above is described a standard model that calculates probabilities for each class, but we need to expand it by introducing ponders that can be adjusted through learning. Basically, this means the possibility of following up the label y_i with y_{i+1} can be quantified, linking the neighboring labels to each other, which is why this variation is named Conditional Random Field with a linear chain. All probabilities are thus factored with $P(y_{i+1}|y_i)$, using the exponential function to reformat this value as an emission score $E(x, y)$ expanded by the transition score $V(y_i, y_{i+1})$ as follows.

$$p(y|X) = \frac{\exp(\sum_{i=1}^{k} E(x_i, y_i) + \sum_{i=1}^{k-1} V(y_i, y_{i+1}))}{Z(X)} \quad (3)$$

Parameter $V(y_i, y_{i+1})$ is defined as a matrix consisting of elements obtained through learning while the model transitions from the position i in the sequence to the position $i + 1$ in the same sequence. on another words, it shows the chance that y_{i+1} succeeds after y_i.

3.3 Calculating the NLL Function

With every classification task, a crucial concern is to keep the errors to a minimum while the model is being trained with input data. A common way to achieve this goal is to use a loss function (L), and feed model predictions into it along with the accurate labels. This function has two possible outputs, 0 or a value grater than 0, depending on whether the two input values match or not. When probabilities $P(y|X)$ are calculated, it is obviously important to eliminate erroneous predictions. This problem can be solved by using the negative log value of the probability of error. This quantity is often referred to as the loss based on negative log probabilities, or NLL loss. It can be summarized with the formula $L = -log(P(y|X))$, and modulated with different types of log properties as follows.

$$\begin{aligned} &-log(P(y|X)) = \\ &-log(\frac{\exp(\sum_{i=1}^{k} E(x_i, y_i) + \sum_{i=1}^{k-1} V(y_i, y_{i+1}))}{Z(X)}) \\ &= log(Z(X)) - log(\exp(\sum_{i=1}^{k} E(x_i, y_i) \\ &+ \sum_{i=1}^{k-1} V(y_i, y_{i+1}))) \\ &= log(Z(X)) - (\sum_{i=1}^{k} E(x_i, y_i) + \sum_{i=1}^{k-1} V(y_i, y_{i+1})) \end{aligned} \quad (4)$$

The quantity $log(Z)$ denotes the logarithmic value calculated while the partition was per-

formed (Larochelle, 2021). This value is very useful for the implementation of the forward algorithm. NLL loss represents the forward pass, and can be calculated by reversing the sign before a normal value of log probabilities. Those probabilities can be obtained by calculating all the scores based on the partition and determining the difference between them. This procedure can be made significantly more efficient by the use of a mask matrix, which allows the model to skip any operations that refer to non-essential elements.

4 Experiment

In this section, we will evaluate the training techniques used to improve the algorithm, as well as its performance with different tasks and objectives, and the relationship between architecture of the proposed model and quality of the output.

4.1 Tagging Types

The ultimate objective of NER procedure is to associate a label belonging to a particular class to each included word. It's important to note that some complex named entities could stretch across multiple words, but are always contained in a single sentence. The predominant sentence representation form used in this field is IOB, with words that start a name of an entity marked with B, internally located words marked as I, and other tokens marked with O.

4.2 Data Samples

The model was evaluated using two Arabic public available datasets including ANERcorp and AQMAR. They are formulated specifically for the Arabic NER task.

4.2.1 ANERcorp Dataset

ANERcorp is a high-quality, annotated dataset containing Arabic language content gathered from a variety of publically available media sources. It was created in 2008 and has since been widely accepted as one of the standard datasets used for a variety of linguistic tasks. There are four classes of named entities included in this set, namely Persons, Locations, Organizations, and Miscellaneous. Based on those classes, the dataset includes a number of tags that pertain to named entities, including: *B-PERS*: Beginning of the name of a PERSon. *I-PERS*: Continuation (Inside element) of the name of a PERSon. *B-LOC*: Beginning of the name of a LOCation. *I-LOC*: Inside element present within the name of a LOCation. *B-ORG*: Beginning of the name of an ORGanization. *I-ORG*: Inside element present within the name of an ORGanization. *B-MISC*: Beginning of the name of an entity which doesn't belong to any of the previous classes (Miscellaneous). *I-MISC*: Inside element present within the name of a miscellaneous entity *O*: The word is not a named entity (Other).

Source	Ratio %
http://www.aljazeera.net	34.8
http://www.raya.com	15.5
http://ar.wikipedia.org	6.6
http://www.alalam.ma	5.4
http://www.ahram.eg.org	5.4
http://www.alittihad.ae	3.5
Other newspapers and magazines	17.8

Table 1: Overview of sources of articles container in ANERcorp dataset

There are a total of 316 articles within the ANERcorp dataset, all taken from journalistic sources and online publications. An overview of the content of the dataset regarding the sources, expressed in percentages as in Table 1.

In total, there are more than 150,000 tokens of more than 32,000 types within this dataset, with an average of 4.67 tokens per type. 11% of the content consists of proper names. In collaboration between the original creator of the corpus Yassine Benajiba and researchers from CAMeL Lab and Mind Lab, the dataset was slightly revised in 2020 to correct some imperfections and make it better suited for the type of studies it is needed for. Some of the corrections agreed upon involved spelling errors, blank Unicode characters and diacritical marks. At this time, the dataset was also divided into a training portion (125K words) and testing portion (25K words) to facilitate even better performance. In this study, the latest version of the ANERcorp corpus was used.

4.2.2 AQMAR Dataset

AQMAR (American and Qatari Modeling of Arabic) is a relatively small Arabic language dataset created specifically for the purpose of natural language processing evaluation, including named entity recognition (NER). It was created in collaboration between the US-based Carnegie Mellon University and Qatari governmental institutions, and is widely used in projects of various types.

This dataset consists of more than 3000

sentences taken from around 30 representative Wikipedia articles in Arabic, touching on a wide range of topics from history and science to politics and sports. This dataset was manually annotated with standard four NER classes – persons, locations, organizations, and miscellaneous, in addition to other many tags pertaining to sentiments, relations, etc. Total number of tokens contained in the AQMAR dataset is at 74,000, providing more than enough variability for NLP research purposes.

4.3 Fine-tuning Process

Our proposed model was trained and tested on both ANERCorp and AQMAR datasets, with the idea of fine-tuning the hyperparameters in every iteration. Hyperparameters are set up to create the best possible conditions for recognizing named entities in a labeled dataset based on the words found in the input sequence as shown in Table 2. When the dataset is used for model training and testing, individual sentences from unseen articles are selected at random and fed into the model as input. 80% of the dataset is typically used for training, 10% as a validation set, and the remaining 10% to evaluate the performance of the model on unseen examples.

Both AQMAR and ANERCorp datasets are very useful in model evaluation due to their versatility and the relevance of the content, making it a logical choice to include in this study. We have fine-tuned three pretrained models, including AraBERT, AraElectra, and XLM-Roberta. In the first experiment, we fine-tuned AraBert V2, the large model containing 24 layers of encoders stacked on top of each other, 16 self-attention heads, and a hidden size of 1024. In the second experiment, we used only the discriminator base model for AraElactra. This model has 12 attention heads, 12 hidden layers, and 768 hidden states size. In the third experiment, we used XLM-Roberta, a model with 12 attention heads, 12 hidden layers, and 768 hidden states.

During the three experiments we used an AdamW optimizer, which is useful when fine-tuning a pre-trained model as frozen layers. In all experiments, the learning ratio was $5e-5$, and the number of epochs was 5. The model input is a sequence of tokens that are processed to two vectors, input IDs, and attention masks using the BERT tokenizer. The tested sequence lengths were 128, 256, and 512 tokens. Dropout optimization was applied where a dropout step was introduced immediately before the data reaches the linear layer. We used the same dropout ratio of 0.1.

5 Results

In Table 3, the output of the proposed model is compared with state-of-the-art NER models for the Arabic language. Since the proposed solution doesn't use any sources other than the training sample, the performance of the competing methods was presented without access to such sources for the sake of fair evaluation. The AraBertv2 + CRF model achieved the most impressive result; the F1-macro score of almost 89.6% for this model concatenates the last 6 hidden layers. However, this model achieved these results on a sequence length of 256 tokens. Therefore, we applied the same model on a 512 token length, and we found that the best result is attained when we sum the 11 hidden layers; the F1-macro has not significantly change. In general, all the fine-tuned models outperform the state-of-the-art models, with significant improvements by almost 5% more than the best one, as we can see in Table 3. Tables 4- 6 show the experimental results of our proposed model with respect to the AQMAR dataset, using AraBert and AraElectra models, respectively.

We also tested our model on CANERCorpus a Classical Arabic Named Entity Recognition Corpus that was built by (Salah and Zakaria, 2018). This dataset was used by (Alsaaran and Alrabiah, 2021) and we compare our proposed model against theirs as shown in table 5. CANERCorpus was compiled starting from more than 7,000 hadiths - religious texts written by Islamic scholars that include mentions of a large number of named entities, totaling around 250 thousand words. There were approximately 13,000 named entities identified in the reviewed texts, and they were separated in 20 different classes based on the general category they are related to. In the pre-processing stage (Salah and Zakaria, 2018), the texts were segmented into sentences before they were annotated by three human operators, with majority opinion taken as valid in cases when there was disagreement. Words were annotated in the IOB2 format, which determines whether they are included in the beginning, middle, or the end of the entity name. This dataset is notable for very fine granulation with a lot of unique classes related to Islamic tradition (i.e. Allah, Prophet, Clan) in addition to standard NER classes such as person, time, or organization. For this reason, the dataset provides a strong foundation

Parameter	AraBERT	AraElectra	XLM-Roberta
Max sequence length	[128,256,512]	[128,256,512]	[128,256,512]
No. heads	16	12	12
No. hidden layers	24	12	12
Hidden layer size	1024	768	768
Batch size	4	16	4
Vocab size	64000	64000	250002
loss	Crf loss	Crf loss	Crf loss
Dropout prob	0.1	0.1	0.1
Optimizer	AdamW	AdamW	AdamW
Learning rate	5e-5	5e-5	5e-5
Number of epochs	5	5	5

Table 2: Table of hyperparameters for training the proposed Arabic NER model

Model	Seq	F1-Macro	Accuracy	Precision	Recall	F1-measure
AraBertv1	512	0.82767	0.971329	0.83902	0.816630	0.842
mBERT	512	0.76721	0.96293	0.79244	0.74354	0.784
gigaBERT	512	NA	0.969889	0.82820	0.812253	0.82015
AraBertv2	512	0.8043	0.97004	0.82900	0.81050	0.81965
Mawdoo3	512	NA	0.964331	0.79183	0.755798	0.77339
MARBERT	512	NA	0.966730	0.81267	0.774617	0.79318
ARBERT	512	NA	0.97224	0.84656	0.82582	0.83606
AraBertv2 + CRF (last)	512	0.88888	0.9916173	0.905367	0.912455	0.90889
AraBertv2 + CRF (concat 6)	256	**0.8956**	**0.9919526**	**0.913135**	**0.91378**	**0.91345**
AraBertv2 + CRF (concat 9)	512	0.8933	0.9916918	0.910310	0.913536	0.91192
AraBertv2 + CRF (sum 11)	512	**0.8946**	**0.9917663**	**0.908192**	**0.915954**	**0.91205**
AraElectra +CRF (concat 5)	512	0.872115	0.98755	0.9104595	0.894163	0.9022379
XLM-Roberta +CRF (concat 9)	256	0.875697	0.98968	0.90608	0.90181	0.90181

Table 3: The performance of the proposed Arabic NER model vs state of the art on ANERCorp

for testing Arabic language AI tools with sophisticated NLP capacities.

Thus, it's fair to say that stack architecture of the transformer-based models in combination for conditional random field currently represents one of the most successful NER methodologies that are independent from external sources. Its surprising performance can be explained by the strength of contextualization unique for this approach, which allows it to be accurate even when only limited information is available.

6 Conclusion

In some studies, scalability and versatility of the NER solution were considered alongside accuracy, reflecting the objective to create tools that could be used in practice without too many limitations. In this study, a simple architectural design of transformer-based model was presented, created specifically for tagging of word sequences in the context of Named Entity Recognition. This solution outperforms most of the state-of-the-art methods for this task, including those that rely on knowledge bases. A crucial element of the proposed solutions is their ability to track interdependencies between labels. This can be done with a CRF layer. Also, the process of summing and concatenating vectors has been shown to be effective in generating additional information that helps improve the tagging accuracy. Since vector representations are made on the words level, the model is able to collect contextual clues related to both syntax and morphology. The work in the future will be in two parts as follows. The first is to work on disambiguate the label when there is a word that expresses two different entities, such as the name of a place and a person at the same time. The second is to increase the number of named entity classes as well as working on our own dataset.

Model	#state	Seq	F1-Macro	Accuracy	Precision	Recall	F1-measure
last	1	128	0.862648	0.984333	0.8567	0.879594	0.8680425
concat	12	128	0.88176	0.985708	0.87530	0.895202	0.8851435
sum	8	128	0.88222	0.984944	0.864197	0.88832	0.876095
last	1	256	0.852139	0.983815	0.848448	0.878862	0.8633879
concat	6	256	0.875361	0.985300	0.866348	0.888616	0.8773413
sum	9	256	0.8772	0.98478	0.85322	0.8982412	0.875
last	1	512	0.86588	0.984261	0.8651	0.8809234	0.8729
concat	10	512	0.87350	0.98500	0.87350	0.8766467	0.875
sum	11	512	0.87789	0.985374	0.871121	0.8902439	0.88

Table 4: The performance of the proposed Arabic NER model using bert-large-arabertv2+crf test results on AQMAR dataset

Model	Seq	F1-Macro	Precision	Recall	F1-measure
(Alsaaran and Alrabiah, 2021)	54	NA	94.10	95.54	94.76
AraBertv2 + CRF (concat 6)	54	**91**	**98.11**	**98.42**	**98.26**

Table 5: The performance of the proposed Arabic NER model using bert-large-arabertv2+crf test results on CANERCorpus dataset

Acknowledgements

This work was supported by the Research Department in Elm Company under the Arabic language processing initiative.

References

Ahmed Abdelali, Kareem Darwish, Nadir Durrani, and Hamdy Mubarak. 2016. Farasa: A fast and furious segmenter for arabic. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Demonstrations*, pages 11–16.

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2020. Arbert & marbert: deep bidirectional transformers for arabic. *arXiv preprint arXiv:2101.01785*.

Mohammad Al-Smadi, Saad Al-Zboon, Yaser Jararweh, and Patrick Juola. 2020. Transfer learning for arabic named entity recognition with deep neural networks. *Ieee Access*, 8:37736–37745.

Manar Alkhatib and Khaled Shaalan. 2020. Boosting arabic named entity recognition transliteration with deep learning. In *The thirty-third international flairs conference*.

Norah Alsaaran and Maha Alrabiah. 2021. Classical arabic named entity recognition using variant deep neural network architectures and bert. *IEEE Access*, 9:91537–91547.

Nasser Alshammari and Saad Alanazi. 2020. The impact of using different annotation schemes on named entity recognition. *Egyptian Informatics Journal*.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2021. Araelectra: Pre-training text discriminators for arabic language understanding. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 191–195.

Husamelddin AM Balla and Sarah Jane Delany. 2020. Exploration of approaches to arabic named entity recognition. In *CLEOPATRA@ ESWC*, pages 2–16.

Yassine Benajiba and Paolo Rosso. 2008. Arabic named entity recognition using conditional random fields. In *Proc. of Workshop on HLT & NLP within the Arabic World, LREC*, volume 8, pages 143–153. Citeseer.

Yassine Benajiba, Imed Zitouni, Mona Diab, and Paolo Rosso. 2010. Arabic named entity recognition: using features extracted from noisy data. In *Proceedings of the ACL 2010 conference short papers*, pages 281–285.

Brahim Ait Benali, Soukaina Mihi, Ismail El Bazi, and Nabil Laachfoubi. 2021. New approach for arabic named entity recognition on social media based on feature selection using genetic algorithm. *International Journal of Electrical and Computer Engineering*, 11(2):1485.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised

Model	#states	Seq	F1-Macro	Accuracy	Precision	Recall	F1-measure
last	1	128	0.869587	0.978834	0.875871	0.86620689	0.871012482
concat	10	128	0.865022	0.97758	0.86192	0.8583333	0.860125261
sum	2	128	0.859811	0.978417	0.853556	0.8547486	0.854152128
last	1	256	0.862752	0.978032	0.864902	0.8758815	0.870357393
concat	2	256	0.875771	0.978344	0.87883	0.87638888	0.877607788
sum	3	256	0.861978	0.978552	0.864902	0.8601108	0.8625
Last	1	512	0.872761	0.979593	0.862116	0.8792613	0.870604781
concat	10	512	0.862209	0.977719	0.86908	0.85714285	0.863070539
Sum	2	512	0.8559	0.977719	0.866295	0.859116	0.862690707

Table 6: The performance of the proposed Arabic NER model using araelectra-base-discriminator +crf test results on AQMAR dataset

cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.

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*.

Marwa Elgamal, Mohamed Abou-Kreisha, Reda Abo Elezz, and Salwa Hamada. 2020. An ontology-based name entity recognition ner and nlp systems in arabic storytelling. *Al-Azhar Bulletin of Science*, 31(2-B):31–38.

Archana Goyal, Vishal Gupta, and Manish Kumar. 2018. Recent named entity recognition and classification techniques: a systematic review. *Computer Science Review*, 29:21–43.

Mourad Gridach. 2016. Character-aware neural networks for arabic named entity recognition for social media. In *Proceedings of the 6th workshop on South and Southeast Asian natural language processing (WSSANLP2016)*, pages 23–32.

Chadi Helwe and Shady Elbassuoni. 2019. Arabic named entity recognition via deep co-learning. *Artificial Intelligence Review*, 52(1):197–215.

Mohammad Hudhud, Hamed Abdelhaq, and Fadi Mohsen. 2021. Arabianer: A system to extract named entities from arabic content. In *ICAART (1)*, pages 489–497.

Hussein Khalil, Taha Osman, and Mohammed Miltan. 2020. Extracting arabic composite names using genitive principles of arabic grammar. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 19(4):1–16.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Hugo Larochelle. 2021. lectures on conditional random fields.

Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. 2020. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*.

Marwa Muhammad, Muhammad Rohaim, Alaa Hamouda, and Salah Abdel-Mageid. 2020. A comparison between conditional random field and structured support vector machine for arabic named entity recognition.

Ossama Obeid, Nasser Zalmout, Salam Khalifa, Dima Taji, Mai Oudah, Bashar Alhafni, Go Inoue, Fadhl Eryani, Alexander Erdmann, and Nizar Habash. 2020. Camel tools: An open source python toolkit for arabic natural language processing. In *Proceedings of the 12th language resources and evaluation conference*, pages 7022–7032.

Mai Oudah and Khaled Shaalan. 2012. A pipeline arabic named entity recognition using a hybrid approach. In *Proceedings of COLING 2012*, pages 2159–2176.

Arfath Pasha, Mohamed Al-Badrashiny, Mona T Diab, Ahmed El Kholy, Ramy Eskander, Nizar Habash, Manoj Pooleery, Owen Rambow, and Ryan Roth. 2014. Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of arabic. In *Lrec*, volume 14, pages 1094–1101. Citeseer.

Ramzi Esmail Salah and Lailatul Qadri Binti Zakaria. 2018. Building the classical arabic named entity recognition corpus (canercorpus). In *2018 Fourth International Conference on Information Retrieval and Knowledge Management (CAMP)*, pages 1–8. IEEE.

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.

Ayah Zirikly and Mona Diab. 2015. Named entity recognition for arabic social media. In *Proceedings of the 1st workshop on vector space modeling for natural language processing*, pages 176–185.

The Articulatory and acoustics Effects of Pharyngeal Consonants on Adjacent Vowels in Arabic Language

Fazia Karaoui, Amar Djeradi
LSCSP, FEI-USTHB
Algiers- Algeria
{fkaraoui, adjeradi}@usthb.dz

Yves Laprie
LORIA-CNRS
Nancy, France
yves.laprie@loria.fr

Abstract

This study uses physiological and acoustic data from a Moroccan Arabic male speaker to examine the articulations involved in the production of pharyngeal sounds and their effects on the adjacent vowels. It is known that vowel quality varies depending on the place or the manner of articulation of a preceding or a succeeding consonant; the coarticulation process affects considerably the production of phonemes in sequence. The present study shows that the larynx vertical movement gesture for pharyngeal consonants in Moroccan Arabic overlaps as coarticulation on adjacent vowels. The obtained results give us an overview of the larynx activity during the production of the three short vowels /a, i, u/ in an environment where the larynx rises relative to the rest position. The consequences of change in the vertical dimension of the pharynx due to the larynx raising on vowel quality are also explored. The coarticulatory effects will be assessed through the frequency modifications of F1, F2, F3, F4 and the variation in the distance F2-F1.

1 Introduction

One of the major difficulties in studying speech production is the problem of observing how speakers coordinate various articulatory movements, i.e. the aspects that connect physiology and speech production. In a continuous speech, a set of coordinative structures is involved to organize the execution of such stereotypes as the length of the vocal tract and laryngeal adjustment. The larynx is composed of several muscles responsible for his vertical movements also for fixing it to neighbouring organs (Marchal, 2010), which makes the adjustment of the laryngeal activity a complex task. It is situated under the hyoid bone and the tongue, it follows their movements. The Larynx height parameter is reported differently in different studies for different sounds, it contributes significantly to pharyngeal volume and sound quality in Arabic gutturals. Interestingly, a number of studies have observed substantial elevation of the larynx during the articulation of the Arabic pharyngeal consonants /ħ, ʕ/.The pharyngeal consonants have a primary place of articulation in the pharynx region of the vocal tract (McCarthy, 1994). The literature describes pharyngeal consonants as segments exerting fairly strong coarticulatory effects on their adjacent context. That is the reason that we deal with pharyngeal consonants in the present work, knowing that each regional dialects of Arabic have distinct coarticulatory and pronunciation patterns (Embarki, 2007; Ghazali, 2002), it is most appropriate to examine coarticulatory patterns within a contemporary vernacular Arabic dialect. This paper describes a physiological and acoustic study of the pharyngeal sounds of Moroccan Arabic (MA) and the articulatory and acoustic effects of these sounds on neighbouring vowels. The aims are to investigate the main articulatory attributes that correspond to the pharyngeal feature in MA, mainly the larynx activity. Measure the coarticulatory effects that pharyngeal consonants exert on adjacent vowels in the sequences, identify evidence for coarticulation principles of pharyngeal consonants and adjacent vowels. A number of previous studies have dealt with the acoustic consequences of the change in pharyngeal length on vowel production especially the lowering of the larynx (Marchal, 2010), but the lack of precision with respect to the vertical movement of the larynx has leads us to quantify the displacement of the larynx during vowel production in different contexts, especially in cases where the larynx rises considerably as in pharyngeal consonants production. In addition, the adjustment of the larynx height parameter in the articulatory model is essential for the production of vowels with good accuracy. No previous studies that have explicitly examined larynx activity associated with the vowels in the pharyngeal environment,

our study is a more comprehensive investigation that shed light on the larynx activity during the production of the three MA short vowels in pharyngeal neighbouring, including articulatory and acoustic components. In Arabic, the consonants transcribed as /ħ, ʕ/ have been conventionally categorized as pharyngeal, suggesting a place of articulation at the pharyngeal wall. Furthermore, the default laryngeal setting for these sounds is a raised larynx position. Two dimensions of movement anterior posterior and raising lowering of the larynx are adequate to label pharyngeal sound (Esling, 1996). A number of experimental studies have focused on the production of Arabic pharyngeal consonants in various dialects using physiological methods such as x-ray images or fiberscopes video monitoring (Laufer, 1988; Djeradi, 2006). However, a few studies have presented measurements bearing on the phonatory articulators' movements of Arabic back consonants and their influence on the adjacent vowels articulation. According to the lateral x-rays tracings of the vocal tract outlines for one Lebanese speaker (Delattre, 1971) and one Tunisian speaker (Ghazeli, 1977), two points of constriction were observed, one was at the hard palate (about 6 centimetre from the lips) while the other was located at the level of the epiglottis (about 3-4 centimetre above the glottis). Also, it was observed that the larynx was raised at about 9 mm relative to the default position (Ghazeli, 1977). The measurement of the elevation of the larynx during Arabic pharyngeal consonants in Alwan study is 0.7cm (Alwan, 1989). Knowing that the coarticulation process affects considerably the production of phonemes in sequence and the coarticulatory effects of neighboring segments on each other can operate in both directions. The intergestural timing relations exist between adjacent vowels, between vowels and preceding and following consonants, and between consonants in sequences. Byrd (1994, 1996) points out that these relationships allow gestures to overlap spatially and temporally resulting in an acoustic output which varies according to the behaviour of active gestures (Byrd, 1996, 1994). Thus the coarticulation has been viewed consistently as an expansion of traits. These are not only the result of the mechanical limitation of the articulators, they also depend on phonological constraints specific to the linguistic system, often attributed to the consonantal and vowel inventory of the system. The phonological specificities of the consonantal system explain the limitation of the transconsonant coarticulation. The motor and acoustic aspects of coarticulation are subjects of numerous studies, thus, many studies on the acoustic consequences of change in the vertical dimension of the pharynx have been carried out. The raising or lowering of the larynx alters the length of the pharyngeal cavity and this alteration plays an important part in determining voice quality (Marchal, 2010). McCarthy (1994) states that the main effect of pharyngeal consonant coarticulation that has been observed is an elevation of the first formant F1 by about 100 Hz, as measured in steady-state portions of an adjacent vowel(McCarthy, 1994), similar results were reported by (Al-Ani, 1970; Butcher, 1987; Zawaydeh, 1999). Alwan (1989) found with a synthesized speech that the primary perceptual correlate to coarticulation on a vowel for a pharyngeal segment was a high F1 and a low F2, she found that listeners prefer an (F2-F1) value that is lower for pharyngeal and the optimal pharyngeal candidate has an F2 close to F1. Stevens (1972) suggests that a high F1 and a low F2 of pharyngeal together form a strong spectral region, which contributes to the stability of these segments. Bin-Muqbil (2006) suggested that F2 values in vowels /i/, /a/ and /u/ after pharyngeal consonants are not significantly different than those after plain consonants in almost all cases(Bin-Muqbil, 2006). Other studies indicated some variation in F2 values after pharyngeal consonants(Al-Ani, 1970; Butcher, 1987; Zawaydeh, 1999; Ghazeli, 1977). Marchal (2010), stated that the net result of the larynx lowering is to bring F3 closer to F4 (Marchal, 2010). This phenomenon of the reduced distance between F3 and F4 has also been observed in the singing voice by Sundberg (2003), who considers this narrowing of the difference between the two upper formants as the principal characteristic of sung vowels when they are produced with a lowered larynx (Sundburg, 2003). In our study, the way of envisaging how the speaker coordinates his articulatory movement is to situate the process in a physiological theory of speech production. Thus, an articulatory model is built from cineradiographic images of a sagittal view of a Moroccan male speaker vocal tract, in order to approach the geometry of the speaker vocal tract. The present study focused on looking into issues related to the coarticulation process in order to adjust the larynx height parameter in articulatory models and evenly quantify the coarticulatory ef-

fects that pharyngeal consonants exerts on adjacent vowels in the sequences to identify evidence for coarticulation principles of pharyngeal consonants and adjacent vowels. Based on x-ray images of the vocal tract, we examined the vertical larynx movement, the hyoid bone location, the constriction opening and its location during the production of the MA pharyngeal /ħ, ʕ/ consonants, and we explored the behaviour of these articulations during the production of the adjacent vowels; the coarticulatory effect of these sounds on neighboring vowels in order to identify mainly the extreme larynx position during the production of the vowels in the pharyngeal environment.

2 Materials and Methods

The data used in this study were taken from a data base (DOnnées Cinéradiographiques VAlorisées et recherches sur la Coarticulation, Inversion et évaluation de Modéles
physiques) (DOCVACIM) (Sock, 2011). DOCVACIM database makes available a set of multilingual and multimedia data on speech production containing cineradiographic images of the vocal tract and acoustic signals. The Phonetics Institute of Strasbourg (IPS), the Grenoble Institute of Speech Communication and the LORIA in Nancy share some of the best x-ray movies on speech production that were made at the IPS, dealing with linguistic issues in languages spoken in Europe, in Africa, in Asia and in Latin America. The radiographic film used in this study consists of 2700 vocal tract x-ray images of a native Moroccan Arabic adult male speaker, and the acoustic data consists of 60 sentences in Moroccan Arabic dialect. For this study, we processed the images that correspond to the production of Arabic vowels adjacent to pharyngeal consonants. These articulatory data allowed us to extract combinations of articulators for this phonetic type. In the following section, we presented the tools allowing their analyses and their application in the context of speech production, and articulatory modelling approaches.

2.1 Treatment of Acoustic Data

The acoustic database consists of 60 sentences uttering by an adult male speaker in Moroccan Arabic dialect which is the L1 of the speaker. In this study we selected sequences that contain the pharyngeal consonants, we treated seventeen tokens of / ʕ/, eight tokens of /ħ/, and we have also treated sentences from the corpus containing vowels in pharyngeal neighboring; fifteen tokens of the three vowels distributed as follow: ten tokens of /a/, three tokens of /i/ and two tokens of /u/. On the other hand, we processed sequences containing vowels adjacent to plain coronal consonants: twenty-two tokens of /a/, twenty tokens of /u/ and thirteen tokens of /i/. We used the software Praat to segment the sentences into phonemes and also for the phonetic annotation (shown in Fig. 1), the length of each sentence is measured and also the duration of the entire sequence. After segmentation, we synchronized each phone with the corresponding x-ray images of the speaker vocal tract. Thus, for each phoneme, we have the corresponding vocal tract x-ray images, and then we processed the vocal tract x-ray images corresponding to each phoneme.

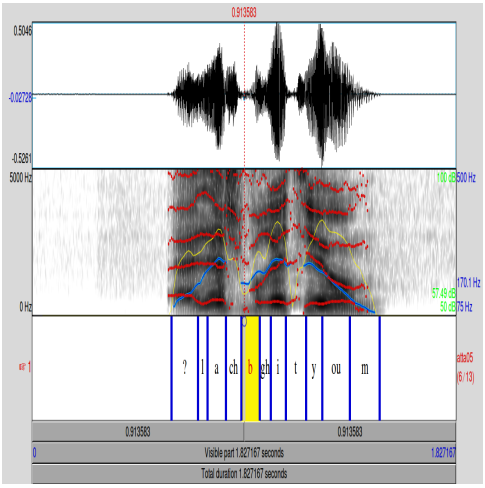


Figure 1: Segmentation of the phrase / ʕla ʃ/ /bɣit//yum/.

2.2 Treatment of X-Ray Images

X-ray images cannot be used directly, but manual, automatic or semi-automatic processing is necessary (Laprie, 2014). The contours of the articulators involved during speech production are tracked (tracking provides the displacement parameters, i.e. rotation and translation). The automatic tracking of bone structures is carried out, semiautomatic tracking for slightly overlapping organs, and finally manual delineation of the tongue. The contours are annotated and exploited for direct measurement of the articulators' displacement. In this work, we have drawn the main articulator contours, particularly those of the tongue, lower and upper lips, the larynx, the glottis, the jaw, the hard palate and the hyoid bone. Before the delineation of the phona-

tory organs contours a phonetic annotations and synchronization of the annotation is carried out via visual events which produce an acoustic event simultaneously. Prototypes of the processed vocal tract x-ray images are given in Figure 2.

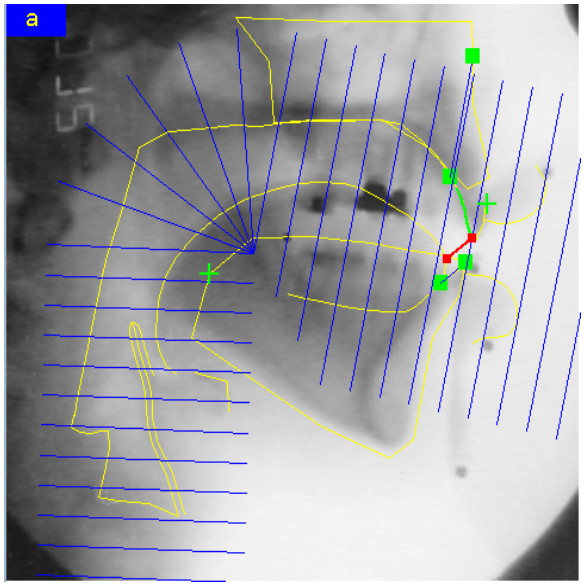


Figure 2: Prototype of the processed x-ray image (the grid, the articulators contours, the annotation in the top left of the image) during the production of the vowel /a/.

2.3 Measurement of the Displacement of the Phonatory Organs

The method of measurement of the displacement of the phonatory organs is based on an angular reference (semi-polar coordinates). Since the vocal tract configurations are different for each speaker, it is necessary to make an adapted grid for our subject. The measurement grid operates in relation to an orthonormal basis that is drawn beforehand on tracing paper (Bouarourou, 2014). This method is used in the analysis of numerous data collected on different subjects. The advantage of using this method is in the fact that it allows us to observe the maximum displacements: the larynx, the hyoid bone, the lips aperture and protrusion, the displacement of the tongue, the mandible, etc. In our study, we focused on the measurements of the larynx center position, the constriction opening, the constriction location and the hyoid bone position, knowing that these components are the effective gestures for producing the pharyngeal consonants as mentioned above.

Consonants	/ʕ/	/ħ/
HB raising (cm)	0.724 SD (0.297)	0.597 SD (0.283)
LC raising (cm)	0.872 SD (0.205)	1.644 SD (0.383)
co (cm)	1.157 SD (0.329)	1.078 SD (0.118)
cl (cm)	13.758 SD (0.209)	13.408 SD (0.349)

Table 1: The average values with (SD) regarding the measures of the elevation of the hyoid bone and the larynx center, the constriction opening and location during the production of the pharyngeal /ʕ/ and / ħ/

3 Results and Discussions

Table 1 summarizes the elevation of the hyoid bone (HB) and the larynx center (LC), the constriction opening (co) and the constriction location (cl)during the production of the pharyngeal /ħ, ʕ/. We focused on the vertical movement of the LC and the HB relative to the rest position, all measures are carried out relative to the reference point (upper incisor). The average values (AV) with standard deviation (SD) are calculated, the values are given in centimetre.

We noticed that the LC rises by 0.872 cm relative to the rest position during the production of the pharyngeal / ʕ/ in different contexts. The average value of the HB elevation is 0.724 cm relative to the rest position. The constriction location is at 13.758 cm. The constriction opening is 1.157 cm. Thus, the production of the voiced pharyngeal consonant / ʕ/ involves a considerable elevation of both the larynx center (0.872 cm) and the hyoid bone (0.724 cm) relative to the rest position. The obtained value of the larynx center elevation is close to that given in (Ghazeli, 1977) and Alwan (1986) (about 0.9 cm and 0.7 cm). For the voiceless pharyngeal /ħ/, the average value of the LC raising is 1.644 cm, and the SD is equal to 0.383. The elevation of the LC during the production of the voiceless pharyngeal /ħ/ is 0.772 cm more than that of the voiced / ʕ/. Thus the elevation of the LC is more important during the production of the voiceless /ħ/ compared to the obtained value for the production of the voiced / ʕ/. The average value of HB raising is 0.597 cm and the SD is equal to 0.283. The constriction location is at 13.408 cm, and the constriction opening

Vowels in the studied contexts	HB raising	LC raising
/a/ in /ħ/ neighboring	0.458	1.197
/a/ in /ʕ/ neighboring	0.894	0.835
/a/ in plain coronal	0.290	0.361
/i/ in /ħ/ neighboring	0.447	1.184
/i/ in /ʕ/ neighboring	0.141	0.667
/i/ in plain coronal	0.429	0.404
/u/ in /ʕ/ neighboring	1.159	1.152
/u/ in plain coronal	0.428	0.3

Table 2: The elevation of the hyoid bone and the larynx center during the production of the vowels /a, u, i/ in pharyngeal neighboring and in plain coronal contexts

is 1.078 cm, the obtained constriction opening is close to those given in (Sylak, 2013); the pharyngeal articulation was modeled with a length of constriction of 1.069 cm . From the obtained measurements, we stated that the Moroccan Arabic pharyngeal consonants are produced with a raised larynx and evenly a raised hyoid bone relative to the rest position and these results are in accordance with the results regarding the Tunisian Arabic and Iraqi Arabic and Kurdish (Al-Ani, 1970; Butcher, 1987; Zawaydeh, 1999; Ghazeli, 1977; Alwan, 1989; Al-Tamimi, 2009). In the second part of this section, we explore the articulatory effects of these consonants on adjacent vowels. First, the positions of LC and HB are measured during the production of the three vowels in plain coronal contexts (LCplc and HBplc), and then in pharyngeal contexts. During the production of the vowel /a/ (twenty two tokens of /a/ in plain coronal contexts), LCplc rises by 0.361 cm and the HBplc rises by 0.290 cm. For the vowel /u/ (twenty tokens of /u/ in plain coronal contexts in the corpus), the LCplc rises by 0.428cm and the HBplc rises by 0.3 cm. For the vowel /i/, the LCplc rises by 0.429 cm and the HBplc rises by 0.404 cm (for the thirteen tokens of /i/ adjacent to plain coronal contexts in the corpus). The elevation of the LC and HB during the production of the three vowels in plain coronal consonants neighboring does not exceed 0.429 cm relative to the rest position. In the other hand, in pharyngeal neighboring, we noticed that the elevation of the LC and the HB is more important as shown by the measurements given in Table 2.

For the vowel /a/ in / ʕ/ neighboring, the LC rises by 0.835 cm relative to the rest position, it rises more compared to LCplc (0.361 cm). The HB rises by 0.894 cm, it rises also more compared to HBplc (0.290 cm). In /ħ/ neighboring, the LC rises by 1.197 cm, it rises 0.82 cm more compared to LCplc. The HB rises by 0.458 cm more than HBplc. We noticed that LC and HB rise more in the voiceless /ħ/ neighboring compared to the voiced /ʕ/. For the vowel /i/, in /ħ/ neighboring, the LC rises by 1.184 cm relative to the rest position, it rises 0.76 cm more compared to LCplc. The HB rises by 0.447 cm, it is in the same range of displacement as in plain coronal contexts. In /ʕ/ neighboring, the LC rises by 0.667 cm, it rises by 0.24 cm more compared to LCplc. The HB rises by 0.141 cm (close to the rest position). From these measurements, we noticed that the elevation of the LC during the production of /i/ in the voiceless /ħ/ neighbouring is more important compared to the obtained LC in the context of the voiced /ʕ/. For the vowel /u/ in /ʕ/ neighboring, the LC rises by 1.152 cm relative to the rest position, it rises by 0.72 cm more compared to LCplc. The HB rises by 1.159 cm (0.86 cm more than HBplc). We conclude that the larynx vertical movement and the hyoid bone gestures for pharyngeal consonants in Moroccan Arabic overlaps as coarticulation on adjacent vowels; it exhibits a coarticulatory spread to adjacent vowel, and that consonants do not exert the same influence on adjacent vowels as shown by the obtained measurements. As mentioned earlier, the acoustic output varies according to the behavior of the active gestures and the obtained results from the articulatory study lead to additional questions about the acoustic consequences of change in the vertical movement of the larynx on vowels adjacent to pharyngeal consonants. Hence, an acoustic study of the vowel quality was conducted in order to determine the extent to which vowels could be affected by that environment. The software Praat is used for the segmentation of the words into phonemes and for the measurement of the formants values.

3.1 Acoustic Results

This part of the study will observe how vowel quality is affected by a pharyngeal consonants neighboring in Moroccan Arabic language. The Coarticulatory effects will be evaluated through the frequency modifications of the formants F1, F2, F3, F4 and the variation in the distance between the two first

formants F2-F1. The Table 3 summarizes the formants values of the three short vowels /a, u, i/ in the studied contexts.

The formants values of the vowel /a/ in plain coronal contexts are: the AV of F1 is 519.60 Hz with SD equal to 46.46, F2 is 1 573.437 Hz with SD =94.37, F3 is equal to 2 485.809 Hz, with SD equal to 137.43, the AV of F4 is 3 676.291 Hz with SD equal to 95.33. In /ħ/ neighboring, we noticed that the value of the first formant F1 increases by 223.76 Hz, F2 undergoes a moderate increase of 24.62 Hz, F3 decreases by -11.65 Hz and F4 increases by 148.54 Hz. In /ʕ/ neighboring, F1 increases by 134.79 Hz, F2 increases by 132.78 Hz, F3 undergoes a moderate increase of 53.87 Hz, and F4 increases by 91.89 Hz. The effect of the voiceless /ħ/ on F1 and F4 is more important than that of the voiced /ʕ/, F2 is more influenced by the voiced /ʕ/, F3 is less influenced by the pharyngeal contexts. We measure the coarticulatory effects that the studied pharyngeal consonants exert on adjacent vowel /a/, thus the distance between F2-F1 shows a significant decrease around 800 Hz compared to plain coronal contexts (1 000 Hz), the mean difference of F2-F1 in pharyngeal contexts and F2-F1 in plain coronal consonant contexts is 200 Hz, it reflects a considerable coarticulatory effect in /ħ/ context and weak in plain coronal contexts. The mean difference of F2-F1 in /ʕ/ contexts and F2-F1 in plain coronal contexts is 2.003 Hz, it reflects a coarticulatory effect which is exerted in a similar way in the two contexts. The voiceless pharyngeal /ħ/ has a considerable effect on the adjacent vowel /a/, in the voiced /ʕ/ neighboring the difference is weak compared to that measured in /ħ/ contexts. Concerning the vowel /i/ in plain coronals contexts, the AV of F1 is 378.77 Hz with SD equal to 49.5, the AV of F2 is 1 881.35 Hz, with SD =128.08, F3 is equal to 2 729.54 Hz, and SD is equal to 188.23, the AV of F4 is 3 681.63 Hz, with SD = 84.33. In /ħ/ neighboring, F1 increases by 214.32 Hz, both F2 and F3 undergo a slight variation, F4 increases by around 72 Hz. The mean difference between F2-F1 in /ħ/ contexts and F2-F1 in plain coronal contexts is 214.06 Hz, the coarticulatory effect is strong in /ħ/ contexts. In /ʕ/ neighboring, F1 increases by 181.83 Hz, F2 increases by 90.35 Hz, the mean difference of F2-F1 in /ʕ/ contexts and F2-F1 in plain consonants contexts is 91.44 Hz, the coarticulatory effect is weak compared to that measured in /ħ/. F3 increases by 86.88 Hz, and F4 increases by 305.56 Hz. Thus, the voiceless pharyngeal /ħ/ has a strong coarticulatoty effect on the vowel /i/ as for the vowel /a/. As regards the vowel /u/ in plain coronal contexts, the AV of F1 is 464.41 Hz with SD = 48.66, F2 is equal to 1 184.15 Hz, and SD equal to 178.35, the AV of F3 is 2 542.37Hz, with SD equal to 156.04, the AV of F4 is 3 607.9, and SD =103.82. In /ʕ/ neighboring F1 increases by 239.81 Hz, F2 decreases by -64.86 Hz, the distance F2-F1 decreases in pharyngeal contexts (around 400 Hz) compared to plain coronal contexts (around 700 Hz), the mean difference of F2-F1 is 300 Hz. F3 increases by 88.95 Hz, and F4 increases by 127.49 Hz. For the three vowels, the first formant F1 increased significantly over 200 Hz in the pharyngeal environment; it reaches 223 Hz for /a/, 239 Hz for /u/ and 214 Hz for /i/. From these results, we reported that the vowel height represented by the first formant F1 influenced greatly by the adjacent pharyngeal consonants for /a/ and /i/. This result converges with those given in (Al-Ani, 1970; Butcher, 1987; Ghazeli, 1977; Bin-Muqbil, 2006; Zawaydeh, 1999). The larynx position rises from 0.6 cm to 1.19 cm in pharyngeal neighboring, the F1 increases up to 200 Hz (25% to 43% for /a/, 48% to 60 % for /i/, and around 50% for /u/) in these environments. We noticed a correlation between the elevation of the larynx and the degree of the coarticulatory effects.

4 Conclusions

This study presents results corresponding to the production of the MA vowels in the contexts where the larynx rises considerably, and the fact that the larynx rises decreases the length of the pharynx cavity, this involves a shortening (or a contracting) of the entire vocal tract which can explain well the obtained results. In contrast, when the larynx is lowered, the results will be opposite to that given above because the lowering of the larynx has the physical effect of lengthening the vocal tract and obviously the acoustic effects is expected to be opposite to those given in our study as reported throughout Marchal's study (2010) regarding the consequences of change in the vertical dimension of the pharynx due to the larynx lowering. In this study we focused on the larynx and the Hyoid bone raising which are the main active gestures during the production of the studied pharyngeal consonants, the obtained results show a correlation between the elevation of the larynx and the degree

vowels	F1(Hz)	F2(Hz)	F3(Hz)	F4(Hz)
/a/ in /ħ/	743.36 SD(21.2)	1598.06 SD(241.87)	2474.15 SD (181.6)	3824.83 SD(127.6)
/a/ in /ʕ/	654.39 SD(40.42)	1706.22 SD(159.46)	2539.67 SD(66.43)	3768.18 SD(89.16)
/a/ in pcc	519.60 SD (46.46)	11573.43 SD(94.37)	2485.80 SD(137.43)	3676.29 SD(95.33)
/u/ in /ʕ/	704.22	1119.64	2631.32	3735.39
/u/ in pcc	464.41 SD (48.66)	1 184.15 SD (178.35)	2542.37 SD (156.04)	3607.9 SD (103.82)
/i/ in / ħ/	593.09	1881.61	2708.745	3754.375
/i/ in /ʕ/	560.56	1971.70	2816.42	3 987.19
/i/ in pcc	378.77 SD (49.50)	1881.35 SD (128.08)	2729.54 SD (188.23)	3681.63 SD (84.33)

Table 3: Average values of the formants with SD for the three short vowels in pharyngeal and plain coronal contexts

of the coarticulatory effects. The studied pharyngeal consonants, by exerting a strong coarticulatory effect, are therefore distinguished mainly by a narrowing of the distance between the two first formants F2-F1. Thus, we can deduce that the increase in the values of F2-F1 reflects a decrease in the coarticulatory effect exerted by the consonant on adjacent vowel, and vice versa. When the difference (in Hz) between the two F2-F1 (plain coronal consonants – pharyngeal consonants) is high, it reflects totally opposite degrees of coarticulatory effect, strong in pharyngeal contexts and weak in plain consonants contexts. When the difference between F2-F1 (plain coronals) and F2-F1 (pharyngeal) tends towards zero or weak, it reflects a coarticulatory effect which is exerted in a similar way in the two contexts. The net result of laryngeal raising is to bring F1 close to F2 for the three studied vowels /a, i, u/.

References

Al-Ani. 1970. *Arabic phonology an acoustical and physiological investigation.*

Al-Tamimi. 2009. Effect of pharyngealisation on vowels revisited: Static and dynamic analyses of vowels in moroccan and jordanian arabic. In *Workshop Pharyngeals and Pharyngealisation, Newcastle University*.

Alwan. 1989. Perceptual cues for place of articulation for the voiced pharyngeal and uvular consonants. *Journal of the Acoustical Society of America*, 86(2):549–556.

Bin-Muqbil. 2006. *Phonetic and phonological aspects of Arabic emphatics and gutturals, PhD dissertation, University of Wisconsin-Maddison*.

Bouarourou. 2014. *la gémination en tarifit, considérations phonologiques, étude acoustique et articulatoire, PhD dissertation,Strasbourg university*.

Kusay Butcher. 1987. Some acoustic and aerodynamic consequences of pharyngeal consonants in iraqi arabic. *phonetica*, 44:156–172.

Byrd. 1994. *Articulatory Timing in English Consonant Sequences, PhD dissertation, University of California, Los Angeles*.

Byrd. 1996. Influences on articulatory timing in consonant sequences. *Journal of Phonetics*, 24:209–244.

Delattre. 1971. Pharyngeal feature in the consonants of arabic, german, french and american english. *phonetica*, 23:129–155.

Djeradi. 2006. *Analyse et characterisation des fricatives d'arriére de l'arabe: au plan acoustique, articulatoire et "timing",PhD dissertation, University of science and technology houari boumediene of Algiers*.

Guilleminot Al-Maqtari Embarki, Yeou. 2007. An acoustic study of coarticulation in modern standard arabic and dialectal arabic: Pharyngealized vs. non-pharyngealized articulation. In *2007 In Proceedings of the 16th International Congress of Phonetic Sciences*, pages 141–146.

Esling. 1996. Pharyngeal consonants and the aryepiglottic sphincter. *Journal of the International Phonetic Association*, 26(2):65–88.

Barkat Ghazali, Hamdi. 2002. Speech rhythm variation in arabic dialects.

Ghazeli. 1977. *Back Consonants and Backing Coarticulation in Arabic,PhD dissertation,University of Texas, Austin*.

Vaxelaire Elie Laprie, Sock. 2014. Comment faire parler les images aux rayons x du conduit vocal. In *In SHS Web of Conferences*, volume 8, page 14. Association for Computational Linguistics.

Baer Laufer. 1988. The emphatic and pharyngeal sounds in hebrew and in arabic. *language and Speech*, 31(2):181–205.

Marchal. 2010. *From Speech Physiology to Linguistic Phonetics,John Wiley and Sons,ISBN, 0470610409, 978047610404*.

McCarthy. 1994. The phonetics and phonology of semitic pharyngeals. In *in Patricia A. Keating, ed., Phonological Structure and Phonetic Form, Papers in Laboratory Phonology Cambridge, UK: Cambridge University Press*, page 191–233.

Laprie Perrier-Vaxelaire Sock, Hirsch. 2011. An x-ray database, tools and procedures for the study of speech production. In *9th International Seminar on Speech Production (ISSP 2011), Montreal, Canada*, pages 41–48.

Sundburg. 2003. Research on the singing voice in retrospect. In *Speech Music and Hearing Laboratory-Quarterly Progress and Status Report, Stockholm*, pages 11–22.

Sylak. 2013. Pharyngealization in chechen is gutturalization. In *Proceedings of the 37th Annual Meeting of the Berkeley Linguistics Society: Special Session on Languages of the Caucasus*, pages 81–95.

Zawaydeh. 1999. *The phonetics and phonology of gutturals in Arabic, Ph.D. Dissertation, Indiana University*.

A Comparative Study on Language Models for the Kannada Language

Danish Ebadulla , Rahul Raman , Hridhay Kiran Shetty , Mamatha H.R.
PES University Bangalore, India
{danishebadulla, rahulraman, hridhayshetty}@pesu.pes.edu
mamathahr@pes.edu

Abstract

We train word embeddings for Kannada, a Dravidian language spoken by the people of Karnataka, a southern state in India. The word embeddings are trained using the latest deep learning language models, to successfully encode semantic properties of words. We release our best models on HuggingFace, a popular open source repository of language models to be used for further Indic NLP research. We evaluate our models on the downstream task of text classification and small custom analogy and similarity tasks. Our best model attains accuracy on par with the current State of the Art while being only a fraction of its size. We hope that by publicly releasing our trained models, we will help in accelerating research and easing the effort involved in training embeddings for downstream tasks.

1 Introduction

Distributed representation is the foundation of NLP, as advances in language modelling serve as a stepping stone for many NLP tasks. Popular domains like text classification, text generation, translation, sentiment analysis, NER etc can be advanced with access to contextualized word embeddings. Rise in quality of embeddings is synonymous with an improvement in downstream NLP tasks.

India is a diverse and rapidly growing country. With advances in technology, electronic devices are making their way into the hands of every citizen of the country, giving them the ability to access information that was previously out of reach for them. But this also presents another problem. India has over 22 official languages and several thousand more languages and dialects. It is of paramount importance that we develop NLP tools that bridge this gap and help India progress faster.

Indian languages are considered resource poor and have very little monolingual corpora that is publicly available for NLP tasks. Dravidian languages in particular are far behind Indo-Aryan languages. With access to such few resources, training a language model is very challenging, as it is very easy to overfit your model and lose its ability to generalise. Many corpora are also domain specific, making it difficult for the model to generalise context.

In this paper, we experiment with the latest language models on the Kannada language. Using the monolingual corpora provided by indicNLP[1] we have trained the models from scratch and fine-tuned them. We evaluate these models on the news dataset classification provided by indicNLP[2]. We also perform some custom word similarity and analogy tests on the generated embeddings. We show that lightweight transformer based models such as RoBERTa (Liu et al., 2019) and ELECTRA(Clark et al., 2020) outperform previously used mainstream models. We release these models on the popular transformers open source repository HuggingFace[3] where our fine tuned models, capable of generating quality word embeddings will significantly improve all Kannada language downstream tasks.

2 Related Work

One of the earliest papers to perform embedding generation on Kannada at scale was fastText by Facebook (Bojanowski et al., 2016). They proposed an improvised approach for the skip-gram model, representing each word as a bag of character n-grams. This overcame the main drawback in Word2Vec (Mikolov et al., 2013), where words were considered as atomic units leading to subpar performance on morphologically rich languages such as Kannada. fastText's embeddings are used as a benchmark for comparison of results in several

[1] https://github.com/AI4Bharat/indicnlp_corpus
[2] https://github.com/AI4Bharat/indicnlp_corpus#indicnlp-news-article-classification-dataset
[3] https://huggingface.co

Indic language model papers.

Anoop Kunchukuttan et al. (Kunchukuttan et al., 2020) released the indicNLP corpus in 2020, a monolingual corpora for 10 Indian languages sourced from various domains and sites. Word embeddings trained on FastText using this corpora were also released. A news classification dataset to be used as a downstream evaluation task was also released. Their embeddings were compared against the original fastText embeddings and were found to outperform the latter in several languages.

Gaurav Arora (Arora, 2020) released the Natural language Toolkit for Indic languages a few months later, which also released embeddings for 13 Indic languages that outperformed indicNLP and fastText. ULMFiT (Howard and Ruder, 2018) and TransformerXL (Dai et al., 2019) were used to train the embeddings and the data sourced from Wikipedia was only a fraction of the indicNLP corpora's size. A 2 step augmentation technique was used to improve the performance of their models. Kumar Saurav et al. (Kumar et al., 2020) also released word embeddings for 14 Indian languages in a single repository, although their results are not competitive with Anoop Kunchukuttam or Gaurav Arora. They trained their embeddings on several transformer architectures such as BERT (Devlin et al., 2018) and ELMo (Peters et al., 2018) and tested them on several custom tasks.

3 Methodology

3.1 Dataset

Our pre-training data is sourced from the indicNLP monolingual corpora, a collection of 10 Indic languages and the iNLTK monolingual corpora. The indicNLP Kannada corpora has 14 million sentences and 174 million tokens. The iNLTK[4] corpora has 26,000 sentences sourced from Wikipedia articles. We use the news classification released by indicNLP for the downstream task of text classification. We also build small custom datasets for word similarity and word analogy tests. To ensure fair comparison for downstream tasks across all models, we train all our models on the same corpora.

3.2 Preprocessing

The corpora was cleaned to remove any foreign tokens and fix formatting errors. Shuffling and deduplication was applied after combining all the corpora sources. An md5 hash was applied to deduplicate the corpora, leaving us with roughly 11 million sentences after it was applied on the corpora. To make initial training easier, any sentences greater than 30 words or having english in more than 30% of the sentence were removed.

3.3 Tokenization and Vocabulary

All our models use either SentencePiece (Kudo and Richardson, 2018) or BertWordPiece for tokenization and vocabulary generation. With the help of Sentence-Piece API [5] tokens were generated by experimenting with the hyper parameters. Vocabulary size ranged from 8,000-32,000 with incremental steps of 4,000. BertWordPiece was trained to generate a vocabulary size of 40,000 with words having a minimum frequency of 4.

Previous works claim that higher vocabulary sizes correspond to a lower chance of Out Of Vocabulary words occuring and this usually translates to better performance in downstream tasks. But without a morphologically motivated technique to segment subwords, increasing the vocabulary size might lead to an increased occurrence of different inflections of the same word. Hence, we decide to compare varying vocabulary sizes and their performance.

3.4 Experimental Setup

We evaluate our models on the downstream task of text classification using the indicNLP news classification dataset which has around 24,000 training examples and 3,000 test examples with 3 labels. All models were trained using a single 12GB NVIDIA Tesla K80 GPU.

4 Models and Evaluation

4.1 Word2Vec

We start off with both the Word2Vec models to set an initial benchmark. The tokenization was done with SentencePiece with byte pair encoding algorithm. The model architecture was provided for by the gensim API[6]. The API also provides a simple interface for tuning hyper-parameters.

The CBOW model gave us better results when compared to the skip-gram model in the word similarity test. We noticed that Word2Vec models with lower vocabulary size had more meaningful words in the similarity predictions, with lower vocabulary models predicting synonyms of the input word

[4]https://github.com/goru001/inltk

[5]https://github.com/google/sentencepiece

[6]https://radimrehurek.com/gensim

and higher vocabulary models predicting inflected forms of the input word. Some of the similar words in higher vocabulary size models had no meaning by itself, which might probably indicate over tokenization, which might be good for predicting unknown words but results in diminished quality of the word embeddings. We hypothesize that this might be due to the small size of the input corpus.

4.2 FastText

The fastText API[7] was used to train our model. The publicly released Kannada language model has an approximate vocabulary size of 1.7 million. With the API we pre-trained a fastText model from scratch with both CBOW and skip-gram architecture. The fastText API takes it's input directly and handles the tokenization. Due to very few hyper parameters provided by the fastText API for tuning the model, further experimentation was done with the gensim API. With the gensim API, first the input data was tokenized with SentencePiece. The API provides hyper parameters for tokenization, vocabulary frequency and the architecture which helped us fine tune our model for better accuracy in the news classification dataset. The API's supervised module was used to perform text classification on the news dataset.

4.3 RoBERTa

Since base BERT models require a large corpus and access to heavy computation resources, we trained embeddings on a RoBERTa model with distilBERT's (Sanh et al., 2019) configuration using the HuggingFace API. Byte Pair Encoding (Sennrich et al., 2015) was used to tokenize the corpus after which the tokenizer weights were transferred to the roBERTa tokenizer. The vocabulary size was set to 32,000 and the model's configuration was set to 6 hidden layers, 12 attention heads and 768 embedding size. The size of the model was 68 M parameters. After the pre-training phase 2 linear layers were added to fine tune the model on the classification task. As RoBERTa performs in-memory tokenization, due to resource constraints we were unable to train the model on the entire corpora that was used for Word2Vec and fastText. The model was trained for 1 epoch. Hyper parameters such as batch size, hidden layers, number of attention layers and the embedding size were tuned to accommodate the decreased model size.

[7]https://github.com/facebookresearch/fastText

4.4 ELECTRA

We also trained embeddings using one of Google research's newer models, ELECTRA. It is a BERT model that performs Masked Language Modelling using a discriminator instead of a generator. The model is trained to corrupt words with high probability in place of the MASK and the discriminator tries to identify these corrupt words. Since ELECTRA generates `tf.pretrain` records of the input corpora and stores them offline, it is not limited by memory and is capable of training on the entire corpora. The model uses the BertWordPiece tokenizer. The vocabulary size was set to 40,000. We used the 'small' version of the model which has 14M parameters and trained it for 200,000 steps. Maximum sequence length was set to 512. After pre-training the model, it was fine tuned and evaluated on a text classification task using the ktrain library on the Kannada news articles dataset. The pretrained model has been uploaded to HuggingFace[8] for future use.

5 Results

Our models are compared against Facebook's fastText model trained on Wikipedia and CommmonCrawl, indicNLP and iNLTK on a text classification task using the indicNLP News Classification dataset. The results are documented in Table 1.

We found that our Word2Vec and fastText models trained with a vocabulary size of 8,000 had more meaningful similar word predictions compared to the same models with a 32,000 vocabulary size. fastText outperformed Word2Vec models and our fastText model's accuracy was marginally lower than the original fastText model. Figure 1 shows some notable results from our experiments on word similarity. Word2Vec results were observed to be heavily influenced by the domain of the dataset and contained pronouns in word similarity results as it considers word as the atomic token value. In comparison, fastText produces significantly better results at it considers the n-gram characters' information as an atomic unit.

We can also observe that the lower vocabulary models produce words that are synonyms of the input word while the large vocabulary models produce inflections of the same word. The official fastText model had very different words at the morpheme level but these words were distinctly similar to the actual word.

[8]https://huggingface.co/DarkWolf/kn-electra-small

Model	Word Similarity					
	ರಾಜ (king)			ಮನುಷ್ಯ (man)		
W2V - cbow	ರಾಜನಾದ Rājanāda 'The king'	ಪ್ರವಾದಿ Pravādi 'Prophet'	997ರಲ್ಲಿ 997Ralli 'In 997'	ಫರಿಸಾಯನು Pharisāyanu 'The Pharisee'	ಪುತ್ರನೇ Putranē 'Son'	ಮನುಷ್ಯನನ್ನು Manuṣyanannu 'The man'
W2V - sg	ದಾವೀದ Dāvīda 'David'	ಸೊಲೊಮೋನ Solomōna 'Solomon'	ಅಮಜ್ಯನು Amājiyā 'Amaziah'	ಬಿದ್ದುಹೋಗುವದು Bidduhōguvadu 'To fall off'	ಮನುಷ್ಯನ Manuṣyana 'Man's'	ಇಸ್ರಾಯೇಲನಿಗೆ Isrāyēlanige 'For the Israelites'
FT - inltk	ರಾಜನ Rājana 'King's'	ರಾಜ್ Rāj 'Raj'	ರಾಜಕೀಯ Rājakīya 'Politics'	ಮನುಷ್ಯರ Manuṣyara 'Human beings'	ಭಗವಂತ Bhagavanta 'Lord'	ದೆವ್ವ Devva 'Devil'
FT - 8K	ಅರಸ Arasa 'King'	ಅರಸನಾದ Arasanāda 'The king'	ಕುಮಾರ Kumāra 'Son'	ಪುರುಷನು Puruṣanu 'The man'	ಪುರುಷ Puruṣa 'Male'	ಮನುಷ್ಯನಿಂದ Manuṣyaninda 'By man'
FT - 16K	ರಾಮ Rāma 'Rama'	ಪ್ರವಾದಿ Pravādi 'Prophet'	ಅರಸ Arasa 'King'	ಕುಮಾರನು Kumāranu 'Son'	ಸ್ತ್ರೀಗೆ Strīge 'Female'	ಹುಡುಗ Huḍuga 'Boy'
FT - 32K	ನಾಥ Nātha 'Nath'	ನಾಟ Nāṭa 'Nata'	ರಾಜನ Rājana 'King's'	ಹುಡುಗಿಯ Huḍugiya 'Girl'	ಮನುಷ್ಯನಿಂದ Manuṣyaninda 'By man'	ಫರಿಸಾಯನು Pharisāyanu 'The Pharisee'
FT - original	ಸಿಂಘನನು Siṅghananu 'Lioness'	ರಾಣಿ Rāṇi 'Queen'	ಎನುತಲೆ Enutali 'Chattering'	ಪ್ರವಾದಿಯೂ Pravādiyū 'Prophetic'	ಪುತ್ರನೇ Putranē 'Son'	ಮನೆಯವರೇ Maneyavarē 'Housekeeper'
FT - sg	ರಾಜನು Rājanu 'The king'	ರಾಜನ Rājana 'King's'	ರಾಜೀ Rājī' 'Rajee'	ಬಿದ್ದುಹೋಗುವದು Bidduhōguvadu 'To fall off'	ಮನುಷ್ಯನಿಗೆ Manuṣyanige 'For the man'	ಇವನು Ivanu 'He'
FT - cbow	ರಾಜನ Rājana 'King's'	ರಾಜನಾದ Rājanāda 'The king'	ರಾಜವೈಭವದ Rājavaibhavada 'Of royalty'	ಮನುಷ್ಯನ Manuṣyana 'Man's'	ರೊಳಗಿಂದ Roḷaginda 'From within'	ಮನುಷ್ಯನಿಂದ Manuṣyaninda 'By man'

Figure 1: **Word Similarity**. W2V - Word2Vec ; FT - fastText ; cbow - continuous bag of words ; sg- skip-gram

The RoBERTa model, despite being trained for only 1 epoch and on only a subset of the corpora, outperforms Word2Vec and fastText in text classification, also managing to beat indicNLP on the same task with a classification accuracy of 97.37%.

Our ELECTRA model is built using the 'small' version with 14M parameters and was fine tuned on the text classification task after pre-training for 200,000 steps. It obtained a classification accuracy of 98.53%. This accuracy is only marginally lower than iNLTK's accuracy on the same task, despite having a fraction of the parameters.

Model Name	Vocab Size	Accuracy
fastText WC	1.7M	96.2%
fastText_32	32K	94.2%
fastText_14	14K	94.2%
fastText _8	8K	94.3%
indicNLP	-	97.2%
RoBERTa	32K	97.37%
ELECTRA	40K	98.53%
iNLTK	25K	98.87%

Table 1: Text classification accuracy. The models in italics are our models while the others are popular Kannada language models for comparison

6 Future Work

Future work will involve training ELECTRA models on all Indic languages. ELECTRA has proven to be a competitive and efficient choice to develop language models for Indic languages by achieving accuracy on par with much larger and compute intensive BERT models. BERT based models have proved to be superior to the previously utilised mainstream models like Word2Vec and FastText. With more training and fine tuning, lightweight BERT models might even be able to outperform their mainstream counterparts in low resource settings. We believe that the vocabulary size dilemma can be overcome by using a linguistically motivated subword segmentation technique like Morfessor[9]. This will help us identify frequently occurring suffixes and eliminate the occurrence of inflections in the vocabulary. We also intend to release RoBERTa and ELECTRA pretrained models on HuggingFace for Kannada and other Indic languages to further research on distributed representation.

[9]https://github.com/aalto-speech/morfessor

References

Gaurav Arora. 2020. inltk: Natural language toolkit for indic languages. *CoRR*, abs/2009.12534.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomás Mikolov. 2016. Enriching word vectors with subword information. *CoRR*, abs/1607.04606.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: pre-training text encoders as discriminators rather than generators. *CoRR*, abs/2003.10555.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *CoRR*, abs/1901.02860.

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.

Jeremy Howard and Sebastian Ruder. 2018. Fine-tuned language models for text classification. *CoRR*, abs/1801.06146.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *CoRR*, abs/1808.06226.

Saurav Kumar, Saunack Kumar, Diptesh Kanojia, and Pushpak Bhattacharyya. 2020. "a passage to India": Pre-trained word embeddings for Indian languages. 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 352–357, Marseille, France. European Language Resources association.

Anoop Kunchukuttan, Divyanshu Kakwani, Satish Golla, Gokul N. C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. Ai4bharat-indicnlp corpus: Monolingual corpora and word embeddings for indic languages. *CoRR*, abs/2005.00085.

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.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *CoRR*, abs/1802.05365.

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.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *CoRR*, abs/1508.07909.

Using Bloom's Taxonomy to Classify Question Complexity

Sabine Ullrich
Research Institute CODE
Universität der Bundeswehr München
sabine.ullrich@unibw.de

Michaela Geierhos
Research Institute CODE
Universität der Bundeswehr München
michaela.geierhos@unibw.de

Abstract

Question answering is widespread and a variety of answer taxonomies exists in research that divides responses into simple and complex. Multi-hop answering has become popular when the complexity of questions and answers increases. However, determining when multi-hop reasoning becomes necessary is not yet clear.

We propose to apply Bloom's taxonomy to the determination of question complexity in question-answering systems. Originating in pedagogy, Bloom's taxonomy measures question complexity to determine learning progress levels. Subsequently, the determined question complexity can help in deciding whether an entity or phrase is sufficient as an answer or whether reasoning chains should be given.

1 Introduction

When determining the answer type in a question-answering (QA) system, the question type must be considered first. While entities or short sentences are sufficient for simple, factual questions, more complex questions require more complex answers. For example, simple product-related questions, such as "Does Kindle support Japanese?", can be easily answered by a yes/no response. When extracting interpretative questions that require logical thinking, reasoning chains can be used to generate answers. Imagine a complex question such as "What is the current situation in Syria?". Answering this question is not easy and cannot be done by a simple knowledge graph or ontology. To explain why this answer is correct and to provide a cohesive line of argumentation, multi-hop reasoning chains are required to connect successive propositions.

While several approaches exist that present taxonomies for question and answer types, the complexity of questions has not yet been measured to classify the required answers. Assuming that complex questions require complex answers, we need to ask the question "What makes a question complex?". How can we determine the complexity of a question and at what level of complexity are multi-hop reasoning chains useful or even essential?

In pedagogy, Bloom's Taxonomy of Educational Objectives (Bloom et al., 1956) helps to capture a learner's level of understanding. At the lowest level of the taxonomy, simple memorization is required to reproduce a fact or concept, while as the level increases, the abstraction level also increases. The lower levels serve as base knowledge, while higher levels represent the deeply processed knowledge that can be abstracted and transferred for specific purposes (Cannon and Feinstein, 2005).

This paper examines how Bloom's Taxonomy can be used to classify questions in QA systems according to their complexity. Furthermore, it discusses which factors contribute to the complexity of a question and when multi-hop reasoning is required instead of simple information extraction.

2 Related Work

Several approaches attempt to classify answers in QA systems by constructing a question taxonomy. Questions are grouped either flatly (Eichmann and Srinivasan, 1999; Litkowski, 1999) or hierarchically (Takaki, 2000; Suzuki et al., 2003). Kim (2014) proposes a method for defining answers and ambiguity within questions. Moreover, taxonomies exist for specific question types such as the taxonomy for opinion questions (Bayoudhi et al., 2013), classifications based on data source, analysis types, and response forms (Mishra and Jain, 2016).[1] However, none of these surveys defines how these taxonomies can be used to calculate question complexity.

[1] For an extensive list see Sundblad (2007).

Datasets containing multi-hop reasoning chains are widely used (Yang et al., 2018; Jhamtani and Clark, 2020; Wiegreffe and Marasović, 2021). Reasoning chains provide appropriate answers to questions posed in the respective datasets. For general questions asked in QA scenarios, it is unclear if or when a multi-hop reasoning chain is required as an answer. This is because question complexity measurement and reasoning chains have not yet been combined.

Often, question complexity is used in education to determine the difficulty of student exams. For example, Luger and Bowles (2013) measure the difficulty of multiple choice questions. Research on community QA services is often domain-specific, comparing the difficulty of topic-related words within certain domains (Liu et al., 2013; Wang et al., 2014). Others use provided meta-information such as user expertise to estimate question difficulty (Sun et al., 2018) or measure relative complexity by comparing users' questions (Thukral et al., 2019).

Research most closely related to ours comes from Padó (2017), which shows how Bloom's Taxonomy can approximate the difficulty of questions in a short-answer corpus. Together with measuring the diversity of student responses, the difficulty can be estimated from lower to higher levels of the taxonomy. In addition, textual entailment methods can infer levels from the question wording (Anderson and Krathwohl, 2014). However, their approach is only used in the context of grading students, so we propose to adapt it for measuring question complexity in QA systems.

3 Approach

Our method combines Bloom's Taxonomy (Bloom et al., 1956) and question classification for QA systems. We plan to classify the difficulty of questions by grouping them in Bloom's revised matrix (Anderson and Krathwohl, 2014). This matrix contains two dimensions: the knowledge dimension on the vertical axis and the cognitive process dimension on the horizontal axis (Cannon and Feinstein, 2005). This means that the complexity to understand and answer a question increases from left to right, and the complexity of knowledge further increases from top to bottom. The squares in the matrix were left empty by Cannon and Feinstein (2005). We fill each of these squares with typical question keywords, ranging from simple factual questions ("list", "define", "name") to more complex questions ("explain", "analyze", "justify"). These keywords can then in turn be mapped to specific question types. For example, "Who invented..." might be a representative for a factual question in the cognitive process dimension "remember". The three steps to follow are ...

1. filling in the matrix with keywords,
2. assigning categories to question types, and
3. defining the difficulty for the question types.

The question we want to answer is at what level of knowledge and cognitive level multi-hop reasoning is required. The levels could then be used as a basis for classifying responses, as more complex questions will require complex answers. Determining the threshold of complexity in some experiments remains for future work. The approach is also intended to give a very general idea of how to measure question complexity, which is why domain dependence is not considered.

We use existing keywords that we can map to Bloom's Taxonomy and perform classification on a QA dataset. The questions of the dataset are analyzed syntactically such that the model can be independently applied to other domains.

4 Proof of Concept

In the following, we will show how to map Bloom's Taxonomy to question difficulty estimation. Therefore, typical question keywords will be filled into the revised matrix of Bloom's Taxonomy. Then, the questions will be tagged with their respective Part-of-Speech (PoS) tags to capture their syntactic features. For classification, a multi-layer perceptron (MLP) will be trained and evaluated on the development data.

4.1 Keyword Mapping

To establish a connection between pedagogy and QA systems, we fill in typical question indicator words from educational studies into the revised version of Bloom's Taxonomy. For each slot in the matrix, keywords used by Bloom to estimate question complexity were assigned to their respective categories. This is important because (a) the keywords may appear directly in search queries and (b) the keywords may be used later to assign question words and templates to categories. The results are presented in Table 1.

The Knowledge Dimension	The Cognitive Process Dimension				
	1 Remember	2 Understand	3 Apply	4 Analyze	5 Evaluate
A Factual	name, list define, label	restate order	state determine	distinguish classify	select according to
B Conceptual	identify locate	describe explain	illustrate show	examine analyze	rank compare
C Procedural	tell describe	summarize translate	solve demonstrate	deduct diagram	conclude choose
D Meta Cognitive	–	interpret paraphrase	find out use	infer examine	justify judge

Table 1: The knowledge dimension matrix by Cannon and Feinstein (2005) filled with key indicators for complex questions. The complexity ranges from low (top left) to very high (bottom right).

The keywords in the matrix indicate the level of complexity within a search query. While simpler questions are located at the top left, more complex questions are positioned on the bottom right of the table. In the next step, questions from the Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al., 2016) will be extracted, to map the keywords to question terms. This helps to extract a selection of questions and classify it according to the cells in the matrix. The question selection will be described in more detail in the next section.

4.2 Question Selection

Next, the keywords from Table 1 are used to extract and classify questions from a QA dataset. The procedure is as follows: For each keyword in the matrix, search through the dataset and extract questions that contain these keywords. For our proof of concept, we searched SQuAD, a reading comprehension dataset consisting of 100,000+ questions from Wikipedia articles. In the dataset, all sentences are searched by keyword and marked correspondingly.

There are two obstacles to overcome. The first is the small amount of about 770 questions that contain keywords. The second is the unequal distribution of samples across the classes. For some categories no questions exist (D5), some classes only have 2 samples (C4) and others are overrepresented (D3) with 414 samples. Two classes have significantly more samples than the rest, namely A1 with 362 samples and D3 with 414 samples. We circumvent both obstacles by transforming the task into a binary classification task and by defining representatives for simple questions (A1) and complex questions (D3). We then argue that the complex samples from D3 will require multi-hop reasoning answers.

To abstract the question structure of the training set, all words are annotated with their respective PoS tag. Since question words may indicate the question complexity, they are included without any adaptations. This will allow us to derive the complexity of a question from its underlying syntactic structure. An example from the class 0 (A1) looks as follows:

Name	an	example	of	a	heavy	isotope
VERB	DET	NOUN	ADP	DET	ADJ	NOUN

An example from class 1 (D3) shows how question particles remain untagged:

What	number	is	*used*	in	perpendicular	computing
WHAT	NOUN	AUX	VERB	ADP	ADJ	NOUN

In the next step, the annotated sentences are used for classifier training. The classifier and the training process are described in the following section.

4.3 Question Classification

The question set comprises about 770 samples and is split into 90% training and 10% validation sets. Following the Google developers guide for choosing our classification model, we calculate the samples/number of words per sample. For a ratio smaller than 1,500, they advise to choose a word n-gram-based MLP. Therefore, we split the samples into n-grams (where $n = \{1, 2, 3, 4\}$) and convert the numbers into vectors. Subsequently, the vectors are scored by importance using tf-idf (short for term frequency-inverse document frequency).

The vectors are fed into the MLP with 3 layers and 64 units and trained for 15 epochs. We added a dropout of 0.2 and early stopping with $patience = 3$ on validation accuracy to prevent overfitting of the model. The results are presented in the next section.

4.4 Evaluation

The results show that binary classification, which distinguishes between classes 0 (simple answer) and 1 (multi-hop answer) with A1 and D3 as representatives, yields good results with a simple MLP. The training loss could be reduced after 15 epochs to 0.21 with an accuracy of 0.92, a validation loss of 0.42, and a validation accuracy of 0.85. Figure 1 shows the loss and validation values for each epoch. The best results were obtained with PoS tag n-grams where $n = \{1, 2, 3, 4\}$ and a learning rate of 1e-3.

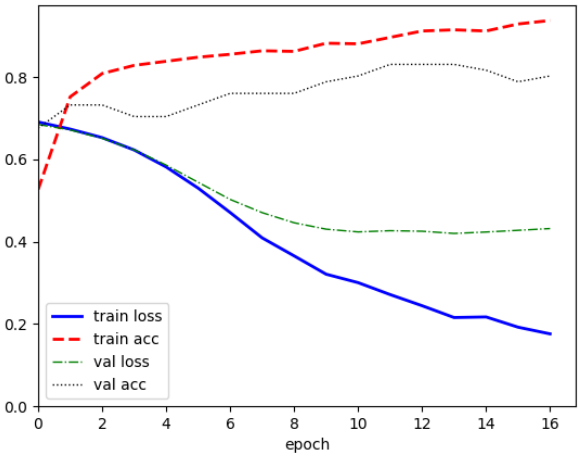


Figure 1: Model loss and accuracy per epoch for 15 epochs after early stopping on validation accuracy with a patience of 3.

When evaluating the model on the validation data, we obtain a weighted F1 value of 0.85 for both classes, with an F1 value of 0.72 for class 0 and 0.88 for class 1. Class 1 achieves the highest recall value of 0.92. A look at the confusion matrix (Figure 2) shows that the vast majority of classes were assigned correctly.

To compute the complexity level, the diagonal of the matrix could be a determinant for the definition of complex questions. For automated calculations, add 1 for each step to the right and down the matrix. If the value is greater than a certain threshold, multi-hop reasoning can be considered.

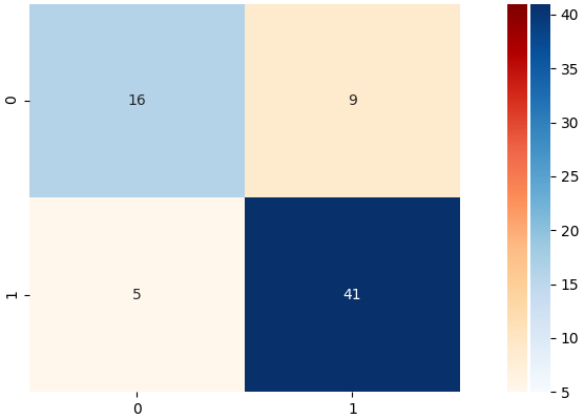


Figure 2: In this confusion matrix for binary classification, 0 represents simple answers (A1 in Bloom's Taxonomy) and 1 means that multi-hop answers are required (D3 in Bloom's Taxonomy).

5 Conclusion and Future Work

We have shown that Bloom's revised taxonomy can be transferred from pedagogy to QA systems. The diagonal of the matrix is a determinant for defining complex questions, ranging from simple questions in the upper left to complex questions on the bottom right. For the proof of concept, we added PoS tags to the questions as syntactic information to train a domain-independent classifier for question complexity. We argued that question words also contribute to complexity, so they were not transformed. Although the unequal distribution of the training data only allowed a binary classification for two representative classes A1 and D3, the classifier already provides good results for computing question complexity.

In the future, we plan to collect a larger number of questions from different types of datasets so that a greater diversity of questions is captured. This is crucial for obtaining a diverse data source with a balanced combination of simple and complex questions. It also allows us to expand the question pool so that more classes can be included in the classification. Next to PoS tagging, a wider variety of linguistic features that contributes to the complexity of a question should be considered. This includes the sequence length and the inclusion of semantic information in the classification model. Finally, a user study could help to determine the specific threshold within Bloom's Taxonomy that indicates the need for multi-hop reasoning.

References

Lorin W. Anderson and David A. Krathwohl, editors. 2014. *A taxonomy for learning, teaching and assessing: A revision of Bloom's*. Pearson Edition.

Amine Bayoudhi, Hatem Ghorbel, and Lamia Hadrich Belguith. 2013. Question Answering System for Dialogues: A New Taxonomy of Opinion Questions. In *International Conference on Flexible Query Answering Systems*, pages 67–78. Springer.

Benjamin S Bloom, Max D Engelhart, EJ Furst, Walker H Hill, and David R Krathwohl. 1956. Handbook I: cognitive domain. *New York: David McKay*.

Hugh M Cannon and Andrew Hale Feinstein. 2005. Bloom beyond Bloom: Using the revised taxonomy to develop experiential learning strategies. In *Developments in Business Simulation and Experiential Learning: Proceedings of the Annual ABSEL conference*, volume 32.

David Eichmann and Padmini Srinivasan. 1999. Filters, webs and answers: The University of Iowa TREC-8 results. In *Proceedings of the 8th Text Retrieval Conference (TREC-8)*. Citeseer.

Harsh Jhamtani and Peter Clark. 2020. Learning to explain: Datasets and models for identifying valid reasoning chains in multihop question-answering. In *Proc. of the 2020 EMNLP*, pages 137–150.

Yang-woo Kim. 2014. Typology of ambiguity on representation of information needs. *Reference and User Services Quarterly*, 53(4):313–325.

Kenneth C Litkowski. 1999. Question-Answering Using Semantic Relation Triples. In *Proceedings of the 8th Text Retrieval Conference (TREC-8)*, pages 349–356.

Jing Liu, Quan Wang, Chin-Yew Lin, and Hsiao-Wuen Hon. 2013. Question difficulty estimation in community question answering services. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 85–90.

Sarah KK Luger and Jeff Bowles. 2013. Two methods for measuring question difficulty and discrimination in incomplete crowdsourced data. In *First AAAI Conference on Human Computation and Crowdsourcing*.

Amit Mishra and Sanjay Kumar Jain. 2016. A survey on question answering systems with classification. *Journal of King Saud University-Computer and Information Sciences*, 28(3):345–361.

Ulrike Padó. 2017. Question difficulty–how to estimate without norming, how to use for automated grading. In *Proceedings of the 12th workshop on innovative use of NLP for building educational applications*, pages 1–10.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2383–2392.

Jiankai Sun, Sobhan Moosavi, Rajiv Ramnath, and Srinivasan Parthasarathy. 2018. QDEE: question difficulty and expertise estimation in community question answering sites. In *Twelfth International AAAI Conference on Web and Social Media*.

Håkan Sundblad. 2007. *Question classification in question answering systems*. Ph.D. thesis, Institutionen för datavetenskap.

Jun Suzuki, Hirotoshi Taira, Yutaka Sasaki, and Eisaku Maeda. 2003. Question classification using HDAG kernel. In *Proceedings of the ACL 2003 workshop on Multilingual summarization and question answering*, pages 61–68.

Toru Takaki. 2000. NTT DATA TREC-9 Question Answering Track Report. In *Proceedings of the 9th Text Retrieval Conference (TREC-9)*.

Deepak Thukral, Adesh Pandey, Rishabh Gupta, Vikram Goyal, and Tanmoy Chakraborty. 2019. DiffQue: Estimating relative difficulty of questions in community question answering services. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(4):1–27.

Quan Wang, Jing Liu, Bin Wang, and Li Guo. 2014. A regularized competition model for question difficulty estimation in community question answering services. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1115–1126.

Sarah Wiegreffe and Ana Marasović. 2021. Teach me to explain: A review of datasets for explainable NLP. *arXiv preprint arXiv:2102.12060*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2369–2380.

Towards Phone Number Recognition For Code Switched Algerian Dialect

Khaled Lounnas
LCPTS-FEI, USTHB
Algiers, Algeria
klounnas@usthb.dz

Mourad Abbas
High Council of Arabic Language
Algiers, Algeria
abb.mourad@gmail.com

Mohamed Lichouri
LCPTS-FEI, USTHB
Algiers, Algeria
mlichouri@usthb.dz

Abstract

This paper addresses the problem of phone number recognition taking into account some of the peculiarities of dialectal Arabic used in the daily life of Algerian people as code-switching and accent variety. Accordingly, we have set up an ASR system aiming to have the capacity to cope with these peculiarities, i.e. to recognize sequences of digits that may have been spoken in Algerian dialect, in French, or in both. For this purpose, we built an in-house corpus composed of 100 couples of digits (from '00' to '99') spoken in French and dialectal Arabic by two persons. The findings show that, our ASR system behaves more or less effectively when dealing with one language. In fact, it yielded a WER of 1.4 % for French, and 7.1 % for both Blida and Algiers dialect, using 13 MFCC's Coefficients and 32 GMMs. Unfortunately, due to the code-switching phenomenon between these dialects and French, and the limited data size, the performance degrades drastically and reaches a WER of 17.3% with 13 MFCCs and 64 GMMs.

1 Introduction

Automatic Speech recognition (ASR) system provides users with the ability to use their voices rather than the keyboard to search for information. Nowadays, mobile phones have emerged as a trendy area offering attractive platforms for speech recognition-based functions that can help in solving various issues in mobile telephony (Varga et al., 2002), automated contact centres and consumer electronics items. Many applications have been implemented as intelligent telephone answering systems (Lobanov et al., 1997) that use standard speech modems to respond to incoming calls and recognize the call recipient and caller's name. Interactive voice response (IVR) systems can be used for mobile purchases, banking payments, services, retail orders, travel information.

A number of manufacturers currently offer mobile phones with built-in voice interfaces (Wu et al., 1998; Tabani et al., 2017). Most of these interfaces are developed to support particular languages (Salimbajevs, 2018), for example English,French and Hindi (Deka et al., 2018), whereas there are many other languages and dialects, especially those with low resources such as Arabic dialects.

The purpose of this work is to implement an application to recognize digits spoken in Algerian dialects, taking into account the code switching phenomenon that characterizes these dialects. In this direction, we first developed a new in-house speech corpus composed of one hundred spoken digits (from "00"-"99"). This corpus was recorded by two native speakers from two different cities in Algeria: Algiers, Blida. Furthermore, the same speakers recorded the digits (from "00"-"99") in French . This corpus is used for training the acoustic and language models in different configurations in different situations provided that users may pronounce the sequence of digits in both dialectal Arabic and French spoken with different accents. We will give more details about this in the following sections.

This paper is organized as follows: literature review is presented in section 2. In section 3, a brief description of the linguistic material has been given. Section 4 is devoted to experiments and results. Finally, the conclusion is presented in section 5.

2 State of the art

2.1 Spoken Digits Corpora

Unlike their previous work on designing ASR based on romanized characters (Satori et al., 2007), Satori et al in (Satori et al., 2009) built an in house speech based digits corpora to train the CMU Sphinx tool in an entirely Arabic environment. Other researches focused on applying Deep Neural Network (DNN) technology to solve Spo-

ken Arabic digits recognition issue, in (Mahfoudh Ba Wazir and Huang Chuah, 2019) authors used a corpus made of 1040 spoken digits. The aim is to design Long Short-Term Memory (LSTM) model which has the capability to treat problems associated with temporal dependencies requiring long-term learning and to solve the vanishing gradient problems associated with RNN. their reported findings show that the LSTM model can achieve 69% in accuracy when recognizing spoken Arabic digits. In (Sharmin et al., 2020), authors carried out experiments on the first ten digits spoken in Bengali. They achieved classification by the way of Convolutional Neural Network (CNN), where they obtained an accuracy of 98.37%. In the same context, it has been shown in (Sharan, 2020) that using Wavelet Scalogram and CNN performed on a dataset including 56,290 segments belonging to ten spoken digits, their proposed approach surpass other methods and achieve a test error of 2.84% . In (Djellab et al., 2017), the authors conducted a study on the classification of regional accents in complex linguistic environments in the case of Algerian dialects, tested on their prepared corpus entitled Algerian Modern Colloquial Arabic Speech Corpus.

2.2 Speech recognition

(Alotaibi et al., 2008) addressed the process of automatic digits recognition of Saudi dialect by implementing a Hidden Markov Model using from SAAVB corpora (Alghamdi et al., 2008). The findings show that the proposed system reach 93.67% overall correct rate of digit recognition. In (Lounnas et al., 2020), authors investigated at what extent language identification can improve the performance of the Moroccan Automatic Speech Recognition (ASR) system, they found that their proposal greatly improved the overall accuracy, and outperform the baseline system by 33%. Likewise, (Ghangam et al., 2021) propose an approach which consists in designing a compact multilingual speech recognition system based on language identification, the results show that their approach is low memory consumption with an improvement of WER by 30%, compared to the baseline approach.

3 Linguistic Material

Building a typical corpus is a fundamental step for any engineering system. For this reason, we designed our own corpus by soliciting two speakers from two different Algerian cities: Algiers and Blida, to record one hundred digits from "00" to "99", each digit being repeated 15 times. The recordings have been divided into short segments using Praat tool [1]. In Table 1, we show some statistics related to the developed corpus .

Features	Value
Sampling rate	16 KHz
Number of bits	16 bits
Number of Channels	1, Mono
Audio data file format	.wav
# Speakers	2
#Speakers per dialect	2
# Dialect	2
# Language	2
# Tokens per speaker	1500
# speaker's gender	Male
# Total number of tokens	6000
#Number of digits	100 digits (AAD)
	100 digits (BAD)
	100 digits (FR_{AAD})
	100 digits (FR_{BAD})
# Repetitions per word	15
Condition of noise	normal life
Preemphased	$1 - 0.97z^{-}1$
Window type Hamming	25.6 ms
Frames overlap	10 ms

Table 1: Details on the corpus. AAD stands for Algerian Arabic Dialect, BAD: Blida Arabic Dialect, FR_{AAD} and FR_{BAD}: French spoken in AAD and BAD accents, respectively.

4 Experiments and Results

We achieved a number of experiments to measure the performance of the spoken digit recognition system according to the following three situations:
- Building four (acoustic and language) models by training individually each of the following corpus categories: (BAD, AAD, FR_{AAD} and FR_{BAD}).
- Building two (acoustic and language) models based upon (BAD, AAD) and (FR_{AAD}, FR_{BAD}) respectively.
- Building one global (acoustic and language) model based on BAD, AAD, FR_{AAD} and FR_{BAD} in front of speech comprising BAD, AAD and the code switched sequences.

Note that we used CMU-sphinx in the system de-

[1] http://www.fon.hum.uva.nl/praat/

MFCC	13		26		39	
GMM	WER	SER	WER	SER	WER	SER
2	9.5	23.2	10.2	21.4	23.4	46.8
4	5.5	13.6	4.1	10.2	14.6	30.4
8	**2.4**	6.2	2.6	6.6	11	23.2
16	3.2	7.2	4.7	11.6	8.6	19.2
32	4.9	11.6	4.5	10.6	8.8	19
64	10.6	23	8.8	19.6	10.6	21.8

Table 2: Performance of the ASR system for AAD dialect.

MFCC	13		26		39	
GMM	WER	SER	WER	SER	WER	SER
2	16.4	32.2	16.1	30.4	19.8	36.2
4	8.8	17.8	11.1	24.8	14.4	27.6
8	5.8	12.2	5.9	13.8	9.5	19
16	4.8	11	5	11	9.5	18.4
32	**3.8**	8.4	6.4	14.8	9.8	20
64	5	11.2	8.4	19.8	11.7	24.8

Table 3: Performance of the ASR system for BAD dialect.

sign [2]. We defined the size of the acoustic features (MFCC's) to 13, 26 and 39 coefficients, in addition to different values of GMM 2, 4, 8, 16, 32, 64.

4.1 Monolingual Spoken Digits Recognition

Table 2 and 3 present the WER and SER obtained through different setup for dialectal Arabic (dialects spoken in Algiers and Blida). The best WER for AAD was 2.4% using a default configuration (8 GMM and 13 MFCC) followed by a WER of 3.2% with 16 GMM's and 13 MFCC's . The best score achieved for BAD was with (32 GMM and 13 MFCC). It should be noted that for both Dialects 13 MFCC's coefficients is usually enough to get the highest performance.

MFCC	13		26		39	
GMM	WER	SER	WER	SER	WER	SER
2	6	12.6	6.4	13.8	9.3	19.8
4	4.3	9.4	4.8	10.6	5.9	12.4
8	**2.6**	5.4	4	8.2	5.2	10.4
16	4	6.8	**1.8**	3.6	4.8	9.8
32	5.3	10	6.6	11.4	8	14.2
64	9.6	14	11.8	17.4	15.4	23.6

Table 4: Performance of the ASR system for FR_{AAD}.

In a similar way, we present in tables 4 and 5, performance obtained through multiple configurations for French digits spoken by both Algiers and Blida's people (FR_{AAD}, FR_{BAD}), respectively. It can be noticed for FR_{AAD}, in table 4, that the best WER (1.8%) is obtained using the configuration (16 GMM and 26 MFCC). The second best result is achieved using the configuration (8 GMM and 13 MFCC) with a WER of 2.6%. In the case of FR_{BAD}, WER of 1.9% is achieved with 16 GMM and 13 MFCC followed by and WER of 2.1% achieved by the default setup (8 GMM and 13 MFCC). The performance obtained for French is slightly higher than that obtained for Blida and Algiers dialects.

MFCC	13		26		39	
GMM	WER	SER	WER	SER	WER	SER
2	13.7	28.8	12.4	26.4	15.2	31
4	4.6	10	7.3	15.2	8.4	17.6
8	2.1	4.4	5.5	11.6	7.1	15
16	**1.9**	3.8	3.7	7.4	6.1	12.6
32	2.5	5.4	4.1	8.6	5.9	12
64	7.1	13	9.1	17.4	11.2	21.4

Table 5: Performance of the ASR system for FR_{BAD}.

4.2 Bilingual / Multilingual Spoken Digits Recognition

Unlike the Monolingual ASR system where we trained four models using the four corpora (FR_{AAD}, FR_{BAD}, BAD, AAD) separately, we built two models for bilingual ASR based on training the merged corpora of the couples (FR_{AAD}, FR_{BAD}) and (BAD, AAD), in addition, we trained one single model for Multilingual ASR based upon the four corpora.

For bilingual ASR, as can be noticed in table 6, results show that merging (FR_{AAD}, FR_{BAD}) improved the performance (reduction of WER by 0.4%). On the contrary, recognition of the digits spoken in the two Arabic dialects has been degraded. In fact, the best obtained WER is 7.1% (table 7), which is less than WER recorded for monolingual ASR: (AAD, 2.4%) and (BAD, 3.8%). The reason is that French digits are spoken in standard way by the two speakers which makes the related corpus bigger unlike the two Arabic dialects. However, the difference in pronunciation of the two speakers makes the AAD and BAD corpora different than the French one is. Note that we found more than 50 couple of digits spoken differently in the two Arabic dialects.

For multi-lingual ASR, a global model has been trained based on the whole corpus comprising

[2] https://cmusphinx.github.io/

MFCC	13		26		39	
GMM	WER	SER	WER	SER	WER	SER
2	12.8	26.9	13.4	27.8	14.1	28.7
4	6.5	13.7	6.5	13.9	9.3	19.7
8	3.5	7.4	4.8	10.2	6.9	14.9
16	1.9	4	4.4	9.3	6.3	13.5
32	**1.4**	2.8	3.5	7.3	6.4	13.4
64	2.3	4.5	4.3	8.9	6	12.4

Table 6: Performance of the ASR system for French spoken in both FR_{AAD} and FR_{BAD}.

MFCC	13		26		39	
GMM	WER	SER	WER	SER	WER	SER
2	26.3	47.9	21.6	42.6	27.7	49.5
4	17.4	34.3	15.1	32	19.7	37.6
8	12.6	26.2	10.7	21.6	15.1	27.9
16	8.8	18	8.5	18.1	11.5	22.4
32	**7.1**	15	8.8	17.3	10.9	20.9
64	7.4	15.9	8.2	16.8	11	21.2

Table 7: Performance of the ASR system for both AAD and BAD dialects.

BAD, AAD, FR_{AAD} and FR_{BAD} corpora. This is to deal with code switching phenomenon that characterizes the sequences to be recognized. The best WER obtained is about 17.3% using 13 MFCC and 64 GMM. These results show the necessity to use Monolingual ASR approach which is best performing on condition to integrate a language identification component.

5 Conclusion

In this paper, we tackled the digits recognition issue for code switched Algerian dialect. The main challenge is that Algerian people use introduce French sequences in their conversations. The results of our experiments show that it is possible to deal with this task using a global model, on condition that larger corpora are used.

MFCC	13		26		39	
GMM	WER	SER	WER	SER	WER	SER
2	44.6	69.3	44.4	70	45.8	71.8
4	36.1	57.9	35.7	58.7	38.6	62.4
8	30	51.9	30.1	49	31.7	52.5
16	24.1	43	25.9	44.4	27.4	46.2
32	19.9	36.4	22.1	38.9	24.8	42.8
64	17.3	32.2	19.2	34.8	22.9	40.6

Table 8: Performance of the ASR system for code switched sequences (AAD, BAD, FR_{AAD}, FR_{BAD}.

References

Mansour Alghamdi, Fayez Alhargan, Mohammed Alkanhal, Ashraf Alkhairy, Munir Eldesouki, and Ammar Alenazi. 2008. Saudi accented arabic voice bank. *Journal of King Saud University - Computer and Information Sciences*, 20:45–64.

Yousef Ajami Alotaibi, Mansour Alghamdi, and Fahad Alotaiby. 2008. Using a telephony saudi accented arabic corpus in automatic recognition of spoken arabic digits. In *Proceedings of 4th International Symposium on Image/Video Communications over Fixed and Mobile Networks*, pages 43–60. Citeseer.

Barsha Deka, Joyshree Chakraborty, Abhishek Dey, Shikhamoni Nath, Priyankoo Sarmah, SR Nirmala, and Samudra Vijaya. 2018. Speech corpora of under resourced languages of north-east india. In *2018 Oriental COCOSDA-International Conference on Speech Database and Assessments*, pages 72–77. IEEE.

Mourad Djellab, Abderrahmane Amrouche, Ahmed Bouridane, and Noureddine Mehallegue. 2017. Algerian modern colloquial arabic speech corpus (amcasc): regional accents recognition within complex socio-linguistic environments. *Language Resources and Evaluation*, 51(3):613–641.

Sangeeta Ghangam, Daniel Whitenack, and Joshua Nemecek. 2021. Dyn-asr: Compact, multilingual speech recognition via spoken language and accent identification. *arXiv preprint arXiv:2108.02034*.

Boris M Lobanov, Simon V Brickle, Andrey V Kubashin, and Tatiana V Levkovskaja. 1997. An intelligent telephone answering system using speech recognition. In *Fifth European Conference on Speech Communication and Technology*.

Khaled Lounnas, Hassan Satori, Mohamed Hamidi, Hocine Teffahi, Mourad Abbas, and Mohamed Lichouri. 2020. Cliasr: a combined automatic speech recognition and language identification system. In *2020 1st International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*, pages 1–5. IEEE.

Abdulaziz Saleh Mahfoudh Ba Wazir and Joon Huang Chuah. 2019. Spoken arabic digits recognition using deep learning. In *2019 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, pages 339–344.

Askars Salimbajevs. 2018. Creating lithuanian and latvian speech corpora from inaccurately annotated web data. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

H Satori, M Harti, and N Chenfour. 2007. Arabic speech recognition system based on cmusphinx. In *2007 International Symposium on Computational Intelligence and Intelligent Informatics*, pages 31–35. IEEE.

Hassan Satori, Hussein Hiyassat, Mostafa Haiti, and Noureddine Chenfour. 2009. Investigation arabic speech recognition using cmu sphinx system. *International Arab Journal of Information Technology (IAJIT)*, 6(2).

Roneel V Sharan. 2020. Spoken digit recognition using wavelet scalogram and convolutional neural networks. In *2020 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, pages 101–105. IEEE.

Riffat Sharmin, Shantanu Kumar Rahut, and Mohammad Rezwanul Huq. 2020. Bengali spoken digit classification: a deep learning approach using convolutional neural network. *Procedia Computer Science*, 171:1381–1388.

Hamid Tabani, Jose-Maria Arnau, Jordi Tubella, and Antonio González. 2017. Performance analysis and optimization of automatic speech recognition. *IEEE Transactions on Multi-Scale Computing Systems*, 4(4):847–860.

Imre Varga, Stefanie Aalburg, Bernt Andrassy, Sergey Astrov, Josef G Bauer, Christophe Beaugeant, Christian Geißler, and H Hoge. 2002. Asr in mobile phones-an industrial approach. *IEEE transactions on speech and audio processing*, 10(8):562–569.

Su-Lin Wu, Brian Kingsbury, Nelson Morgan, and Steven Greenberg. 1998. Performance improvements through combining phone-and syllable-scale information in automatic speech recognition. In *ICSLP*, volume 1, pages 160–163.

Technical Program

Friday, November 12^{th}, 2021– 09:30 - 17:50 (GMT+1)

09:30 – 10:00	**Opening Session**
10:00 – 12:15	**Oral Session 1 : Speech Recognition** *Chair: Dr. Abed Alhakim Freihat*
	Toshiko Shibano, Xinyi Zhang, Mia Taige Li, Haejin Cho, Peter Sullivan and M. Abdul-Mageed. **Speech Technology for Everyone: Automatic Speech Recognition for Non-Native English.** *Christopher Haberland and Ni Lao.* **Orthographic Transliteration for Kabyle Speech Recognition.** *Anuj Gopal.* **Automated Recognition of Hindi Word Audio Clips for Indian Children using Clustering-based Filters and Binary Classifier.** *Anugunj Naman and Kumari Deepshikha.* **Indic Languages Automatic Speech Recognition using Meta-Learning Approach.** *Khaled Lounnas, Mourad Abbas and Mohamed Lichouri.* **Towards Phone Number Recognition for Code Switched Algerian Dialect.** *Shuang Ao and Xeno Acharya.* **Learning ULMFiT and Self-Distillation with Calibration for Medical Dialogue System.** *Caroline Kendrick, Mariano Frohnmaier and Munir Georges.* **Audio-Visual Recipe Guidance for Smart Kitchen Devices.**
12h20 - 13h30	Break
13:30 – 14:15	**Keynote 1 : Understanding Arabic Transformer Models.** *Dr. Ahmed Abdelali, QCRI*
14:30– 16:10	**Oral Session 2 : Speech Processing.** *Chair: Dr. Ahmed Abdelali*
	Sofia Eleftheriou, Panagiotis Koromilas and Theodoros Giannakopoulos. **Automatic Assessment of Speaking Skills Using Aural and Textual Information.** *Tanvi Dinkar, Beatrice Biancardi and Chloé Clavel.* **From local hesitations to global impressions of a speaker's feeling of knowing.** *Fazia Karaoui, Amar Djeradi and Yves Laprie.* **The Articulatory and acoustics Effects of Pharyngeal Consonants on Adjacent Vowels in Arabic Language.** *Anna Favaro, Licia Sbattella, Roberto Tedesco and Vincenzo Scotti.* **ITAcotron 2: Transfering English Speech Synthesis Architectures and Speech Features to Italian.** *Marco Gaido, Matteo Negri, Mauro Cettolo and Marco Turchi.* **Beyond Voice Activity Detection: Hybrid Audio Segmentation for Direct Speech Translation.**
16:10 – 18:00	**Oral Session 3 : Linguistics Resources.** *Chair: Prof. Hadda Cherroun*
	Ignacio de Toledo Rodriguez, Giancarlo Salton and Robert Ross. **Formulating Automated Responses to Cognitive Distortions for CBT Interactions.** *Hadi Khalilia, Abed Alhakim Freihat and Fausto Giunchiglia.* **The Quality of Lexical Semantic Resources: A Survey.** *Mohamed Amine Cheragui, Abdelhalim Hafedh Dahou and Mohamed Abdelmoazz.* **A3C: Arabic Anaphora Annotated Corpus.** *Carl Strathearn and Dimitra Gkatzia.* **The Task2Dial Dataset: A Novel Dataset for Commonsense enhanced Task-based Dialogue Grounded in Documents.** *Mohamed Lichouri and Mourad Abbas.* **Machine Translation for Zero and Low-resourced Dialects using a New Extended Version of the Dialectal Parallel Corpus (Padic v2.0).**

Saturday, November 13^{th}, 2021– 09:00 - 18:50 (GMT+1)

09:00 – 09:40	**Keynote 2 : Figurative Language Analysis.** *PD Dr. Valia Kordoni, University of Humboldt*
10:00 – 11:40	**Oral Session 4 : Language Model.** *Chair: PD Dr. Valia Kordoni*
	Aviad Rom and Kfir Bar. **Supporting Undotted Arabic with Pre-trained Language Models.** *Hayastan Avetisyan and David Broneske.* **Identifying and Understanding Game-Framing in Online News: BERT and Fine-Grained Linguistic Features.** *Anjali Ragupathi, Siddharth Shanmuganathan and Manu Madhavan.* **Compressive Performers in Language Modelling.** *Danish Mohammed Ebadulla, Rahul Raman, Hridhay Kiran Shetty and Mamatha H.R..* **A Comparative Study on Language Models for the Kannada Language.** *Matteo Muffo, Roberto Tedesco, Licia Sbattella and Vincenzo Scotti.* **Static Fuzzy Bag of Words: a Lightweight and Fast Sentence Embedding Algorithm.**
11:40 – 12:20	**Keynote 3 : AI Technology Commercialization: From Research to Product Innovation.** *Dr. Hussein Al-Natsheh, Beyond Limits*
12h30 - 13h30	**Break**
13:30– 15:30	**Oral Session 5 : Sentiment Analysis.** *Chair: Dr. Mohamed Mediani*
	Gerhard Hagerer, David Szabo, Andreas Koch, Maria Luisa Ripoll Dominguez, Christian Widmer, Maximilian Wich, Hannah Danner and Georg Groh. **End-to-End Annotator Bias Approximation on Crowdsourced Single-Label Sentiment Analysis.** *Akbar Karimi, Leonardo Rossi and Andrea Prati.* **Improving BERT Performance for Aspect-Based Sentiment Analysis.** *Kasturi Bhattacharjee, Rashmi Gangadharaiah and Smaranda Muresan.* **Domain and Task-Informed Sample Selection for Cross-Domain Target-based Sentiment Analysis.** *Mehrdad Nasser, Mohamad Bagher Sajadi and Behrouz Minaei-Bidgoli.* **A Sample-Based Training Method for Distantly Supervised Relation Extraction with Pre-Trained Transformers.** *Aicha Chorana and Hadda Cherroun.* **User Generated Content and Engagement Analysis in Social Media case of Algerian Brands.**
15:10 – 15:50	**Keynote 4 : Bring All Your Features: Arabic Diacritic Recovery Using a Feature-Rich Recurrent Neural Model** *Dr. Kareem Darwish, AiXplain*
16:00 – 18:20	**Oral Session 6 : Language Generation, Text Summarization and Machine Translation.** *Chair: Dr. Kareem Darwish*
	Abdul Waheed, Muskan Goyal, Nimisha Mittal, Deepak Gupta, Ashish Khanna and Moolchand Sharma. **BloomNet: A Robust Transformer based model for Bloom's Learning Outcome Classification.** *Aditeya Baral, Himanshu Jain, Deeksha D and Dr. Mamatha H R.* **MAPLE – MAsking words to generate blackout Poetry using sequence-to-sequence Learning.** *Samira Lagrini and Mohammed Redjimi.* **Use of Rhetorical Relations for Arabic Text Summarization.** *Mourad Abbas and Mohamed Lichouri.* **TPT: An Empirical Term Selection for Arabic Text Categorization.** *Muhammad Saleh Al-Qurishi and Riad Souissi.* **Arabic Named Entity Recognition Using Transformer based-CRF Model.** *Sabine Ullrich and Michaela Geierhos.* **Using Bloom's Taxonomy to Classify Question Complexity.** *Mohamed Amine Menacer, Fatma Ben Hamda, Ghada Mighri, Sabeur Ben Hamidene and Maxime Cariou.* **An interpretable person-job fitting approach based on classification and ranking.** *Pierre Colombo, Chloé Clavel, Chouchang Yack and Giovanna Varni.* **Beam Search with Bidirectional Strategies for Neural Response Generation.**
18h40 - 18h50	**Closing Session**