SocialNLP 2020

The Eighth International Workshop on Natural Language Processing for Social Media

Proceedings of the Conference

July 10, 2020
Online

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 978-1-952148-22-4

SocialNLP 2020@ACL Chairs' Welcome

It is our great pleasure to welcome you to the Eighth Workshop on Natural Language Processing for Social Media-SocialNLP 2018, associated with ACL 2020. SocialNLP is an inter-disciplinary area of natural language processing (NLP) and social computing. We hold SocialNLP twice a year: one in the NLP venue, the other in the associated venue such as those for web technology or artificial intelligence. This year the other version has been successfully held in conjunction with TheWebConf 2020 (formerly WWW), and we are very happily looking forward the NLP version in ACL 2020. The submissions to this year's workshop were still of high quality with the accepted threshold 3.67 (maximum 5), which again leads to a competitive selection process. We received submissions from Asia, Europe, and the United States. After a rigorous review process, we only accepted 8 oral papers and thus the acceptance rate was 40 percent. These exciting papers include novel and practical topics for researchers working on NLP for social media, such as bias mitigation, domain transfer, and dataset constructed for the newly emerged research problems. We believe they will benefit our research community.

Besides the main workshop, we are having this year a new EmotionX challenge, EmotionGIF. At the time we compose this proceedings, we already have 13 international teams registered and the challenge is still ongoing. We have a special session for this challenge to exchange related ideas and experience in the workshop. We hope this challenge series can bring participants from the research problem to the real solution.

This year, we are excited to have Prof. Pascale Fung from Hong Kong University of Science and Technology as our keynote speaker. We encourage attendees to (virtually) attend this keynote speech to have more discussions with outstanding researchers. Prof. Fung not only will talk about their research of misinformation, but will share their winning system in Kaggle. We believe her talk will bring our audience, both for the main workshop and for the challenge, useful insights.

Putting together SocialNLP 2020 was a team effort. We first thank the authors for providing the quality content of the program. We are grateful to the program committee members, who worked very hard in reviewing papers and providing feedback to authors. For a lot of tedious work coming from the challenge, we thank the challenge co-chairs Mr. Boaz Shmueli and Dr. Ming Sun for their great effort. Finally, we especially thank the Workshop Committee Chairs Prof. Milica Gašić, Dr. Dilek Hakkani-Tur, Dr. Saif M. Mohammad, and Dr. Ves Stoyanov for helping us on all the complicated logistics for this year's online version.

We hope you enjoy the workshop!

Organizers
Lun-Wei Ku, Academia Sincia, Taiwan
Cheng-Te Li, National Cheng Kung University, Taiwan

Organizers:

Lun-Wei Ku, Academia Sinica
Chenge-Te Li, National Cheng Kung University

Organizers of EmotionGIF Challenge:

Lun-Wei Ku, Academia Sinica
Boaz Shmueli, Academia Sinica
Ming Sun, Facebook

Program Committee:

Sabine Bergler, Concordia University
Kalina Bontcheva, University of Sheffield
Yung-Chun Chang, Taipei Medical University
Berlin Chen, National Taiwan Normal University
Hsin-Hsi Chen, National Taiwan University
Hai Leong Chieu, DSO National Laboratories
Jinho Choi, Emory University
Nigel Collier, University of Cambridge
Ann Devitt, Trinity College Dublin
MeiXing Dong, University of Michigan
Koji Eguchi, Kobe University
Graeme Hirst, University of Toronto
Hen-Hsen Huang, Natural Taiwan University
David Jurgens, University of Michigan
Pallika Kanani, Oracle Labs
Roman Klinger, University of Stuttgart
June-Jei Kuo, National Chung Hsing University
Tsung-Ting Kuo, University of California, San Diego
Els Lefever, Ghent University
Chuan-Jie Lin, National Taiwan Ocean University
Zhunchen Luo, China Defense Science and Technology Information Center
Bruno Martins, University of Lisbon
Manuel Montes-y-Gómez, INAOE
Scott Nowson, PwC Middle East
Haris Papageorgiou, Athena Research and Innovation Center
Michael Paul, University of Colorado Boulder
Georgios Petasis, NCSR Demokritos
Paolo Rosso, Universitat Politècnica de València
Saurav Sahay, Intel Labs
Yohei Seki, University of Tsukuba
Mário J. Silva, Universidade de Lisboa
Thamar Solorio, University of Houston
Paola Velardi, Sapienza University of Rome
Xiaojun Wan, Peking University

Hsin-Min Wang, Academia Sinica
Jenq-Haur Wang, National Taipei University of Technology
Ingmar Weber, Qatar Computing Research Institute, HBKU
Steven Wilson, University of Edinburgh
Shih-Hung Wu, Chaoyang University of Technology
Liang-Chih Yu, Yuan Ze University
Zhe Zhang, IBM Watson

Keynote Speaker:

Pascale Fung, Hong Kong University of Science and Technology

Table of Contents

Conference Program

July 10, 2020

9:05–9:10 Opening

9:10–10:10 *Keynote: Managing Information and Debunking Misinformation in the Time of Covid-19*
Pascale Fung, Hong Kong University of Science and Technology

10:30–10:50 Coffee Break

10:50–11:30 Technical Session 1

10:50–11:10 *Enhancing Bias Detection in Political News Using Pragmatic Presupposition*
Lalitha Kameswari, Dama Sravani and Radhika Mamidi

11:10–11:30 *Demoting Racial Bias in Hate Speech Detection*
Mengzhou Xia, Anjalie Field and Yulia Tsvetkov

11:30–12:30 Data Session

11:30–11:50 *NARMADA: Need and Available Resource Managing Assistant for Disasters and Adversities*
Kaustubh Hiware, Ritam Dutt, Sayan Sinha, Sohan Patro, Kripa Ghosh and Saptarshi Ghosh

11:50–12:10 *BEEP! Korean Corpus of Online News Comments for Toxic Speech Detection*
Jihyung Moon, Won Ik Cho and Junbum Lee

12:10–12:30 *Stance Prediction for Contemporary Issues: Data and Experiments*
Marjan Hosseinia, Eduard Dragut and Arjun Mukherjee

July 10, 2020 (continued)

12:30–14:00 Lunch

14:00–15:00 EmotionGIF Challenge

15:00–15:30 Coffee Break

15:30–15:50 *Challenges in Emotion Style Transfer: An Exploration with a Lexical Substitution Pipeline*
David Helbig, Enrica Troiano and Roman Klinger

15:50–16:10 *Incorporating Uncertain Segmentation Information into Chinese NER for Social Media Text*
Shengbin Jia, Ling Ding, Xiaojun Chen, Shijia E and Yang Xiang

16:10–16:30 *Multi-Task Supervised Pretraining for Neural Domain Adaptation*
Sara Meftah, Nasredine Semmar, Mohamed-Ayoub Tahiri, Youssef Tamaazousti, Hassane Essafi and Fatiha Sadat

16:30–16:35 Closing Remark

Enhancing Bias Detection in Political News Using Pragmatic Presupposition

Lalitha Kameswari, Dama Sravani, Radhika Mamidi
Language Technologies Research Centre
International Institute of Information Technology
Hyderabad, Telangana, India
{v.a.lalitha, dama.sravani}@research.iiit.ac.in
radhika.mamidi@iiit.ac.in

Abstract

Usage of presuppositions in social media and news discourse can be a powerful way to influence the readers as they usually tend to not examine the truth value of the hidden or indirectly expressed information. Fairclough and Wodak (1997) discuss presupposition at a discourse level where some implicit claims are taken for granted in the explicit meaning of a text or utterance. From the Gricean perspective, the presuppositions of a sentence determine the class of contexts in which the sentence could be felicitously uttered. This paper aims to correlate the type of knowledge presupposed in a news article to the bias present in it. We propose a set of guidelines to identify various kinds of presuppositions in news articles and present a dataset consisting of 1050 articles which are annotated for bias (positive, negative or neutral) and the magnitude of presupposition. We introduce a supervised classification approach for detecting bias in political news which significantly outperforms the existing systems.

1 Introduction

In today's situation where we see several instances of social media being used to interfere with politics in controversial ways, the platforms that have been considered as sources of information are now often seen as politically biased. Especially in newspapers and news websites, sometimes the reporters tend emphasize more on particular view points selectively, and present biased information which is aligned with their personal political ideology. This can lead to widespread alteration of mass political opinion and impact the decision of the voters.

In this paper, we aim to establish a correlation between presupposition and bias in political news articles, and use the knowledge of presupposition to enhance the task of automatic bias detection.

Presuppositions are linguistic tools whose function is to enable us to take some information for granted without actually asserting it. For instance, consider the utterance "Sam will visit California again". This utterance presupposes that Sam has visited California before, and asserts that he will visit once again in future.

Based on their function in discourse, Alcarza (1999) classifies presuppositions into two levels - Semantic and Pragmatic. The propositions which the reader or listener assumes to be true come under the class of Semantic Presuppositions. On the other hand, he defines pragmatic presupposition as "the proposition that a writer or a speaker has taken its truth value for granted in his statement. It consists of previous information about the knowledge, beliefs, ideology and scale of values that the reader or listener must be acquainted with in order to understand the meaning".

The notion of pragmatic presupposition is highly useful in analysing media and political discourse such as news articles and election campaign speeches. Using them in an article or a speech could be an indicator of some hidden intentions and strategies, such as avoiding some key information, or manipulating the audience to focus on certain aspects which favour the speaker by indirectly suggesting that they are true.

Similar to this classification is another popular dichotomy which is widely used for studying implicature as Conventional or Conversational. This idea is extended to the context of presuppositions, based on how they arise (Simons, 2013). Karttunen and Peters (1979) define a presupposition as conventional when the presuppositional content arises due to the properties of lexical items present in a sentence. In their view, "certain lexical items have, in addition to their truth content, a special presuppositional content, which is carried through the compositional process to produce a propositional

Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pages 1–6
Vancouver, Canada, July 30 - August 4, 2017.
https://doi.org/10.18653/v1/P17

presupposition."

On the other hand, there can be some presuppositions which do not contain any lexical triggers. Stalnaker (1974) defines them as Conversational Presuppositions. He suggests that they are the "inferences which are licensed by general conversational principles, in combination with the truth conditions of the presupposing utterance".

2 Related Work

Though there have been several speculations in the linguistic research community about the extra linguistic information provided by the use of presuppositions, very few of them are backed up with proper surveys and observations. The initial direction towards such research was motivated by Van Dijk's idea that in Critical Discourse Analysis, one should closely look at the propositions which in turn suggest some other propositions to be true, but in fact are either not true or controversial. He pointed out some examples from Opinion Discourse (Van Dijk, 1995). For instance, the editorial sections of news usually contain a lot of such propositions which aid in persuading the reader to agree to the given interpretation of some news in the editorial.

Wang (2010) conducted a study on how presuppositions can make newspaper advertisements more effective by compensating for the small place occupied by them. He argued that when an advertisement directs the readers to infer some data which is not directly mentioned, they tend to pay more attention to the product being advertised.

Bekalu (2006) took a small sample of data from 5 newspapers and analysed the use of presuppositions in the articles. He manually analysed how presuppositions can contribute in differentiating between the styles of reporting in the pro-government and anti-government stance of the newspaper.

However, none of these studies have tested the validity of their claims on a large corpus and no computational work has been done in this domain so far.

Moreover, all of the above research was carried out for English news, and there has been little work on Politics and News discourse in Telugu, which is a low resource language. Mukku et al. (2016) applied ML techniques for Sentiment Analysis of Telugu news articles. Kameswari and Mamidi (2018) carried out a case study on political influence through linguistic choices on a corpus of election campaign speeches. Gangula et al. (2019) proposed an attention mechanism to detection of bias in Telugu news articles. To our knowledge there has been no work on presupposition in Telugu till date.

Our research is the first of its kind which proposes guidelines to identify presuppositions in political news and use that information to enhance the computational methods to detect bias in political news articles.

3 Corpus Creation and Annotation

To validate our idea computationally, we need a large dataset of news articles which have been annotated for their bias and magnitude of presupposition. There is no such dataset which captures both the features, so we took the corpus[1] created by Gangula et al. (2019) and modified it. It consists of 1329 articles collected from various newspapers in Telugu, a Dravidian language spoken widely in Telangana and Andhra Pradesh in India. Each article was annotated with a label out of the 6 labels they chose - BJP, TDP, Congress, TRS, YCP and None. The first five labels represent the bias towards or against those parties (marked by "Positive" or "Negative" in their dataset), and "None" denotes that the article is Unbiased.

We created a modified version of the corpus according to our requirement as follows. The original corpus consisted of 218 unbiased articles and 1111 articles which had bias towards some party. Out of those, it was found that some were very short, and some had very little or no mention about any political parties or events. Such articles were filtered out and we were left with 1050 articles of which 850 were biased and 200 were unbiased. Since our main aim was to see the the contribution of presupposition to the biased content in the article, we did not keep the existing labels of "Positive" and "Negative" to denote the direction of bias. All the biased articles were labelled with a bias label of 1 and unbiased articles with 0.

Our annotated corpus[2] is publicly available to ensure reproducibility of the results and to facilitate further research in this domain.

3.1 Annotating for presuppositions

For our purpose, there is a need for a systematic way to identify and quantify the presuppositions

[1] https://bit.ly/2vsUqjk
[2] https://bit.ly/34MqM5Y

in an article. After discussions and observation of several articles, we came up with a novel annotation scheme and guidelines.

3.1.1 Annotation Scheme

Each article is split into individual sentences. Each sentence is given a score of 1 if it contains any pragmatic presupposition which the reader is not expected to know. If no such presuppositions are present, the sentence is given a score of 0. After evaluating all the individual sentences, the score of the article is calculated as the mean of all the individual sentence scores.

3.1.2 Annotation guidelines

To ensure consistency in annotation as well as to capture the linguistic information at both semantic and pragmatic levels, we propose the following annotation guidelines:

1. **Coreference**: If an article contains multiple references to an entity such as a person or an organization, each sentence containing such reference is marked as 1 if it is not expected to be known by the reader or requires additional background information. In other cases, the sentence is marked as 0.

2. **Deixis**: If any person, place, time or discourse deixis is observed in a sentence, we recursively go to the previous reference of the entity in the article. If there is sufficient context in the article to resolve deixis, the sentence is marked as 0. However, if all the previous references are marked as 1, the sentence is marked as 1.

3. **Presence of certain verbal suffixes**: If there is any reference to the events in the past/present or some party policies which were not described and do not fall under the minimum knowledge the reader is expected to have, then the sentence is marked as 1. In Telugu, such references are generally identified by morphological suffixes such as ***-ina, -ani, -tuna, -unTE***, etc.

   e.g: *Dilli lO ErpATu **cEsina** dharnA*
   "The strike **organised** in Delhi"

4. **Verbal Nouns**: If a sentence contains one or more verbs in nominal form indicating change or continuation of state, then it is marked as 1.

   e.g: *telaNGANA dEsam lO agrasthAnam lO **konasAgaDam***
   "Telangana **continuing** being in the first position in the country"

5. **Rhetorical Questions**: If a sentence contains some rhetorical question which is suggestive of some action which is not common knowledge, then the sentence is marked as 1.

   e.g: *rAjakIya padavula kOsam pEdalani ibbandi peTTaDAniki manasu elA vastundi?*
   "How can someone think of troubling the poor for the sake of political power?"

4 Experiments

Our goal is to detect political bias in an article with and without the presupposition information, and compare the results. For this purpose, similar to Gangula et al. (2019), we label Political bias detection as a classification problem. The presence or absence of bias (0 or 1) is treated as the label, and the task is to assign an appropriate label to a news article.

The first step is to represent each article as a vector. Since each vector can be extremely large and sparse, chi-square feature selection algorithm applied, which reduces the size of the vector to 10000.

We performed experiments with the following six classifers:

1. **Bernoulli Naive Bayes**: Naive Bayes (NB) classifier is a probabilistic classifier which uses Bayes Theorem. It evaluates the probability of an event given the probability of another event which has previously occurred. Bernoulli Naive Bayes is a binomial model, particularly useful if the feature vectors are binary (i.e., 0s and 1s).

2. **Multinomial Naive Bayes**: This is similar to the Bernoulli Naive Bayes classifier. It just extends the binomial model to a multinomial model, typically suitable for classification with discrete features.

3. **Logistic Regression**: Logistic regression is a statistical model which is used to estimate the probability of a response based on predictor variables.

Classifier	Multi-nomial NB		Bernoulli NB		Logistic Regression		SVM		Random Forest		MLP	
Category	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Article	83.15	78.54	71.83	72.89	79.73	70.92	77.23	71.38	82.05	76.86	80.81	77.39
Headline + Article	84.98	82.64	74.56	75.23	81.28	73.19	81.26	74.84	84.26	79.98	85.27	81.26
Article + Presupposition	91.22	94.47	85.96	90.44	93.70	96.01	92.26	95.02	90.96	94.96	95.23	95.30
Headline+ Article+ Presupposition	93.56	96.32	89.21	92.01	95.35	97.11	95.87	96.58	92.16	96.25	96.49	95.68

Table 1: Average Accuracy (Acc) in percentage and F1 score for each experiment

4. **Support Vector Machines (SVM)**: SVM is a non probabilistic classifier which constructs a set of hyperplanes in a high-dimensional space separating the data into classes. We implemented SVM with radial basis function as the choice of kernel.

5. **Random Forest Classifier**: Random Forest (RF) is an ensemble of Decision Trees, which are structures that use a tree-like model for the decisions and likely outcomes. Random Forests construct multiple decision trees and take each of their prediction into consideration for giving the final output.

6. **Multi Layer Perceptrons (MLP)**: A multi-layer perceptron (MLP) is a feed-forward artificial neural network model which maps input data sets on an appropriate set of outputs.

For training purpose, Scikit-learn (Pedregosa et al., 2011) implementations have been used for all the classifiers with default hyperparameters. We conducted each experiment four times, each differing in the input given to the classifier. Following are the four categories of inputs which were used:

1. Article
2. Headline + Article
3. Article + Presupposition Value
4. Headline + Article + Presupposition Value

In categories 2, 3 and 4, the entities were concatenated to form a final vector which was given to the classifier. In all the experiments, 10-fold cross validation was carried out. The accuracy and F1 scores for each experiment were calculated.

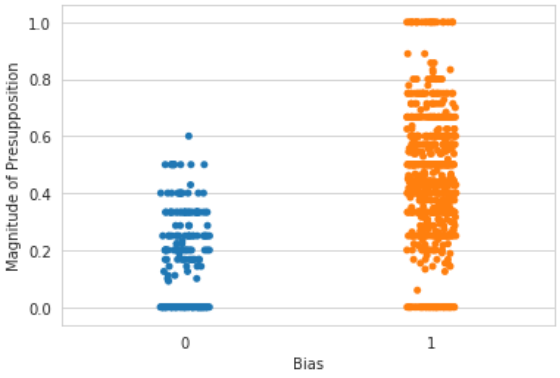


Figure 1: Distribution of the articles in our dataset based on their bias and presupposition values

5 Observations and Results

After assigning a presupposition value to each article, we calculated the mean presupposition value for each category. Biased articles have 0.46 as the mean value of their presupposition, whereas in the case of unbiased articles, it was found to be 0.15. Figure 1 shows us the distribution of articles in our dataset in terms of their bias and presupposition values. It can be see that the density of the articles decreases as we move up in case of unbiased articles, with most of them being in the 0.15 - 0.3 range, and no articles were observed with a value higher than 0.6. On the other hand, there were many biased articles with relatively higher values, most of them in the range 0.4 to 0.7, and the maximum value observed was 1.0. Based on the average value of presupposition and the plot in Figure 1, we assert that biased articles usually tend to have higher presupposition content in them.

The experimental results are shown in Table 1. It can be observed that there is a small improvement whenever the headline is added, when compared

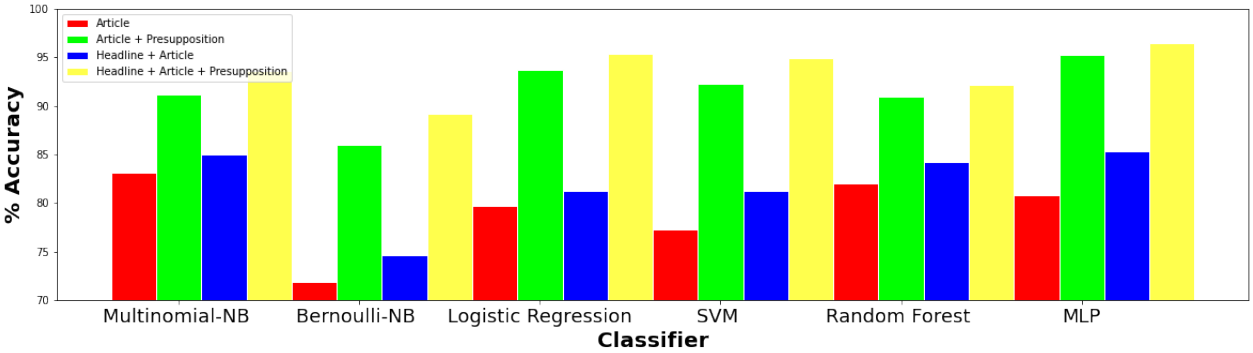


Figure 2: Accuracies of various classifiers for different categories of inputs

to the significant improvement in the performance of each classifier after adding presupposition information. This is seen by observing the difference in performance between categories 1 and 2 and comparing it with the difference in performance between categories 1 and 3 in Table 1. From this, we understand that the knowledge about presupposition contributes more to the detection of bias than the headline. The highest performance for each classifier is observed in category 4, where each classifier has information about the headline, article and the presupposition value.

Figure 2 shows us the performance of each classifiers for the four categories of inputs as discussed in Section 4. The highest performance is achieved by Multi Layer Perceptron classifier with an accuracy of 96.49% and F1 score of 95.68. We can observe that MLP and SVM with RBF Kernel, which are non linear, perform better than all other models. In our task, MLP achieves an improvement of 6.95% over the Attention Network proposed by Gangula et al. (2019), which had the previous best performance in the task of Political Bias detection. This is an example of improving performance by incorporating sophisticated linguistic features, to the point where a simple multilayer perceptron extended with such features performs better than the State-of-the-art Attention based model.

6 Conclusions and Future Work

In this paper, we came up with an interesting correlation between bias and presupposition in news articles. We proved that pragmatic presupposition contributes towards bias in a news article. By using this information, we came up with a supervised method for automatic detection of bias in news articles along with exhaustive guidelines to identify and annotate presuppositions, and a manually annotated dataset to enable further research. The results of our experiments show that our model significantly outperforms all the previous models.

Though we used only news articles for our experiments, our idea is also applicable to other forms of opinion discourse such as Social Media texts, reviews, blogs, etc. where bias in text could lead to spread of misinterpreted information at various levels.

6.1 Future Work

Continuing this work, we plan to come up with an improved scheme for classifying presuppositions into various categories and modified guidelines to annotate accordingly. Subsequently we wish to develop tools to automate the process of presupposition annotation and extend our idea to check whether we can predict the polarity of bias (positive/negative) by the kind of presuppositions present in the text.

We would also like to extend our annotated corpus to accommodate English and other Indian languages by using other corpora such as NELA-GT-2018 (Nørregaard et al., 2019) and come up with better multilingual deep learning models.

Acknowledgements

We would like to thank Kartikey Pant for his insights and helpful suggestions.

References

Enrique Alcarza. 1999. Stylistics in the framework of pragmatics. In *ATAS DEL XXI CONGRESO INTERNACIONAL DE AEDAN*, pages 35–54.

Mesfin Awoke Bekalu. 2006. Presupposition in news discourse. *Discourse & Society*, 17(2):147–172.

Norman Fairclough and Ruth Wodak. 1997. Critical discourse analysis. *Discourse studies: A multidisciplinary introduction*, 2:258–284.

Rama Rohit Reddy Gangula, Suma Reddy Duggenpudi, and Radhika Mamidi. 2019. Detecting political bias in news articles using headline attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 77–84.

Lalitha Kameswari and Radhika Mamidi. 2018. Political discourse analysis: A case study of 2014 andhra pradesh state assembly election of interpersonal speech choices. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*.

Lauri Karttunen and Stanley Peters. 1979. Conventional lmplicature. In *Presupposition*, pages 1–56. Brill.

Sandeep Sricharan Mukku, Nurendra Choudhary, and Radhika Mamidi. 2016. Enhanced sentiment classification of telugu text using ml techniques. *SAAIP@ IJCAI*, 2016:29–34.

Jeppe Nørregaard, Benjamin D Horne, and Sibel Adalı. 2019. Nela-gt-2018: A large multi-labelled news dataset for the study of misinformation in news articles. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 13, pages 630–638.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Mandy Simons. 2013. On the conversational basis of some presuppositions. In *Perspectives on linguistic pragmatics*, pages 329–348. Springer.

Robert Stalnaker. 1974. Pragmatic presupposition//semantics and philosophy. *New York*, pages 197–214.

Teun A Van Dijk. 1990. Discourse & society: a new journal for a new research focus. *Discourse & Society*, 1(1):5–16.

Teun A Van Dijk. 1995. Opinions and ideologies in editorials. In *4th International Symposium of Critical Discourse Analysis, Language, Social Life and Critical Thought, Athens*, pages 14–16.

Ying-fang Wang. 2010. Analysis of presupposition and its function in advertisement. *Canadian Social Science*, 3(4):55–60.

Demoting Racial Bias in Hate Speech Detection

Mengzhou Xia Anjalie Field Yulia Tsvetkov
Language Technologies Institute
Carnegie Mellon University
{mengzhox,anjalief,ytsvetko}@cs.cmu.edu

Abstract

In current hate speech datasets, there exists a high correlation between annotators' perceptions of toxicity and signals of African American English (AAE). This bias in annotated training data and the tendency of machine learning models to amplify it cause AAE text to often be mislabeled as abusive/offensive/hate speech with a high false positive rate by current hate speech classifiers. In this paper, we use adversarial training to mitigate this bias, introducing a hate speech classifier that learns to detect toxic sentences while demoting confounds corresponding to AAE texts. Experimental results on a hate speech dataset and an AAE dataset suggest that our method is able to substantially reduce the false positive rate for AAE text while only minimally affecting the performance of hate speech classification.

1 Introduction

The prevalence of toxic comments on social media and the mental toll on human moderators has generated much interest in automated systems for detecting hate speech and abusive language (Schmidt and Wiegand, 2017; Fortuna and Nunes, 2018), especially language that targets particular social groups (Silva et al., 2016; Mondal et al., 2017; Mathew et al., 2019). However, deploying these systems without careful consideration of social context can increase bias, marginalization, and exclusion (Bender and Friedman, 2018; Waseem and Hovy, 2016).

Most datasets currently used to train hate speech classifiers were collected through crowdsourced annotations (Davidson et al., 2017; Founta et al., 2018), despite the risk of annotator bias. Waseem (2016) show that non-experts are more likely to label text as abusive than expert annotators, and Sap et al. (2019) show how lack of social context in annotation tasks further increases the risk of annotator bias, which can in turn lead to the marginalization of racial minorities. More specifically, annotators are more likely to label comments as abusive if they are written in African American English (AAE). These comments are assumed to be incorrectly labelled, as annotators do not mark them as abusive if they are properly primed with dialect and race information (Sap et al., 2019).

These biases in annotations are absorbed and amplified by automated classifiers. Classifiers trained on biased annotations are more likely to incorrectly label AAE text as abusive than non-AAE text: the false positive rate (FPR) is higher for AAE text, which risks further suppressing an already marginalized community. More formally, the disparity in FPR between groups is a violation of the Equality of Opportunity criterion, a commonly used metric of algorithmic fairness whose violation indicates discrimination (Hardt et al., 2016). According to Sap et al. (2019), the false positive rate for hate speech/abusive language of the AAE dialect can reach as high as 46%.

Thus, Sap et al. (2019) reveal two related issues in the task of hate speech classification: the first is biases in existing annotations, and the second is model tendencies to absorb and even amplify biases from spurious correlations present in datasets (Zhao et al., 2017; Lloyd, 2018). While current datasets can be re-annotated, this process is time-consuming and expensive. Furthermore, even with perfect annotations, current hate speech detection models may still learn and amplify spurious correlations between AAE and abusive language (Zhao et al., 2017; Lloyd, 2018).

In this work, we present an adversarial approach to mitigating the risk of racial bias in hate speech classifiers, even when there might be annotation bias in the underlying training data. In §2, we describe our methodology in general terms, as it can be useful in any text classification task that seeks

Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pages 7–14
Vancouver, Canada, July 30 - August 4, 2017.
https://doi.org/10.18653/v1/P17

to predict a target attribute (here, toxicity) without basing predictions on a protected attribute (here, AAE). Although we aim at preserving the utility of classification models, our primary goal is not to improve the raw performance over predicting the target attribute (hate speech detection), but rather to reduce the influence of the protected attribute.

In §3 and §4, we evaluate how well our approach reduces the risk of racial bias in hate speech classification by measuring the FPR of AAE text, i.e., how often the model incorrectly labels AAE text as abusive. We evaluate our methodology using two types of data: (1) a dataset inferred to be AAE using demographic information (Blodgett et al., 2016), and (2) datasets annotated for hate speech (Davidson et al., 2017; Founta et al., 2018) where we automatically infer AAE dialect and then demote indicators of AAE in corresponding hate speech classifiers. Overall, our approach decreases the dialectal information encoded by the hate speech model, leading to a 2.2–3.2 percent reduction in FPR for AAE text, without sacrificing the utility of hate speech classification.

2 Methodology

Our goal is to train a model that can predict a target attribute (abusive or not abusive language), but that does not base decisions off of confounds in data that result from protected attributes (e.g., AAE dialect). In order to achieve this, we use an adversarial objective, which discourages the model from encoding information about the protected attribute. Adversarial training is widely known for successfully adapting models to learn representations that are invariant to undesired attributes, such as demographics and topics, though they rarely disentangle attributes completely (Li et al., 2018; Elazar and Goldberg, 2018; Kumar et al., 2019; Lample et al., 2019; Landeiro et al., 2019).

Model Architecture Our demotion model consists of three parts: 1) An encoder H that encodes the text into a high dimensional space; 2) A binary classifier C that predicts the target attribute from the input text; 3) An adversary D that predicts the protected attribute from the input text. We used a single-layer bidirectional LSTM encoder with an attention mechanism. Both classifiers are two-layer MLPs with a $\tanh$ activation function.

Training Procedure Each data point in our training set is a triplet $\{(x_i, y_i, z_i); i \in 1 \dots N\}$, where x_i is the input text, y_i is the label for the target attribute and z_i is label of the protected attribute. The (x_i, y_i) tuples are used to train the classifier C, and the (x_i, z_i) tuple is used to train the adversary D.

We adapt a two-phase training procedure from Kumar et al. (2019). We use this procedure because Kumar et al. (2019) show that their model is more effective than alternatives in a setting similar to ours, where the lexical indicators of the target and protected attributes are closely connected (e.g., words that are common in non-abusive AAE and are also common in abusive language datasets). In the first phase (pre-training), we use the standard supervised training objective to update encoder H and classifier C:

$$\min_{C,H} \sum_{i=1}^{N} \mathcal{L}(C(H(x_i)), y_i) \tag{1}$$

After pre-training, the encoder should encode all relevant information that is useful for predicting the target attribute, including information predictive of the protected attribute.

In the second phase, starting from the best-performing checkpoint in the pre-training phase, we alternate training the adversary D with Equation 2 and the other two models (H and C) with Equation 3:

$$\min_{D} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(D(H(x_i)), z_i) \tag{2}$$

$$\min_{H,C} \frac{1}{N} \sum_{i=1}^{N} \alpha \cdot \mathcal{L}(C(H(x_i)), y_i) + (1-\alpha) \cdot \mathcal{L}(D(H(x_i)), 0.5) \tag{3}$$

Unlike Kumar et al. (2019), we introduce a hyper-parameter α, which controls the balance between the two loss terms in Equation 3. We find that α is crucial for correctly training the model (we detail this in §3).

We first train the adversary to predict the protected attribute from the text representations outputted by the encoder. We then train the encoder to "fool" the adversary by generating representations that will cause the adversary to output random guesses, rather than accurate predictions. At the same time, we train the classifier to predict the target attribute from the encoder output.

Dataset	Example
Founta et al. (2018)	I am hungry and I am dirty as hell bruh, need dat shower and dem calories
Blodgett et al. (2016)	so much energy and time wasted hatin on someone when alla that coulda been put towards makin yourself better.... a... https://t.co/awCg1nCt8t

Table 1: Example from Founta et al. (2018) and Blodgett et al. (2016) where the state-of-the-art model misclassifies innocuous tweets (inferred to be AAE) as abusive language. Our model correctly classifies these tweets as non-toxic.

3 Experiments

3.1 Dataset

To the best of our knowledge, there are no datasets that are annotated both for toxicity and for AAE dialect. Instead, we use two toxicity datasets and one English dialect dataset that are all from the same domain (Twitter):

DWMW17 (Davidson et al., 2017) A Twitter dataset that contains 25K tweets annotated as *hate speech*, *offensive*, or *none*. The authors define hate speech as language that is used to expresses hatred towards a targeted group or is intended to be derogatory, to humiliate, or to insult the members of the group, and offensive language as language that contains offensive terms which are not necessarily inappropriate.

FDCL18 (Founta et al., 2018) A Twitter dataset that contains 100K tweets annotated as *hateful*, *abusive*, *spam* or *none*. This labeling scheme was determined by conducting multiple rounds of crowdsourcing to understand how crowdworkers use different labels. Strongly impolite, rude, or hurtful language is considered abusive, and the definition of hate speech is the same as in DWMW17.

BROD16 (Blodgett et al., 2016) A 20K sample out of a 1.15M English tweet corpus that is demographically associated with African American twitter users. Further analysis shows that the dataset contains significant linguistic features of African American English.

In order to obtain dialect labels for the DWMW17 and FDCL18, we use an off-the-shelf demographically-aligned ensemble model (Blodgett et al., 2016) which learns a posterior topic distribution (topics corresponding to African American, Hispanic, White and Other) at a user, message, and word level. Blodgett et al. (2016) generate a AAE-aligned corpus comprising tweets from users labelled with at least 80% posterior probability as using AAE-associated terms. Similarly, following Sap et al. (2019), we assign AAE label to tweets with at least 80% posterior probability of containing AAE-associated terms at the message level and consider all other tweets as Non-AAE.

In order to obtain toxicity labels for the BROD16 dataset, we consider all tweets in this dataset to be non-toxic. This is a reasonable assumption since hate speech is relatively rare compared to the large amount of non-abusive language on social media (Founta et al., 2018).[1]

3.2 Training Parameters

In the pre-training phase, we train the model until convergence and pick the best-performing checkpoint for fine-tuning. In the fine-tuning phase, we alternate training one single adversary and the classification model each for two epochs in one round and train for 10 rounds in total.

We additionally tuned the α parameter used to weight the loss terms in Equation 3 over validation sets. We found that the value of α is important for obtaining text representations containing less dialectal information. A large α easily leads to over-fitting and a drastic drop in validation accuracy for hate speech classification. However, a near zero α severely reduces both training and validation accuracy. We ultimately set $\alpha = 0.05$.

We use the same architecture as Sap et al. (2019) as a baseline model, which does not contain an adversarial objective. For both of this baseline model and our model, because of the goal of demoting the influence of AAE markers, we select the model with the lowest false positive rate on validation set. We train models on both DWMW17 and FDCL18 datasets, which we split into train/dev/test subsets following Sap et al. (2019).

[1] We additionally did a simple check for abusive terms using a list of 20 hate speech words, randomly selected from `Hatebase.org`. We found that the percentage of sentences containing these words is much lower in AAE dataset ($\approx 2\%$) than hate speech datasets ($\approx 20\%$).

Dataset	Accuracy		F1	
	base	ours	base	ours
DWMW17	**91.90**	90.68	75.15	**76.05**
FDCL18	**81.18**	80.27	66.15	**66.80**

Table 2: Accuracy and F1 scores for detecting abusive language. F1 values are macro-averaged across all classification categories (e.g. hate, offensive, none for DWMW17). Our model achieves an accuracy and F1 on par with the baseline model.

	Offensive		Hate	
	base	ours	base	ours
FDCL18-AAE	20.94	**17.69**	3.23	**2.60**
BROD16	16.44	**14.29**	5.03	**4.52**

Table 3: False positive rates (FPR), indicating how often AAE text is incorrectly classified as hateful or abusive, when training with the FDCL18 dataset. Our model consistently improves FPR for offensiveness, and performs slightly better than the baseline for hate speech detection.

4 Results and Analysis

Table 2 reports accuracy and F1 scores over the hate speech classification task. Despite the adversarial component in our model, which makes this task more difficult, our model achieves comparable accuracy as the baseline and even improves F1 score. Furthermore, the results of our baseline model are on par with those reported in Sap et al. (2019), which verifies the validity of our implementation.

Next, we assess how well our demotion model reduces the false positive rate in AAE text in two ways: (1) we use our trained hate speech detection model to classify text inferred as AAE in BROD16 dataset, in which we assume there is no hateful or offensive speech and (2) we use our trained hate speech detection model to classify the test partitions of the DWMW17 and FDCL18 datasets, which are annotated for hateful and offensive speech and for which we use an off-the-shelf model to infer dialect, as described in §3. Thus, for both evaluation criteria, we have or infer AAE labels and toxicity labels, and we can compute how often text inferred as AAE is misclassified as hateful, abusive, or offensive.

Notably, Sap et al. (2019) show that datasets that annotate text for hate speech without sufficient context—like DWMW17 and FDCL18—may suffer from inaccurate annotations, in that annotators

	Offensive		Hate	
	base	ours	base	ours
DWMW17-AAE	**38.27**	42.59	**0.70**	2.06
BROD16	**23.68**	24.34	**0.28**	0.83

Table 4: False positive rates (FPR), indicating how often AAE text is incorrectly classified as hateful or offensive, when training with DWMW17 dataset. Our model fails to improve FPR over the baseline, since 97% of AAE-labeled instances in the dataset are also labeled as toxic.

are more likely to label non-abusive AAE text as abusive. However, despite the risk of inaccurate annotations, we can still use these datasets to evaluate racial bias in toxicity detection because of our focus on FPR. In particular, to analyze false positives, we need to analyze the classifier's predictions of the text as toxic, when annotators labeled it as non-toxic. Sap et al. (2019) suggest that annotators over-estimate the toxicity in AAE text, meaning FPRs over the DWMW17 and FDCL18 test sets are actually lower-bounds, and the true FPR is could be even higher. Furthermore, if we assume that the DWMW17 and FDCL18 training sets contain biased annotations, as suggested by Sap et al. (2019), then a high FPR over the corresponding test sets suggests that the classification model amplifies bias in the training data, and labels non-toxic AAE text as toxic even when annotators did not.

Table 3 reports results for both evaluation criteria when we train the model on the FDCL18 data. In both cases, our model successfully reduces FPR. For abusive language detection in the FDCL18 test set, the reduction in FPR is > 3; for hate speech detection, the FPR of our model is also reduced by 0.6 compared to the baseline model. We can also observe a 2.2 and 0.5 reduction in FPR for abusive speech and hate speech respectively when evaluating on BROD16 data.

Table 4 reports results when we train the model on the DWMW17 dataset. Unlike Table 3, unfortunately, our model fails to reduce the FPR rate for both offensive and hate speech of DWMW17 data. We also notice that our model trained with DWMW17 performs much worse than the model trained with FDCL18 data.

To understand the poor performance of our model when trained and evaluated on DWMW17 data, we investigated the data distribution in the test set and found that the vast majority of tweets

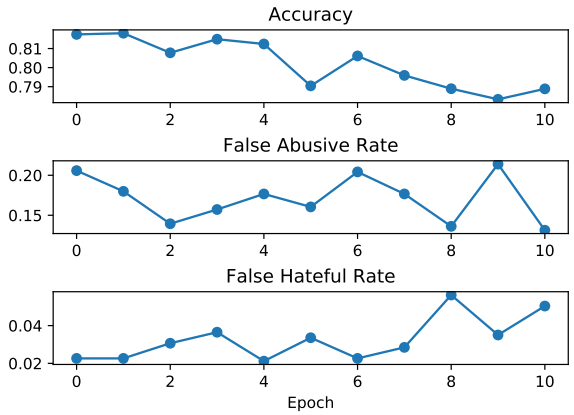


Figure 1: Accuracy of the entire development set of FDCL18 (top), and FPR rate for abusive (middle) and hate (bottom) speech detection for tweets inferred as AAE in the development set. X axis denotes the number of epochs. 0th epoch is the best checkpoint for pre-training step, which is also the baseline model.

labeled as AAE by the dialect classifier were also annotated as toxic (97%). Thus, the subset of the data over which our model might improve FPR consists of merely $< 3\%$ of the AAE portion of the test set (49 tweets). In comparison, 70.98% of the tweets in the FDCL18 test set that were labeled as AAE were also annotated as toxic. Thus, we hypothesize that the performance of our model over the DWMW17 test set is not a representative estimate of how well our model reduces bias, because the improvable set in the DWMW17 is too small.

In Table 1, we provide two examples of tweets that the baseline classifier misclassifies abusive/offensive, but our model, correctly classifies as non-toxic. Both examples are drawn from a toxicity dataset and are classified as AAE by the dialectal prediction model.

Trade-off between FPR and Accuracy In order to better understand model performance, we explored the accuracy and FPR of our model throughout the entire training process. We evaluate the best checkpoint of the pre-trained model (0^{th} epoch) and checkpoints of each epoch during adversarial training and show the results in Figure 1. While the baseline model (0^{th} epoch, before any adversarial training) achieves high accuracy, it also has a high FPR rate, particularly over abusive language. After adversarial training, the FPR rate decreases with only minor changes in accuracy. However, checkpoints with lower FPR rates also often have lower accuracy. While Tables 2 and 3 suggest that our model does achieve a balance between these metrics, Figure 1 shows the difficulty of this task; that is, it is difficult to disentangle these attributes completely.

Eliminatation of protected attribute In Figure 2, we plot the validation accuracy of the adversary through the entire training process in order to verify that our model does learn a text representation at least partially free of dialectal information. Further, we compare using one adversary during training with using multiple adversaries (Kumar et al., 2019). Through the course of training, the validation accuracy of AAE prediction decreases by about 6–10 and 2–5 points for both datasets, indicating that dialectal information is gradually removed from the encoded representation. However, after a certain training threshold (6 epochs for DWMW17 and 8 epochs for FDCL18), the accuracy of the classifier (not shown) also drops drastically, indicating that dialectal information cannot be completely eliminated from the text representation without also decreasing the accuracy of hate-speech classification. Multiple adversaries generally cause a greater decrease in AAE prediction than a single adversary, but do not necessarily lead to a lower FPR and a higher classification accuracy. We attribute this to the difference in experimental setups: in our settings, we focus on one attribute to demote, whereas Kumar et al. (2019) had to demote ten latent attributes and thus required multiple adversaries to stabilize the demotion model. Thus, unlike in (Kumar et al., 2019), our settings do not require multiple adversaries, and indeed, we do not see improvements from using multiple adversaries.

5 Related Work

Preventing neural models from absorbing or even amplifying unwanted artifacts present in datasets is indispensable towards building machine learning systems without unwanted biases.

One thread of work focuses on removing bias at the data level, through reducing annotator bias (Sap et al., 2019) and augmenting imbalanced datasets (Jurgens et al., 2017). Dixon et al. (2018) propose an unsupervised method based on balancing the training set and employing a proposed measurement for mitigating unintended bias in text classification models. Webster et al. (2018) present a gender-balanced dataset with ambiguous name-pair pronouns to provide diversity coverage for real-world data. In addition to annotator bias, sampling

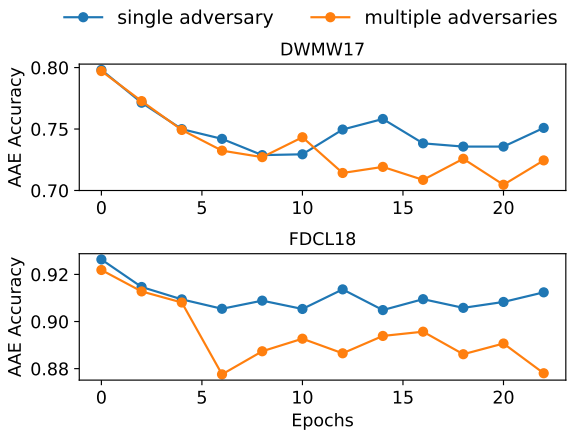


Figure 2: Validation accuracy on AAE prediction of the adversary in the whole training process. The green line denotes the training setting of one adversary and the orange line denotes the training setting of multiple adversaries.

strategies also result in topic and author bias in datasets of abusive language detection, leading to decreased classification performance when testing in more realistic settings, necessitating the adoption of cross-domain evaluation for fairness (Wiegand et al., 2019).

A related thread of work on debiasing focuses at the model level (Zhao et al., 2019). Adversarial training has been used to remove protected features from word embeddings (Xie et al., 2017; Zhang et al., 2018) and intermediate representations for both texts (Elazar and Goldberg, 2018; Zhang et al., 2018) and images (Edwards and Storkey, 2015; Wang et al., 2018). Though previous works have documented that adversarial training fails to obliterate protected features, Kumar et al. (2019) show that using multiple adversaries more effectively forces the removal.

Along similar lines, multitask learning has been adopted for learning task-invariant representations. Vaidya et al. (2019) show that multitask training on a related task e.g., identity prediction, allows the model to shift focus to toxic-related elements in hate speech detection.

6 Conclusion

In this work, we use adversarial training to demote a protected attribute (AAE dialect) when training a classifier to predict a target attribute (toxicity). While we focus on AAE dialect and toxicity, our methodology readily generalizes to other settings, such as reducing bias related to age, gender, or income-level in any other text classification task. Overall, our approach has the potential to improve fairness and reduce bias in NLP models.

7 Acknowledgements

We gratefully thank anonymous reviewers, Maarten Sap, and Dallas Card for their help with this work. The second author of this work is supported by the NSF Graduate Research Fellowship Program under Grant No. DGE1745016. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF. We also gratefully acknowledge Public Interest Technology University Network Grant No. NVF-PITU-Carnegie Mellon University-Subgrant-009246-2019-10-01 for supporting this research.

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.

Su Lin Blodgett, Lisa Green, and Brendan O'Connor. 2016. Demographic dialectal variation in social media: A case study of african-american english. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1119–1130.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Eleventh international aaai conference on web and social media*.

Lucas Dixon, John Li, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2018. Measuring and mitigating unintended bias in text classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 67–73.

Harrison Edwards and Amos Storkey. 2015. Censoring representations with an adversary. *arXiv preprint arXiv:1511.05897*.

Yanai Elazar and Yoav Goldberg. 2018. Adversarial removal of demographic attributes from text data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 11–21.

Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)*, 51(4):85.

Antigoni Maria Founta, Constantinos Djouvas, Despoina Chatzakou, Ilias Leontiadis, Jeremy Blackburn, Gianluca Stringhini, Athena Vakali, Michael Sirivianos, and Nicolas Kourtellis. 2018. Large scale crowdsourcing and characterization of twitter abusive behavior. In *Twelfth International AAAI Conference on Web and Social Media*.

Moritz Hardt, Eric Price, and Nathan Srebro. 2016. Equality of opportunity in supervised learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 3323–3331.

David Jurgens, Yulia Tsvetkov, and Dan Jurafsky. 2017. Incorporating dialectal variability for socially equitable language identification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 51–57.

Sachin Kumar, Shuly Wintner, Noah A Smith, and Yulia Tsvetkov. 2019. Topics to avoid: Demoting latent confounds in text classification. 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 4144–4154.

Guillaume Lample, Sandeep Subramanian, Eric Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2019. Multiple-attribute text rewriting. In *International Conference on Learning Representations*.

Virgile Landeiro, Tuan Tran, and Aron Culotta. 2019. Discovering and controlling for latent confounds in text classification using adversarial domain adaptation. In *Proceedings of the 2019 SIAM International Conference on Data Mining*, pages 298–305. SIAM.

Yitong Li, Timothy Baldwin, and Trevor Cohn. 2018. Towards robust and privacy-preserving text representations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 25–30, Melbourne, Australia. Association for Computational Linguistics.

Kirsten Lloyd. 2018. Bias amplification in artificial intelligence systems. *CoRR*, abs/1809.07842.

Binny Mathew, Ritam Dutt, Pawan Goyal, and Animesh Mukherjee. 2019. Spread of hate speech in online social media. In *Proceedings of the 10th ACM Conference on Web Science*, pages 173–182. ACM.

Mainack Mondal, Leandro Araújo Silva, and Fabrício Benevenuto. 2017. A measurement study of hate speech in social media. In *Proceedings of the 28th ACM Conference on Hypertext and Social Media*, pages 85–94. ACM.

Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A Smith. 2019. The risk of racial bias in hate speech detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1668–1678.

Anna Schmidt and Michael Wiegand. 2017. A survey on hate speech detection using natural language processing. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 1–10, Valencia, Spain. Association for Computational Linguistics.

Leandro Silva, Mainack Mondal, Denzil Correa, Fabrício Benevenuto, and Ingmar Weber. 2016. Analyzing the targets of hate in online social media. In *Tenth International AAAI Conference on Web and Social Media*.

Ameya Vaidya, Feng Mai, and Yue Ning. 2019. Empirical analysis of multi-task learning for reducing model bias in toxic comment detection. *arXiv preprint arXiv:1909.09758*.

Tianlu Wang, Jieyu Zhao, Kai-Wei Chang, Mark Yatskar, and Vicente Ordonez. 2018. Adversarial removal of gender from deep image representations. *arXiv preprint arXiv:1811.08489*.

Zeerak Waseem. 2016. Are you a racist or am i seeing things? annotator influence on hate speech detection on twitter. In *Proceedings of the first workshop on NLP and computational social science*, pages 138–142.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop*, pages 88–93.

Kellie Webster, Marta Recasens, Vera Axelrod, and Jason Baldridge. 2018. Mind the gap: A balanced corpus of gendered ambiguous pronouns. *Transactions of the Association for Computational Linguistics*, 6:605–617.

Michael Wiegand, Josef Ruppenhofer, and Thomas Kleinbauer. 2019. Detection of abusive language: the problem of biased datasets. 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 602–608.

Qizhe Xie, Zihang Dai, Yulun Du, Eduard Hovy, and Graham Neubig. 2017. Controllable invariance through adversarial feature learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 585–596.

Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. 2018. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 335–340.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Ryan Cotterell, Vicente Ordonez, and Kai-Wei Chang. 2019. Gender bias in contextualized word embeddings. 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 629–634.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, and Kai-Wei Chang. 2017. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2979–2989.

NARMADA: Need and Available Resource Managing Assistant for Disasters and Adversities

Kaustubh Hiware*
Mercari, Inc
Tokyo, Japan
hiwarekaustubh@gmail.com

Ritam Dutt*
Indian Institute of Technology
Kharagpur, India
ritam.dutt@gmail.com

Sayan Sinha
Indian Institute of Technology
Kharagpur, India
sayan.sinha@iitkgp.ac.in

Sohan Patro
Microsoft Corporation
Redmond, USA
sopatr@microsoft.com

Kripabandhu Ghosh
Tata Consultancy Services
Pune, India
kripa.ghosh@gmail.com

Saptarshi Ghosh
Indian Institute of Technology
Kharagpur, India
saptarshi@cse.iitkgp.ac.in

Abstract

Although a lot of research has been done on utilising Online Social Media during disasters, there exists no system for a specific task that is critical in a post-disaster scenario – identifying resource-needs and resource-availabilities in the disaster-affected region, coupled with their subsequent matching. To this end, we present NARMADA, a semi-automated platform which leverages the crowd-sourced information from social media posts for assisting post-disaster relief coordination efforts. The system employs Natural Language Processing and Information Retrieval techniques for identifying resource-needs and resource-availabilities from microblogs, extracting resources from the posts, and also matching the needs to suitable availabilities. The system is thus capable of facilitating the judicious management of resources during post-disaster relief operations.

1 Introduction

In recent years, microblogging sites like Twitter and Weibo have played a pivotal role in gathering situational information during disasters or emergency scenarios such as earthquakes, epidemic outbreaks, floods, hurricanes, and so on (Imran et al., 2015; Nazer et al., 2017; Li et al., 2017). Specifically, there are two types of information which are considered useful (or 'actionable') by rescue workers for assisting post-disaster relief operations.[1] These include (i) **Resource needs** that talk about the requirement of a specific resource (such as food, water, shelter) and (ii) **Resource availabilities** that talk about the availability of a specific resource in the region. Some examples of tweets that inform about resource-needs and resource-availabilities, taken from a dataset of tweets related to the 2015 Nepal earthquake, are shown in Table 1. We refer to such tweets as 'needs' and 'availabilities' henceforth.

Needs (excerpts)	**Availabilities (excerpts)**
Mobile phones are not working, no electricity, no **water** in #Thamel, #Nepalquake	Please contact for drinking free service **water** specially for Earthquake Victim. Sanjay Limbu [mobile num]
Over 1400 killed. Many Trapped. **Medical Supplies** Requested.	20,000 RSS personnel with **medical supplies** and other help the first to reach earthquake damaged zones in #Nepal
Nepal earthquake: thousands in need of **shelter** in country little able to cope [url]	can anyone we know pick the 2000 second hand **tents** from Sunauli and distribute it to the people in need in Nepal?

Table 1: Examples of tweets stating resource-needs and tweets stating corresponding matching resource-availabilities, from a dataset of tweets on 2015 Nepal earthquake. The common resources for each pair are shown in boldface (table reproduced from our prior work (Dutt et al., 2019)).

The two major practical challenges faced in this regard include (i) automated *identification* of need and availability posts from social media sites such as Twitter and (ii) automated *matching* of the appropriate needs and availabilities. There have been prior works which have tried to address each of these challenges separately. However, to the best of our knowledge, there exists no system that integrates the two tasks of identifying needs and availabilities and their subsequent matching.

In this work, we present NARMADA (**N**eed and **A**vailable **R**esource **M**anaging **A**ssistant for **D**isasters and **A**dversities), a unified platform for the coordination of relief efforts during disasters by managing the resources that are needed and/or available in the disaster-affected region. NARMADA is designed to be a **semi-automated sys-**

*Equal Contribution

[1] We discussed with relief workers from 'Doctors For You' (http://doctorsforyou.org/) and SPADE (http://www.spadeindia.org/).

Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pages 15–24
Vancouver, Canada, July 30 - August 4, 2017.
https://doi.org/10.18653/v1/P17

tem to ensure supervision and accountability.

In this paper, we describe the Natural Language Processing and Information Retrieval techniques used in NARMADA for the following tasks – (i) identifying resource-needs and resource-availabilities from microblogs, (ii) extracting resource names and other critical information from the posts (e.g., where the resource is needed, the quantity that is needed/available), and (iii) matching the needs to suitable availabilities. The system can be accessed from https://osm-dm-kgp.github.io/Narmada/. Although the system is currently applied over tweets only, NARMADA can also seamlessly integrate information from other sources, as well as enable users to add new information as they deem fit. We believe that the use of this system during a real-time disaster event will help in expediting relief operations.

Our work makes the following contributions.

1) We leverage contextual word embeddings to develop supervised models for automated classification of tweets that inform about need or availability of a resource.

2) We automate the process of categorising the type of resource present in needs and availabilities into food, health, shelter or logistics. This helps us to identify covert information present in tweets.

3) We deploy NARMADA that leverages NLP and IR techniques to identify resource needs and availabilities from microblogs, extract relevant information, and subsequently match needs to suitable availabilities. We believe that such a system would assist in post-disaster relief operations.

2 Related Work

There has been a lot of recent work on utilising Online Social Media (OSM) to facilitate post-disaster relief operations – see (Imran et al., 2015; Nazer et al., 2017; Li et al., 2017) for some recent surveys on this topic. For instance, there have been works on classifying situational and non-situational information (Rudra et al., 2015, 2018), location inferencing from social media posts during disasters (Karimzadeh et al., 2013; Lingad et al., 2013; Paule et al., 2018; Dutt et al., 2018; Kumar and Singh, 2019), early detection of rumours from social media posts (Mondal et al., 2018), emergency information diffusion on social media during crises (Kim et al., 2018), event detection (Hasan et al., 2018), extraction of event-specific informative tweets during disaster (Laylavi et al., 2017) and so on. Tweets specific to particular disasters have been studied in (Gautam et al., 2019), along with their categorisation. Certain other works have focused on the classification of such tweets by determining the probability of them being re-shared in Twitter (Neppalli et al., 2019). A comparison of various learning-based methods has also been recently conducted in (Assery et al., 2019).

Automated retrieval of needs and availabilities have been attempted by employing regular expressions (Purohit et al., 2013), pattern-matching techniques (Temnikova et al., 2015), language models (Basu et al., 2017), and neural IR methods such as word and character embeddings (Basu et al., 2017; Khosla et al., 2017). Likewise, there has been prior research on the automated matching of the needs and availabilities using tf-idf similarity (Purohit et al., 2013) and our prior works (Basu et al., 2018; Dutt et al., 2019) that used word-embeddings for the task. However, no prior work has attempted end-to-end identification and matching of needs and availabilities, which we attempt in this work.

Some information systems have also been implemented for disaster situations such as AIDR (AID, 2015) and Ushahidi (Ush, 2008) which employs crowd-sourced information using social media to assist disaster operations. To our knowledge, none of the existing systems have attempted the specific tasks in this work – identification and matching of resource-needs and resource-availabilities.

3 Dataset

We reuse the dataset made available by our prior works (Khosla et al., 2017; Basu et al., 2018; Dutt et al., 2019) which comprises tweets posted during two disaster events i.e. (i) the earthquake in Nepal in April, 2015 [2], and (ii) the earthquake in central Italy in August, 2016 [3]. Henceforth, we refer to the scenarios as Nepal-quake and Italy-quake.

The tweets were collected using the Twitter Search API[4] with the queries 'nepal quake' and 'italy quake'. The dataset consists of only English tweets since it was observed that most tweets are posted in English to enable rapid communication between international agencies and the local population.

[2] https://en.wikipedia.org/wiki/April_2015_Nepal_earthquake

[3] https://en.wikipedia.org/wiki/August_2016_Central_Italy_earthquake

[4] https://dev.twitter.com/rest/public/search

Removing duplicates and near-duplicates yielded a corpus of *50,068* tweets for Nepal-quake and *70,487* tweets for Italy-quake. However, the number of tweets that inform about needs and availabilities is very low – there are 499 and 1333 need and availability tweets for the Nepal-quake dataset. Likewise, the Italy-quake had only 177 needs and 233 availabilities (see (Dutt et al., 2019) for more details).

4 Methodology

In this section, we describe the methodologies that are incorporated within the NARMADA system. The overarching goal of the system is to facilitate post-disaster relief coordination efforts using the vast information available on social media. To that end, it performs three essential tasks – (i) identifying needs and availabilities, (ii) extracting actionable information from the need and availability tweets, and (iii) matching appropriate needs and availabilities. NARMADA is designed to execute each of the above three tasks in an automated fashion. We elaborate on the specific methodology involved for each of these sub-tasks in the ensuing subsection. However, prior to each of these tasks, we perform pre-processing on the tweet text as follows.

Pre-processing tweets: We employed standard pre-processing techniques on the tweet text to remove URLs (but not email ids), mentions, characters like brackets, 'RT', and other non-ASCII characters like #, &, ellipses and Unicode characters corresponding to emojis. We also segmented CamelCase words and joint alphanumeric terms like 'Nepal2015' into distinct terms ('Nepal' and '2015'). However, we did *not* perform case-folding or stemming on the tweet-text to enable subsequent detection of proper nouns (explained below).

4.1 Identifying needs and availabilities

Identifying needs and availabilities is challenging since they account for only $\approx 3.64\%$ and $\approx 0.58\%$ of the entire Nepal-quake and Italy-quake datasets, respectively. Prior works have approached this problem as a retrieval task using a wide array of techniques such as regular-expressions (Purohit et al., 2013), pattern-matching (Temnikova et al., 2015), language models (Basu et al., 2017), and recently neural IR techniques such as word and character embeddings (Basu et al., 2017; Khosla et al., 2017; Basu et al., 2019).

To enable the real-time deployment, a system needs to filter out tweets on an individual basis. To that end, we decided to adopt a supervised approach for classifying a tweet as 'need', or as 'availability' or as 'others' (i.e., a three-class classification problem). We experimented with different neural architectures for both in-domain and cross-domain classification. In-domain classification implies that the model is trained and tested on tweets related to the same disaster event. On the other hand, cross-domain classification involves training on tweets related to one event (say 'Nepal-quake') and evaluating on tweets related to another event ('Italy-quake') (Basu et al., 2019).

Baseline methods: Convolutional Neural Networks (CNN) have been found to work well in the classification of disaster-related tweets (Caragea et al., 2016; Nguyen et al., 2017). Hence we use the CNN of (Kim, 2014) as a baseline model. We operate on 300-dimensional word-embeddings and fix the feature maps to 100 dimensions. We implement convolutional filters with kernel-size 3, 4, and 5 respectively, with stride 1, and non-linear ReLU activation units. Finally, we apply max-pooling before passing it through a fully-connected layer and softmax with negative log-likelihood (NLL) loss. We experiment with randomly initialized embeddings as well as different kinds of pre-trained embeddings – Glove(Pennington et al., 2014)[5], word2vec (Mikolov et al., 2013) [6], fasttetxt embeddings (Bojanowski et al., 2017)[7] and CrisisNLP embeddings (Imran et al., 2016) trained on tweets posted during many disaster events.

Proposed model: We propose to use a pre-trained BERT model (Devlin et al., 2018) (bert-base-uncased) to represent a tweet as a 768-dimensional embedding. We pass the represented tweet through a fully connected layer which classifies it into the aforementioned three categories. Using BERT pre-trained embeddings helps us in two ways. Firstly, the BERT model itself remains a part of the entire end-to-end system; hence it gets fine-tuned while training. Moreover, BERT uses multiple bidirectional self-attention modules which helps capture contextual information.

In-domain classification: Table 2 notes the per-

[5] https://nlp.stanford.edu/projects/glove/

[6] https://code.google.com/p/word2vec/

[7] https://fasttext.cc/docs/en/english-vectors.html

	Nepal-quake			Italy-quake		
Methodology	Prec	Rec	F1	Prec	Rec	F1
CNN + random	0.803	0.612	0.681	0.926	0.552	0.637
CNN + Glove	0.790	0.668	0.716	0.846	0.680	0.727
CNN + Word2vec	0.796	0.660	0.712	0.847	0.644	0.709
CNN + Fasttext	0.771	0.628	0.683	0.870	0.640	0.703
CNN + CrisisNLP	0.767	0.634	0.682	0.734	0.585	0.635
BERT (proposed for F1)	0.786	0.866	0.823	0.856	0.722	0.779
BERT (proposed for Rec)	0.791	0.872	**0.828**	0.843	0.810	**0.826**

Table 2: Performance of the neural architectures for in-domain classification of tweets into three classes – needs, availabilities, and others. Best F1-scores in boldface.

formances of the various classification models in *in-domain settings*, averaged over both the classes *needs* and *availabilities*. For each of the two datasets, we consider 20% (randomly sampled) of the labelled data as the test set, 70% of the rest as the training set, and the rest 10% was used as the validation set. We report the Precision, Recall and F1-score on the test set as the evaluation measures. We consider F1-score as the primary score since it incorporates both Precision and Recall. The proposed BERT model outperforms all other models in terms of F1-score.

We trained two versions of our proposed BERT model – (i) one version was trained to optimise the F1-score (our primary measure) on the validation set, and (ii) the second version was trained to optimise the *Recall* on the validation set. We specifically tried one version to optimise Recall, since it is usually considered important to identify all needs and availabilities in a disaster situation. As seen in Table 2, both versions of the model achieved comparable performance on the Nepal-quake dataset (F1-scores of 0.823 and 0.828). But the version trained for optimizing Recall achieved substantially higher performance on the Italy-quake dataset where needs and availabilities are much sparser. This improved performance justifies our decision of focusing on improving Recall.

Cross-domain classification: In a cross-domain setting, the model is trained on tweets of one event and then evaluated on tweets of the other event. We compare the performance of the BERT model against the best-supervised model ('Best-SM') of (Basu et al., 2019), which is a CNN classifier initialised with CrisisNLP embeddings. Table 3 shows results when the models are trained on Italy-quake tweets and tested on Nepal-quake tweets. Similarly, Table 4 shows the opposite setting, i.e., the models are trained on Nepal-quake tweets and tested on Italy-quake tweets. In both the cases, we use the BERT model optimised for F1-score, as described above. Even for cross-domain performance, we see that the BERT model outperforms the CNN-based baseline of (Basu et al., 2019).

Method	P@100	R@100	F1@100
Needs			
Best-SM (Basu et al., 2019)	0.443	0.044	0.080
BERT (proposed)	0.320	0.066	**0.110**
Availabilities			
Best-SM (Basu et al., 2019)	0.533	0.019	0.037
BERT (proposed)	0.500	0.038	**0.070**

Table 3: Performance of the neural architectures when trained on Italy-quake and tested on Nepal-quake. Best F1-scores in boldface.

Method	P@100	R@100	F1@100
Needs			
Best-SM (Basu et al., 2019)	0.198	0.056	0.087
BERT (proposed)	0.32	0.184	**0.234**
Availabilities			
Best-SM (Basu et al., 2019)	0.216	0.046	0.076
BERT (proposed)	0.28	0.121	**0.168**

Table 4: Performance of the neural architectures when trained on Nepal-quake and tested on Italy-quake. Best F1-scores in boldface.

4.2 Extracting relevant fields from needs and availabilities

Prior discussions with relief workers helped us identify the following five fields that are deemed relevant in coordinating the relief efforts, namely: (i) resource – which items are needed/available, (ii) quantity – how much of each resource is needed/available, (iii) location – where is the resource needed/available, (iv) source – who needs the resource or who is offering, and (v) contact – how to contact the said source.

We adapt the unsupervised methodology of our prior work (Dutt et al., 2019) to extract the relevant fields from needs and availabilities. We sought to incorporate this technique due to the paucity of labelled instances which discourages a supervised machine learning approach (and because gathering many labelled instances is difficult in a disaster scenario). Moreover, the unsupervised approach was shown to be generalizable across several datasets (Dutt et al., 2019). We describe the adapted methodology in this section.

Unsupervised resource extraction: We start by giving a brief description of the methodology in (Dutt et al., 2019). We perform dependency parsing on the text to obtain a Directed Acyclic

Tweet Text	Resource
villagers in the remote community of ghyangphedi fear *hunger* and *#starvation*	food
earthquake victims *sleeping outside* in nepal	shelter
people are *shivering in the cold*	shelter
free calls to italy in the wake of earthquake	logistics

Table 5: Examples of covert tweets and the corresponding resource class assigned to the tweet by our BERT-based resource classifier.

Graph (DAG). We compile an initial list of headwords (*send, need, donate*, etc.) which consists of the verbs in the query-set and the ROOT word of the DAG. We have identified specific characteristics of the child nodes of the headwords that enable us to label the node as a potential resource.

For example, if a word w is tagged as a NOUN and is the direct object of the 'donates', w can be expected to be a potential resource. We have also identified dependency rules, that increases the list of head-words to improve our recall. We thus obtain a list of potential resources after dependency parsing. We then verify these potential resources by checking for the semantic similarity of the extracted words with a pre-compiled list of resources commonly used during disasters. The resource list is obtained from several reputed sources like UN-OCHA[8], UNHCR[9] and WHO[10]. This pre-compiled list also enables us to categorise the resources into four classes namely *food* (bottled water, biscuits, rice), *health* (blood, medicine, latrines), *shelter* (tents, blankets, tarpaulins), and *logistics* (electricity, helicopters, cash).

Adapting the method to deal with covert tweets: One of the limitations of the unsupervised methodology in (Dutt et al., 2019) is the inability to glean relevant information from *covert tweets* where the resource needed/available is not mentioned explicitly. We illustrate instances of such covert tweets in Table 5. Since the resource name is not explicitly stated in the tweet-text, the methodology in (Dutt et al., 2019) cannot identify the resources for such tweets.

To circumvent this problem, we again use the pre-trained BERT model (Devlin et al., 2018) to encode a tweet. We pass this representation through a linear layer and perform multi-label classification into the aforementioned four categories, i,e. food, health, shelter and logistics. We use multi-label classification since a particular tweet can mention multiple resources. This adaptation helps the methodology to correctly classify many of the covert tweets, as demonstrated in Table 5 (the second column shows the resource-class that is assigned by our methodology).

[8] https://www.unocha.org/
[9] https://www.unhcr.org/
[10] https://www.who.int/

Dataset	Precision	Recall	F1-score
Nepal-quake	0.838	0.882	0.843
Italy-quake	0.825	0.858	0.823

Table 6: Performance of the multi-label BERT-based resource classifier on in-domain classification.

Method	**P@100**	**R@100**	**F1@100**
Nepal-quake			
USM (Dutt et al., 2019)	0.623	0.833	**0.685**
BERT (trained on Italy)	0.484	0.670	0.522
BERT (trained on Italy + 5% Nepal)	0.636	0.834	0.680
Italy-quake			
USM (Dutt et al., 2019)	0.487	0.595	0.516
BERT (trained on Nepal)	0.798	0.862	**0.808**

Table 7: Comparing the BERT-based resource classifier with the unsupervised methodology (USM) of (Dutt et al., 2019) in cross-domain setting. Best F1-scores in boldface.

We report the *in-domain* classification performance of our BERT-based resource classifier for the Nepal-quake and Italy-quake datasets in Table 6. We test on 20% of the data (sampled randomly) and train on the remaining 70% while using 10% for validation. We optimise the model with the highest macro F1-score on the validation set.

Next, we compare the performance of the proposed BERT-based resource classifier with that of the unsupervised methodology of (Dutt et al., 2019) (which we refer to as 'USM'). To ensure a fair comparison, we perform this comparison in a *cross-domain* setting wherein we train the supervised model on one dataset (e.g., Nepal-quake) and evaluate on another (e.g., Italy-quake). We present the results of this comparison in Table 7.

We observe from Table 7 that the BERT resource classifier trained on Nepal-quake significantly outperforms USM over the Italy-quake dataset (F1-score of 0.808 for the BERT method and 0.516 for USM). In contrast, the BERT resource classifier when trained on Italy-quake yielded significantly poorer results on Nepal-quake dataset than USM. However, training only on an additional 5% of labelled instances of the Nepal-quake dataset, demonstrated comparative performance (F1-score of 0.680 for the BERT method and 0.685 for USM). The reason for these performances is as follows. The Italy-quake dataset does not contain mention of several amenities that are heavily prevalent in

Tweet text (excerpts)	Resource	Location	Quantity	Source	Contact
Urgent need of analgesic,antibiotics, betadiene, swabs in kathmandu!! Call for help 98XXX-XXXXX #earthquake #Nepal #KTM (N)	analgesic, antibiotics, betadiene, swabs	kathmandu, ktm, nepal			98XXX-XXXXX
India sends 39 #NDRF team, 2 dogs and 3 tonnes equipment to Nepal Army for rescue operations: Indian Embassy in #Nepal (A)	NDRF team, dogs,	nepal	dogs - 2, NDRF team - 39	India	
Visiting Sindhupalchok devastating earthquake highly affected district . Delivery Women in a tent . No water no toilet (N)	tent, delivery women, water	Sindhupalchok			
Rajasthan Seva Samiti donates more than 800 tents to Nepal Earthquake victims (A)	tents		tents-800	Rajasthan Seva Samiti	

Table 8: Examples of information extracted from need (N) and availability (A) tweets by the methodologies proposed in this work. Red colour indicates wrongly extracted information.

the Nepal-quake dataset, but *not* vice-versa. This difference is mainly because the Italy earthquake was a comparatively mild one in a developed region, and hence not many resources were needed; in contrast, the Nepal earthquake was a severe one in a developing region, and a lot of resources were needed in Nepal. Hence the Nepal-quake dataset contains mention of far more varied resources, as compared to the Italy-quake dataset.

Thus, including the BERT-based resource classifier in addition to the unsupervised methodology improves resource extraction performance, and also lends generalisability across different datasets.

Extracting Locations: We extract geographical locations from the tweet text using the methodology in our prior work (Dutt et al., 2018). First, we apply several unsupervised techniques to extract a set of potential locations. These techniques include (i) segmentating hashtags, (ii) disambiguating proper nouns from parse trees, (iii) identifying phrases with regex matches, (iv) dependency parsing to locate nouns close from words in query-set in the DAG, and (v) employing pre-trained Named Entity Recognizers [11] to identify words tagged as geographical location. Next, we verify these potential locations using a gazetteer. We consider those locations to be valid only if their geospatial coordinates lie within the boundary of the affected region (e.g., Nepal or Italy). We used two gazetteers namely Geonames [12] and Open Street Map [13] to identify locations with varying levels of granularity (as detailed in (Dutt et al., 2018)).

Extracting the source: We consider as viable sources two types of words – (i) proper nouns that are tagged as organisations, persons or geographical locations by a Named Entity Recognizer, and (ii) proper nouns that are child nodes of dependency parsing – provided they have not been identified previously as 'location' or 'resources' during the verification phase. See our prior work (Dutt et al., 2019) for details of the methodology.

Extracting Quantity: For each resource extracted, we identify whether it is preceded by a numeric token. The numeric token may be the orthographic notation of a number (e.g., '100') or may semantically represent a number (e.g., 'hundred'). We assign the numeric token as the quantity of the particular resource.

Extracting Contact: We use regular expressions to identify contacts corresponding to email-ids and phone numbers.

The performance of our information extraction methods (in terms of precision, recall and F1-score) was similar to what is presented in (Dutt et al., 2019). In our experiments, we obtained F1-scores of 0.89, 0.91, 0.76, 0.58 and 1.00 for identifying Resources, Location, Quantity, Source and Contact respectively, for need-tweets. Likewise, the F1-scores for availability-tweets were 0.85, 0.85, 0.84, 0.65 and 1.00 respectively. Table 8 shows some examples of the fields extracted by our methods from some need-tweets and availability-tweets.

4.3 Matching needs and availabilities

We propose a fast and real-time algorithm for matching needs and availabilities based on **proportion of common resources**. Specifically, for a given need-tweet, we compute the match with a particular availability-tweet as the fraction of the resources extracted from the need-tweet, that are also present in the availability-tweet. For the given

[11] We use the inbuilt NER tool of SpaCy (https://spacy.io/)

[12] http://www.geonames.org/

[13] http:420//geocoder.readthedocs.io/providers/OpenStreetMap.html

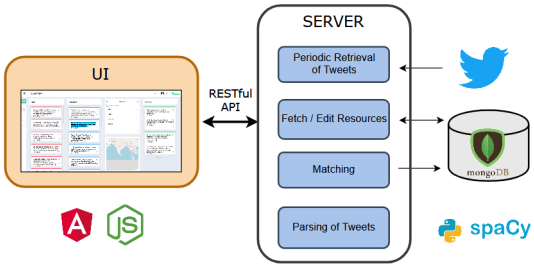


Figure 1: NARMADA's architecture overview

need tweet, availability-tweets are ranked in decreasing order of the fraction of common resources (ties resolved arbitrarily).

We also experiment with some baseline methodologies, namely using common nouns (Basu et al., 2018), tf-idf vectors of the tweet text (Purohit et al., 2013) and local word embeddings of the tweet (Basu et al., 2018). Our methodology (based on the proportion of common resources) obtains an F1-score of 0.84 for Nepal-quake and an F1-score of 0.87 for Italy-quake dataset respectively, which is competitive with the performance of the baselines.

This section described the NLP and IR techniques used in NARMADA. The next section describes the system architecture.

5 System Architecture

The high-level system architecture for NARMADA is shown in Figure 1. The system can be accessed from https://osm-dm-kgp.github.io/Narmada/, where further details and a demonstration video are also provided. NARMADA is designed and built for the Web, thus not restricting it to any particular operating system or browser type, allowing cross-platform (desktop/mobile) functionality.

5.1 User Interface

The user interface has been designed in Typescript using Angular, a popular web-application framework. ngx-admin[14] was used as a boilerplate for front-end components. The interface has been designed to be intuitive, yet presenting as much information as possible without overcrowding. A detailed note is available at https://osm-dm-kgp.github.io/Narmada/.

The user interface comprises a dashboard (shown in Figure 2) that acts as a landing page. Besides providing an initial view of active needs and availabilities (at the present point of time), it displays matched resources. The user is provided with various options to make it easy to search and locate resources as well as highlight items as deemed necessary.

An alternate section is available where users can enter new needs/availabilities manually. The class labels of the information are detected automatically, but the user is allowed to modify the same. Another section for "Completed matches" is to be used for logging completed or exhausted needs and resources. A user manual is also attached to the UI.

5.2 Server

The major services provided by the backend server include classification and categorisation of the tweets in the system. It also provides support for the addition of new information and their automatic categorisation. Facilities have been provided for marking resources once their need is fulfilled or the availability gets exhausted.

The server side uses NodeJS framework and is written in Javascript. Nginx is used as an HTTP server to make the frontend accessible to the public. However, the NLP-related extraction tasks are handled better in Python. The server partly uses a Flask-based Python backend, a micro web framework. The Flask server makes API calls to the deep learning classifiers, featuring BERT, which returns the output. The output is further reflected in the frontend. The server sends information requested by the user interface via *RESTful API*, which supports cached responses on the frontend and enables the system to be scalable, thus allowing more users to use this service. API endpoints are publicly available, which would allow programmatic access to the server's functionalities (see https://osm-dm-kgp.github.io/Narmada/).

6 Discussion

NARMADA intends to assist in crossing the initial barrier in identifying and matching needs and availabilities from social media during the occurrence of a disaster. In practice, it becomes necessary for other service providers to be triggered in order to make sure that the needs are addressed, by proper collection, transportation and provisioning of the matched resources deemed to be available. For instance, the needs and availabilities could be marked on a map, with each type of resource be-

[14] https://github.com/akveo/ngx-admin

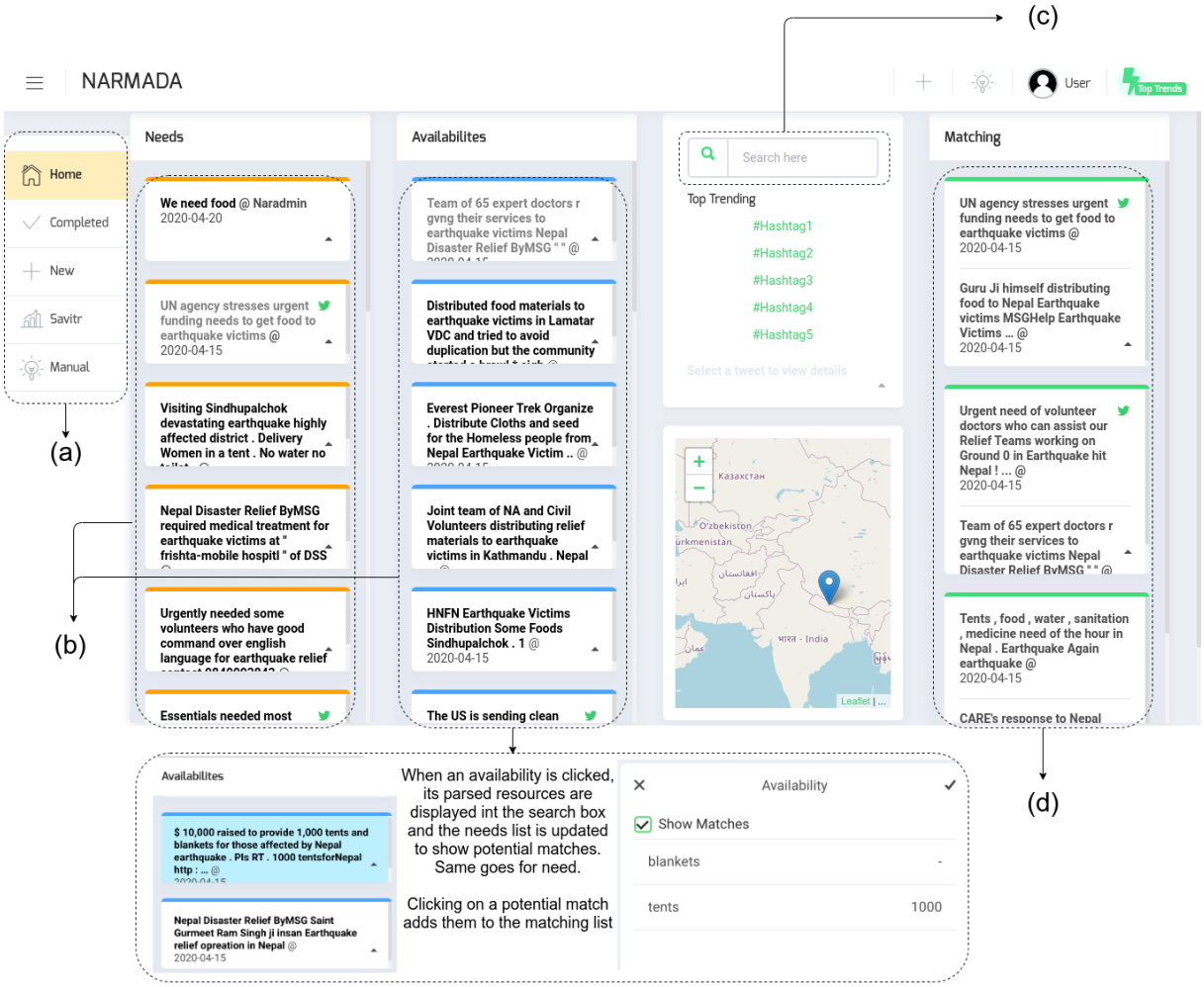


Figure 2: Dashboard of NARMADA – (a) **Navigation Buttons**. (b) **Needs and Availabilities List**: tweets are displayed in reverse chronological order; gray tweet: already matched; black tweet: unmatched; each tweet contains a notch at the bottom-right corner, clicking on which reveals more details. (c) **Search Box**: when a query is entered, the needs and availabilities containing the query-phrase are displayed. (d) **Matching List**: displays the matched needs and availabilities; clicking a matching displays its resources, and gives the user an option to mark it as completed.

ing represented with a different symbol, making it easy to physically locate them. Local volunteers might be provided with a mobile app to help them find nearby needs and availabilities. Misinformation in twitter is common (Bal et al., 2020). The volunteers would also need a facility to confirm that the posted needs and availabilities are indeed genuine, concerning various parameters such as quantity (since at times of disasters, needs may be exaggerated).

7 Conclusion and Future Work

We proposed a system NARMADA for resource management during a disaster situation. Though the system is developed to work across posts from various social media platform, this research focused on data from Twitter. The real-time nature and easy access to large volumes of information provided by Twitter have made it a lucrative choice for disaster analytics.

Currently, the system allows all users to perform any action on the system. One future task would be to implement a login system that would allow different access-levels to different users. For instance, a visitor would be able to only view and query information, a volunteer would be able to add new resources, mark a need as matched, etc., while a system administrator would have rights to undo all actions of all users, etc. The current system does not allow multiple volunteers to communicate within the platform over a resource, which we wish to incorporate in the future. We also plan to incorporate support for vernacular languages, provided the requisite tools are available.

References

2008. Ushahidi. https://www.ushahidi.com/.

2015. Aidr (artificial intelligence for disaster response). http://aidr.qcri.org/.

Nasser Assery, Yuan Xiaohong, Sultan Almalki, Roy Kaushik, and Qu Xiuli. 2019. Comparing learning-based methods for identifying disaster-related tweets. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pages 1829–1836. IEEE.

Rakesh Bal, Sayan Sinha, Swastika Dutta, Risabh Joshi, Sayan Ghosh, and Ritam Dutt. 2020. Analysing the extent of misinformation in cancer related tweets. *arXiv preprint arXiv:2003.13657*.

Moumita Basu, Kripabandhu Ghosh, Somenath Das, Ratnadeep Dey, Somprakash Bandyopadhyay, and Saptarshi Ghosh. 2017. Identifying post-disaster resource needs and availabilities from microblogs. In *Proc. ASONAM*.

Moumita Basu, Anurag Shandilya, Kripabandhu Ghosh, and Saptarshi Ghosh. 2018. Automatic matching of resource needs and availabilities in microblogs for post-disaster relief. In *Comp. Proc. WWW 2018 2018*, pages 25–26.

Moumita Basu, Anurag Shandilya, Prannay Khosla, Kripabandhu Ghosh, and Saptarshi Ghosh. 2019. Extracting resource needs and availabilities from microblogs for aiding post-disaster relief operations. *IEEE Transactions on Computational Social Systems*, 6(3):604–618.

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.

Cornelia Caragea, Adrian Silvescu, and Andrea H Tapia. 2016. Identifying informative messages in disaster events using convolutional neural networks. In *International Conference on Information Systems for Crisis Response and Management*, pages 137–147.

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*.

Ritam Dutt, Moumita Basu, Kripabandhu Ghosh, and Saptarshi Ghosh. 2019. Utilizing microblogs for assisting post-disaster relief operations via matching resource needs and availabilities. *Information Processing & Management*, 56(5):1680–1697.

Ritam Dutt, Kaustubh Hiware, Avijit Ghosh, and Rameshwar Bhaskaran. 2018. Savitr: A system for real-time location extraction from microblogs during emergencies. In *Proc. WWW Workshop SMERP*.

Akash Kumar Gautam, Luv Misra, Ajit Kumar, Kush Misra, Shashwat Aggarwal, and Rajiv Ratn Shah. 2019. Multimodal analysis of disaster tweets. In *2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM)*, pages 94–103. IEEE.

Mahmud Hasan, Mehmet A. Orgun, and Rolf Schwitter. 2018. Real-time event detection from the Twitter data stream using the TwitterNews+ Framework. *Information Processing & Management*. Online: https://doi.org/10.1016/j.ipm.2018.03.001.

Muhammad Imran, Carlos Castillo, Fernando Diaz, and Sarah Vieweg. 2015. Processing Social Media Messages in Mass Emergency: A Survey. *ACM Computing Surveys*, 47(4):67:1–67:38.

Muhammad Imran, Prasenjit Mitra, and Carlos Castillo. 2016. Twitter as a lifeline: Human-annotated twitter corpora for nlp of crisis-related messages. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA).

Morteza Karimzadeh, Wenyi Huang, Siddhartha Banerjee, Jan Oliver Wallgrün, Frank Hardisty, Scott Pezanowski, Prasenjit Mitra, and Alan M. MacEachren. 2013. Geotxt: A web api to leverage place references in text. In *Proceedings of the 7th Workshop on Geographic Information Retrieval*, pages 72–73.

Prannay Khosla, Moumita Basu, Kripabandhu Ghosh, and Saptarshi Ghosh. 2017. Microblog retrieval for post-disaster relief: Applying and comparing neural IR models. *CoRR*, abs/1707.06112.

Jooho Kim, Juhee Bae, and Makarand Hastak. 2018. Emergency information diffusion on online social media during storm Cindy in U.S. *International Journal of Information Management*, 40:153 – 165.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Abhinav Kumar and Jyoti Prakash Singh. 2019. Location reference identification from tweets during emergencies: A deep learning approach. *International journal of disaster risk reduction*, 33:365–375.

Farhad Laylavi, Abbas Rajabifard, and Mohsen Kalantari. 2017. Event relatedness assessment of Twitter messages for emergency response. *Information Processing & Management*, 53:266–280.

Tao Li, Ning Xie, Chunqiu Zeng, Wubai Zhou, Li Zheng, Yexi Jiang, Yimin Yang, Hsin-Yu Ha, Wei Xue, Yue Huang, Shu-Ching Chen, Jainendra Navlakha, and S. S. Iyengar. 2017. Data-Driven Techniques in Disaster Information Management. *ACM Comput. Surv.*, 50(1):1:1–1:45.

John Lingad, Sarvnaz Karimi, and Jie Yin. 2013. Location extraction from disaster-related microblogs. In *Proceedings of the 22nd international conference on world wide web*, pages 1017–1020. ACM.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Tamal Mondal, Prithviraj Pramanik, Indrajit Bhattacharya, Naiwrita Boral, and Saptarshi Ghosh. 2018. Analysis and Early Detection of Rumors in a Post Disaster Scenario. *Information Systems Frontiers*, 20(5).

Tahora H. Nazer, Guoliang Xue, Yusheng Ji, and Huan Liu. 2017. Intelligent disaster response via social media analysis a survey. *SIGKDD Explor. Newsl.*, 19(1):46–59.

Venkata Kishore Neppalli, Cornelia Caragea, Doina Caragea, Murilo Cerqueira Medeiros, Andrea H Tapia, and Shane E Halse. 2019. Predicting tweet retweetability during hurricane disasters. In *Emergency and Disaster Management: Concepts, Methodologies, Tools, and Applications*, pages 1277–1298. IGI Global.

Dat Tien Nguyen, Kamla Al-Mannai, Shafiq R Joty, Hassan Sajjad, Muhammad Imran, and Prasenjit Mitra. 2017. Robust classification of crisis-related data on social networks using convolutional neural networks. In *ICWSM*, pages 632–635.

Jorge David Gonzalez Paule, Yeran Sun, and Yashar Moshfeghi. 2018. On fine-grained geolocalisation of tweets and real-time traffic incident detection. *Information Processing & Management*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proc. EMNLP*.

Hemant Purohit, Carlos Castillo, Fernando Diaz, Amit Sheth, and Patrick Meier. 2013. Emergency-relief coordination on social media: Automatically matching resource requests and offers. *First Monday*, 19(1).

Koustav Rudra, Niloy Ganguly, Pawan Goyal, and Saptarshi Ghosh. 2018. Extracting and summarizing situational information from the twitter social media during disasters. *ACM Trans. Web*, 12(3):17:1–17:35.

Koustav Rudra, Subham Ghosh, Pawan Goyal, Niloy Ganguly, and Saptarshi Ghosh. 2015. Extracting situational information from microblogs during disaster events: A classification-summarization approach. In *Proc. CIKM*.

Irina Temnikova, Carlos Castillo, and Sarah Vieweg. 2015. EMTerms 1.0: A Terminological Resource for Crisis Tweets. In *Proc. ISCRAM*.

BEEP! Korean Corpus of Online News Comments for Toxic Speech Detection

Jihyung Moon[*,†,1], **Won Ik Cho**[*,2], **Junbum Lee**[3]
Department of Industrial Engineering[1],
Department of Electrical and Computer Engineering and INMC[2],
Graduate School of Data Science[3],
Seoul National University, Seoul
{ans1107,tsatsuki,beomi}@snu.ac.kr

Abstract

Toxic comments in online platforms are an unavoidable social issue under the cloak of anonymity. Hate speech detection has been actively done for languages such as English, German, or Italian, where manually labeled corpus has been released. In this work, we first present 9.4K manually labeled entertainment news comments for identifying Korean toxic speech, collected from a widely used online news platform in Korea. The comments are annotated regarding social bias and hate speech since both aspects are correlated. The inter-annotator agreement Krippendorff's alpha score is 0.492 and 0.496, respectively. We provide benchmarks using CharCNN, BiLSTM, and BERT, where BERT achieves the highest score on all tasks. The models generally display better performance on bias identification, since the hate speech detection is a more subjective issue. Additionally, when BERT is trained with bias label for hate speech detection, the prediction score increases, implying that bias and hate are intertwined. We make our dataset publicly available and open competitions with the corpus and benchmarks.

1 Introduction

Online anonymity provides freedom of speech to many people and lets them speak their opinions in public. However, anonymous speech also has a negative impact on society and individuals (Banks, 2010). With anonymity safeguards, individuals easily express hatred against others based on their superficial characteristics such as gender, sexual orientation, and age (ElSherief et al., 2018). Sometimes the hostility leaks to the well-known people who are considered to be the representatives of targeted attributes.

Recently, Korea had suffered a series of tragic incidents of two young celebrities that are presumed to be caused by toxic comments (Fortin, 2019; McCurry, 2019a,b). Since the incidents, two major web portals in Korea decided to close the comment system in their entertainment news aggregating service (Yeo, 2019; Yim, 2020). Even though the toxic comments are now avoidable in those platforms, the fundamental problem has not been solved yet.

To cope with the social issue, we propose the first Korean corpus annotated for toxic speech detection. Specifically, our dataset consists of 9.4K comments from Korean online entertainment news articles. Each comment is annotated on two aspects, the existence of social bias and hate speech, given that hate speech is closely related to bias (Boeckmann and Turpin-Petrosino, 2002; Waseem and Hovy, 2016; Davidson et al., 2017). Considering the context of Korean entertainment news where public figures encounter stereotypes mostly intertwined with gender, we weigh more on the prevalent bias. For hate speech, our label categorization refers that of Davidson et al. (2017), namely *hate*, *offensive*, and *none*.

The main contributions of this work are as follows:

- We release the first Korean corpus manually annotated on two major toxic attributes, namely bias and hate[1].
- We hold Kaggle competitions[2][3][4] and provide benchmarks to boost further research development.
- We observe that in our study, hate speech detection benefits the additional bias context.

[*] Both authors contributed equally to this manuscript.
[†] This work was done after the graduation.

[1] https://github.com/kocohub/korean-hate-speech
[2] www.kaggle.com/c/korean-gender-bias-detection
[3] www.kaggle.com/c/korean-bias-detection
[4] www.kaggle.com/c/korean-hate-speech-detection

Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pages 25–31
Vancouver, Canada, July 30 - August 4, 2017.
https://doi.org/10.18653/v1/P17

2 Related Work

The construction of hate speech corpus has been explored for a limited number of languages, such as English (Waseem and Hovy, 2016; Davidson et al., 2017; Zampieri et al., 2019; Basile et al., 2019), Spanish (Basile et al., 2019), Polish (Ptaszynski et al., 2019), Portuguese (Fortuna et al., 2019), and Italian (Sanguinetti et al., 2018).

For Korean, works on abusive language have mainly focused on the qualitative discussion of the terminology (Hong, 2016), whereas reliable and manual annotation of the corpus has not yet been undertaken. Though profanity termbases are currently available[5][6], term matching approach frequently makes false predictions (e.g., neologism, polysemy, use-mention distinction), and more importantly, not all hate speech are detectable using such terms (Zhang et al., 2018).

In addition, hate speech is situated within the context of social bias (Boeckmann and Turpin-Petrosino, 2002). Waseem and Hovy (2016) and Davidson et al. (2017) attended to bias in terms of hate speech, however, their interest was mainly in texts that explicitly exhibit sexist or racist terms. In this paper, we consider both explicit and implicit stereotypes, and scrutinize how these are related to hate speech.

3 Collection

We constructed the Korean hate speech corpus using the comments from a popular domestic entertainment news aggregation platform. Users had been able to leave comments on each article before the recent overhaul (Yim, 2020), and we had scrapped the comments from the most-viewed articles.

In total, we retrieved 10,403,368 comments from 23,700 articles published from January 1, 2018 to February 29, 2020. We draw 1,580 articles using stratified sampling and extract the top 20 comments ranked in the order of Wilson score (Wilson, 1927) on the downvote for each article. Then, we remove duplicate comments, single token comments (to eliminate ambiguous ones), and comments composed with more than 100 characters (that could convey various opinions). Finally, 10K comments are randomly selected among the rest for annotation.

[5]https://github.com/doublems/korean-bad-words

[6]https://github.com/LDNOOBW/List-of-Dirty-Naughty-Obscene-and-Otherwise-Bad-Words

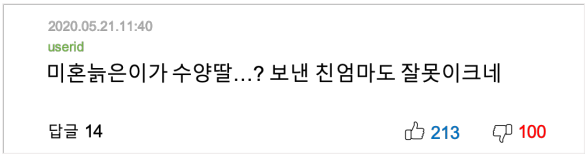


Figure 1: A sample comment from the online news platform. It is composed of six parts: written date and time, masked user id, content, the number of replies, and the number of up/down votes (from top left to bottom right).

We prepared other 2M comments by gathering the top 100 sorted with the same score for all articles and removed with any overlaps regarding the above 10K comments. This additional corpus is distributed without labels, expected to be useful for pre-training language models on Korean online text.

4 Annotation

The annotation was performed by 32 annotators consisting of 29 workers from a crowdsourcing platform *DeepNatural AI*[7] and three natural language processing (NLP) researchers. Every comment was provided to three random annotators to assign the majority decision. Annotators are asked to answer two three-choice questions for each comment:

1. What kind of bias does the comment contain?
   - *Gender bias, Other biases*, or *None*
2. Which is the adequate category for the comment in terms of hate speech?
   - *Hate, Offensive*, or *None*

They are allowed to skip comments which are too ambiguous to decide. Detailed instructions are described in Appendix A. Note that this is the first guideline of social bias and hate speech on Korean online comments.

4.1 Social Bias

Since hate speech is situated within the context of social bias (Boeckmann and Turpin-Petrosino, 2002), we first identify the bias implicated in the comment. Social bias is defined as a preconceived evaluation or prejudice towards a person/group with certain social characteristics: gender, political affiliation, religion, beauty, age, disability, race, or others. Although our main interest is on gender bias, other issues are not to be underestimated.

[7]https://app.deepnatural.ai/

Thus, we separate bias labels into three: whether the given text contains gender-related bias, other biases, or none of them. Additionally, we introduce a binary version of the corpus, which counts only the gender bias, that is prevalent among the entertainment news comments.

The inter-annotator agreement (IAA) of the label is calculated based on Krippendorff's alpha (Krippendorff, 2011) that takes into account an arbitrary number of annotators labeling any number of instances. IAA for the ternary classes is 0.492, which means that the agreement is moderate. For the binary case, we obtained 0.767, which implies that the identification of gender and sexuality-related bias reaches quite a substantial agreement.

4.2 Hate Speech

Hate speech is difficult to be identified, especially for the comments which are context-sensitive. Since annotators are not given additional information, labeling would be diversified due to the difference in pragmatic intuition and background knowledge thereof. To collect reliable hate speech annotation, we attempt to establish a precise and clear guideline.

We consider three categories for hate speech: *hate*, *offensive but not hate*, and *none*. As socially agreed definition lacks for Korean[8], we refer to the hate speech policies of Youtube; Facebook; Twitter. Drawing upon those, we define hate speech in our study as follows:

- If a comment explicitly expresses hatred against individual/group based on any of the following attributes: sex, gender, sexual orientation, gender identity, age, appearance, social status, religious affiliation, military service, disease or disability, ethnicity, and national origin
- If a comment severely insults or attacks individual/group; this includes sexual harassment, humiliation, and derogation

However, note that not all the rude or aggressive comments necessarily belong to the above definition, as argued in Davidson et al. (2017). We often see comments that are offensive to certain individuals/groups in a qualitatively different manner. We identify these as offensive and set the boundary as follows:

[8]Though a government report is available for the Korean language (Hong, 2016), we could not reach a fine extension to the quantitative study on online spaces.

(%)	Hate	Offensive	None	Sum (Bias)
Gender	10.15	4.58	0.98	15.71
Others	7.48	8.94	1.74	18.16
None	7.48	19.13	39.08	65.70
Sum (Hate)	25.11	32.66	41.80	100.00

Table 1: Distribution of the annotated corpus.

- If a comment conveys sarcasm via rhetorical expression or irony
- If a comment states an opinion in an unethical, rude, coarse, or uncivilized manner
- If a comment implicitly attacks individual/group while leaving rooms to be considered as freedom of speech

The instances that do not meet the boundaries above were categorized as *none*. The IAA on the hate categories is $\alpha = 0.496$, which implies a moderate agreement.

5 Corpus

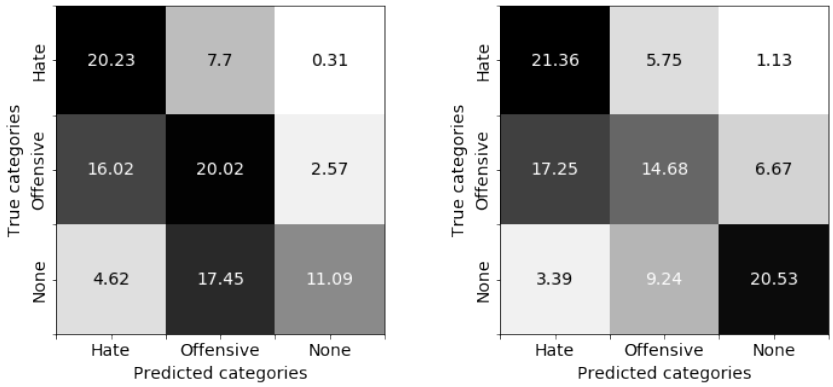
Release From the 10k manually annotated corpus, we discard 659 instances that are either skipped or failed to reach an agreement. We split the final dataset into the train (7,896), validation (471), and test set (974) and released it on the Kaggle platform to leverage the leaderboard system. For a fair competition, labels on the test set are not disclosed. Titles of source articles for each comment are also provided, to help participants exploit context information.

Class distribution Table 1 depicts how the classes are composed of. The bias category distribution in our corpus is skewed towards *none*, while that of *hate* category is quite balanced. We also confirm that the existence of hate speech is correlated with the existence of social bias. In other words, when a comment incorporates a social bias, it is likely to contain hate or offensive speech.

6 Benchmark Experiment

6.1 Models

We implemented three baseline classifiers: character-level convolutional neural network (CharCNN) (Zhang et al., 2015), bidirectional long short-term memory (BiLSTM) (Schuster and Paliwal, 1997), and bidirectional encoder representations from Transformer (BERT) (Devlin et al., 2018) based model. For BERT, we adopt



(a) BERT predictions

(b) BERT predictions with bias label

Figure 2: Confusion matrix on the model inference of hate categories.

F1	Bias (binary)	Bias (ternary)	Hate
Term Matching	-	-	0.195
CharCNN	0.547	0.535	0.415
BiLSTM	0.302	0.291	0.340
BERT	**0.681**	**0.633**	**0.525**
BERT (+ bias)	-	-	**0.569**

Table 2: F1 score of benchmarks on the test set. Note that the term matching model checks the presence of hate or offensiveness. Therefore, in this case, we combine *hate* and *offensive* into a single category, turning the original ternary task into binary.

F1	Gender	Others	None	Bias (ternary)
CharCNN	0.519	0.259	0.826	0.535
BiLSTM	0.055	0.000	0.819	0.291
BERT	0.693	0.326	0.880	**0.633**

Table 3: Detailed results on macro-F1 of Bias (ternary)

KoBERT[9], a pre-trained module for the Korean language, and apply its tokenizer to BiLSTM as well. The detailed configurations are provided in Appendix B, and we additionally report the term matching approach using the aforementioned profanity terms to compare with the benchmarks.

6.2 Results

Table 2 depicts F1 score of the three baselines and the term matching model. The results demonstrate that the models trained on our corpus have an advantage over the term matching method. Compared with the benchmarks, BERT achieves the best performance for all the three tasks: binary and ternary bias identification tasks, and hate speech detection. Each model not only shows different performances but also presents different characteristics.

Bias detection When it comes to the gender-bias detection, the task benefits more on CharCNN than BiLSTM since the bias label is highly correlated with frequent gender terms (e.g., *he, she, man, woman, ...*) in the dataset. It is known that CharCNN well captures the lexical components that are present in the document.

However, owing to that nature, CharCNN sometimes yields results that are overly influenced by the specific terms which cause false predictions. For example, the model fails to detect bias in *"What a long life for a GAY"* but guesses *"I think she is the prettiest among all the celebs"* to contain bias. CharCNN overlooks *GAY* while giving a wrong clue due to the existence of female pronouns, namely *she* in the latter.

Similar to the binary prediction task, CharCNN outperforms BiLSTM on ternary classification. Table 3 demonstrates that BiLSTM hardly identifies *gender* and *other* biases.

BERT detects both biases better than the other models. From the highest score obtained by BERT, we found that rich linguistic knowledge and semantic information is helpful for bias recognition.

We also observed that all the three models barely perform well on *others* (Table 3). To make up a system that covers the broad definition of *other* bias, it would be better to predict the label as the non-*gender* bias. For instance, it can be performed as a two-step prediction: the first step to distinguish whether the comment is biased or not and the second step to determine whether the biased comment is gender-related or not.

[9]https://github.com/SKTBrain/KoBERT

Hate speech detection For hate speech detection, all models faced performance degradation compared to the bias classification task, since the task is more challenging. Nonetheless, BERT is still the most successful, and we conjecture that hate speech detection also utilizes high-level semantic features. The significant performance gap between term matching and BERT explains how much our approach compensates for the false predictions mentioned in Section 2.

Provided *bias* label prepend to each comment as a special token, BERT exhibits better performance. As illustrated in Figure 2, additional bias context helps the model to distinguish *offensive* and *none* clearly. This implies our observation on the correlation between bias and hate is empirically supported.

7 Conclusions

In this data paper, we provide an annotated corpus that can be practically used for analysis and modeling on Korean toxic language, including hate speech and social bias. In specific, we construct a corpus of a total of 9.4K comments from online entertainment news service.

Our dataset has been made publicly accessible with baseline models. We launch Kaggle competitions using the corpus, which may facilitate the studies on toxic speech and ameliorate the cyberbullying issues. We hope our initial efforts can be supportive not only to NLP for social good, but also as a useful resource for discerning implicit bias and hate in online languages.

Acknowledgments

We greatly thank Hyunjoong Kim for providing financial support and Sangwoong Yoon for giving helpful comments.

References

James Banks. 2010. Regulating hate speech online. *International Review of Law, Computers & Technology*, 24(3):233–239.

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. Semeval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 54–63.

Robert J Boeckmann and Carolyn Turpin-Petrosino. 2002. Understanding the harm of hate crime. *Journal of social issues*, 58(2):207–225.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Eleventh international aaai conference on web and social media*.

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*.

Mai ElSherief, Vivek Kulkarni, Dana Nguyen, William Yang Wang, and Elizabeth Belding. 2018. Hate lingo: A target-based linguistic analysis of hate speech in social media. In *Twelfth International AAAI Conference on Web and Social Media*.

Facebook. Facebook's policy on hate speech. https://www.facebook.com/communitystandards/hate_speech. Accessed: 2020-04-19.

Jacey Fortin. 2019. Sulli, south korean k-pop star and actress, is found dead. *New York Times*.

Paula Fortuna, João Rocha da Silva, Leo Wanner, Sérgio Nunes, et al. 2019. A hierarchically-labeled portuguese hate speech dataset. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 94–104.

Seong Soo Hong. 2016. *Hate speech: Survey and Regulations*. National Human Rights Commission of the Republic of Korea.

Klaus Krippendorff. 2011. Computing krippendorff's alpha-reliability.

Taku Kudo and John Richardson. 2018. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

Justin McCurry. 2019a. K-pop singer goo hara found dead aged 28. *The Guardian*.

Justin McCurry. 2019b. K-pop under scrutiny over 'toxic fandom' after death of sulli. *The Guardian*.

Michal Ptaszynski, Agata Pieciukiewicz, and Paweł Dybała. 2019. Results of the poleval 2019 shared task 6: First dataset and open shared task for automatic cyberbullying detection in polish twitter. *Proceedings ofthePolEval2019Workshop*, page 89.

Manuela Sanguinetti, Fabio Poletto, Cristina Bosco, Viviana Patti, and Marco Stranisci. 2018. An italian twitter corpus of hate speech against immigrants. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Twitter. Twitter's policy on hate speech. https://help.twitter.com/en/rules-and-policies/hateful-conduct-policy. Accessed: 2020-04-19.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL student research workshop*, pages 88–93.

Edwin B Wilson. 1927. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212.

Junsuk Yeo. 2019. Kakao suspends online comments for entertainment articles after sulli's death. *The Korea Herald*.

Hyunsu Yim. 2020. Why naver is finally shutting down comments on celebrity news. *The Korea Herald*.

Youtube. Youtube's policy on hate speech. https://support.google.com/youtube/answer/2801939?hl=en. Accessed: 2020-04-19.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.

Ziqi Zhang, David Robinson, and Jonathan Tepper. 2018. Detecting hate speech on twitter using a convolution-gru based deep neural network. In *European semantic web conference*, pages 745–760. Springer.

A Annotation Guideline

A.1 Existence of social bias

The first property is to note which social bias is implicated in the comment. Here, social bias means hasty guess or prejudice that 'a person/group with a certain social identity will display a certain characteristic or act in a biased way'. The three labels of the question are as follows.

1. Is there a gender-related bias, either explicit or implicit, in the text?
   - If the text includes bias for gender role, sexual orientation, sexual identity, and any thoughts on gender-related acts (e.g., "*Wife must be obedient to her husband's words*", or "*Homosexual person will be prone to disease.*")
2. Are there any other kinds of bias in the text?
   - Other kinds of factors that are considered not gender-related but social bias, including race, background, nationality, ethnic group, political stance, skin color, religion, handicaps, age, appearance, richness, occupations, the absence of military service experience[10], etc.
3. A comment that does not incorporate the bias

A.2 Amount of hate, insulting, or offense

The second property is how aggressive the comment is. Since the level of "aggressiveness" depends on the linguistic intuition of annotators, we set the following categorization to draw a borderline as precise as possible.

1. Is strong hate or insulting towards the article's target or related figures, writers of the article or comments, etc. displayed in a comment?
   - In the case of insulting, it encompasses an expression that can severely harm the social status of the recipient.
   - In the case of hate, it is defined as an expression that displays aggressive stances towards individuals/groups with certain characteristics (gender role, sexual orientation, sexual identity, any thoughts on gender-related acts, race, background, nationality, ethnic group, political stance, skin color, religion, handicaps, age, appearance, richness, occupations, the absence of military service experience, etc.).
   - Additionally, it can include sexual harassment, notification of offensive rumors or facts, and coined terms for bad purposes or in bad use, etc.
   - Just an existence of bad words in the document does not always fall into this category.

[10] Frequently observable in Korea, where the military service is mandatory for males.

2. Although a comment is not as much hateful or insulting as the above, does it make the target or the reader feel offended?
    - It may contain rude or aggressive contents, such as bad words, though not to the extent of hate or insult.
    - It can emit sarcasm through rhetorical questions or irony.
    - It may encompass an unethical expression (e.g., jokes or irrelevant questions regarding the figures who passed away).
    - A comment conveying unidentified rumors can belong to this category.
3. A comment that does not incorporate any hatred or insulting

B Model Configuration

Note that each model's configuration is the same for all tasks except for the last layer.

B.1 CharCNN

For character-level CNN, no specific tokenization was utilized. The sequence of Hangul characters was fed into the model at a maximum length of 150. The total number of characters was 1,685, including '[UNK]' and '[PAD]' token, and the embedding size was set to 300. 10 kernels were used, each with the size of [3,4,5]. At the final pooling layer, we used a fully connected network (FCN) of size 1,140, with a 0.5 dropout rate (Srivastava et al., 2014). The training was done for 6 epochs.

B.2 BiLSTM

For bidirectional LSTM, we had a vocab size of 4,322, with a maximum length of 256. We used BERT SentencePiece tokenizer (Kudo and Richardson, 2018). The width of the hidden layers was 512 (=256×2), with four stacked layers. The dropout rate was set to 0.3. An FCN of size 1,024 was appended to the BiLSTM output to yield the final softmax layer. We trained the model for 15 epochs.

B.3 BERT

For BERT, a built-in SentencePiece tokenizer of KoBERT was adopted, which was also used for BiLSTM. We set a maximum length at 256 and ran the model for 10 epochs.

Stance Prediction for Contemporary Issues: Data and Experiments

Marjan Hosseinia
University of Houston
mhosseinia@uh.edu

Eduard Dragut
Temple University
edragut@temple.edu

Arjun Mukherjee
University of Houston
arjun@cs.uh.edu

Abstract

We investigate whether pre-trained bidirectional transformers with sentiment and emotion information improve stance detection in long discussions of contemporary issues. As a part of this work, we create a novel stance detection dataset covering 419 different controversial issues and their related pros and cons collected by procon.org in nonpartisan format. Experimental results show that a shallow recurrent neural network with sentiment or emotion information can reach competitive results compared to fine-tuned BERT with 20× fewer parameters. We also use a simple approach that explains which input phrases contribute to stance detection.

1 Introduction

Stance detection identifies whether an opinion is in favor of an idea or opposes it. It has a tight connection with sentiment analysis; however, stance detection usually investigates the two-sided relationship between an opinion and a question. For example, 'should abortion be legal?' or 'is human activity primarily responsible for global climate change?'

Contemporary debatable issues, even though non-political, usually carry some political weight and controversy. For example, legislators may allow soda vending machines in our school or consider obesity as a health issue that directly impacts soda manufacturers and insurance companies respectively. On a larger scale, an issue such as climate change is being discussed in US presidential debates constantly. Meanwhile, information about these issues is mostly one-sided and provided by left or right partisan resources. Such information forms public beliefs, has persuasive power, and promotes confirmation bias (Stanojevic et al., 2019), the humans' tendency to search for the information which confirms their existing beliefs [1]. Confirmation bias permits internet debates and promote discrimination, misinformation, and hate speech, all of which are emerging problems in user posts of social media platforms.

Although there are many attempts to automatic identification and removal of such contents from online platforms, the need for accessing bi-partisan information that cultivates critical thinking and avoids confirmation bias remains. In this regard, a few web sources, such as procon.org, present information in a non-partisan format and being used as a resource for improving critical thinking in educational training by teachers [2].

Here, we aim to improve such resources by automatic stance detection of pro or con-perspectives regarding a debatable issue. We extend our previous work (Hosseinia et al., 2019) by creating a new dataset from procon.org with 419 distinct issues and their two-sided perspectives annotated by its experts [3]. Then, we leverage external knowledge to identify the stance of a perspective towards an issue that is mainly represented in the form of a question.

The latest progress in pre-trained language models (Howard and Ruder, 2018) and transformers (Devlin et al., 2019; Yang et al., 2019) allows one to create general models with less amount of effort for task-specific text classification. In this work, we show that bidirectional transformers can produce competitive results even without fine-tuning by leveraging auxiliary sentiment and emotion information (Dragut et al., 2010). Experimental results show the effectiveness of our model and its remarkable performance. The model has a signif-

[1] www.procon.org/education.php
[2] https://www.procon.org/view.background-resource.php?resourceID=004241
[3] https://github.com/marjanhs/procon20/

Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pages 32–40
Vancouver, Canada, July 30 - August 4, 2017.
https://doi.org/10.18653/v1/P17

icantly smaller size compared to the BERT-base model.

The main contributions of this work are as following:

- Proposing a simple but efficient recurrent neural network that leverages sentence-wise sentiment or token-level emotion of input sequence with BERT representation for detecting the stance of a long perspective against its related question.
- Creating a novel dataset for stance detection with more than 6K instances.
- Explaining the word/phrase contribution of input sequence using max-pooling engagement score for stance detection.

2 Related Works

We group stance detection methods based on underlying data and approaches as follows:

- *Tweets* are collected from SemEval 2016, Task 6, (Mohammad et al., 2016) and organized in two categories. The first category, which represents a supervised setting, includes tweets that cover opinions about five topics, "Atheism", "Climate Change", "Feminist Movement", "Hillary Clinton", and "Legalization of Abortion". The second category, which represents weakly supervised settings, includes tweets that cover one topic, but the training data is unlabeled.
- *Claims* are obtained from Wikipedia in (Bar-Haim et al., 2017). Each claim is defined as a brief statement that is often part of a Wikipedia sentence. The claim dataset contains 55 different topics.
- *Debates* are gathered from various online debate resources, including `idebate`, `debatewise` and `procon` in the form of perspective, claim, and evidence for substantiated perspective discovery. 49 out of its 947 claims are from `procon` (Chen et al., 2019). Claims and perspectives are short sentences and have been used for stance detection in (Popat et al., 2019).

Current approaches on stance detection use different types of linguistic features, including word/character n-grams, dependency parse trees, and lexicons (Sun et al., 2018; Sridhar et al., 2015; Hasan and Ng, 2013; Walker et al.). There are also end-to-end neural network approaches that learn topics and opinions independently while joining them with memory networks (Mohtarami et al., 2018), bidirectional conditional LSTM (Augenstein et al., 2016), or neural attention (Du et al., 2017). There are also some neural network approaches that leverage lexical features (Riedel et al., 2017; Hanselowski et al., 2018). A consistency constraint is proposed to jointly model the topic and opinion using BERT architecture (Popat et al., 2019). It trains the whole massive network for label prediction. None of these approaches incorporate bidirectional transformers with sentiment and emotion in a shallow neural network as we propose in this paper. Additionally, our focus is to find the stance of 100-200 words long discussions, which are commonly present in nonpartisan format.

3 Dataset

We collect data from `procon.org`, a non-profit organization that presents opinions on controversial issues in a nonpartisan format. Issues (questions) and their related responses are professionally researched from different online platforms by its experts. The dataset covers 419 different detailed issues ranging from politics to sport and healthcare. The dataset instances are pairs of issues, in the form of *questions*, and their corresponding *perspectives* from proponents and opponents. Each *perspective* is either a pro or a con with 100-200 words that supports its claim with compelling arguments. Table 1 provides some examples of the questions from the dataset. The dataset statistics are also presented in Table 2. We may use the words *opinion* and *perspective* interchangeably as both refer to the same concept in this work.

4 Model

Utilizing pre-trained models has been widely popular in machine translation and various text classification tasks. Prior efforts were hindered by the lack of labeled data (Zhang et al., 2019). With the growth of successful pre-trained models, a model fine-tuned on a small portion of data can compete with models trained on 10× more training data without pre-training (Howard and Ruder, 2018). Recently, transformer models trained on both directions of language simultaneously, such as BERT (Devlin et al., 2019) and XLNet (Yang et al., 2019),

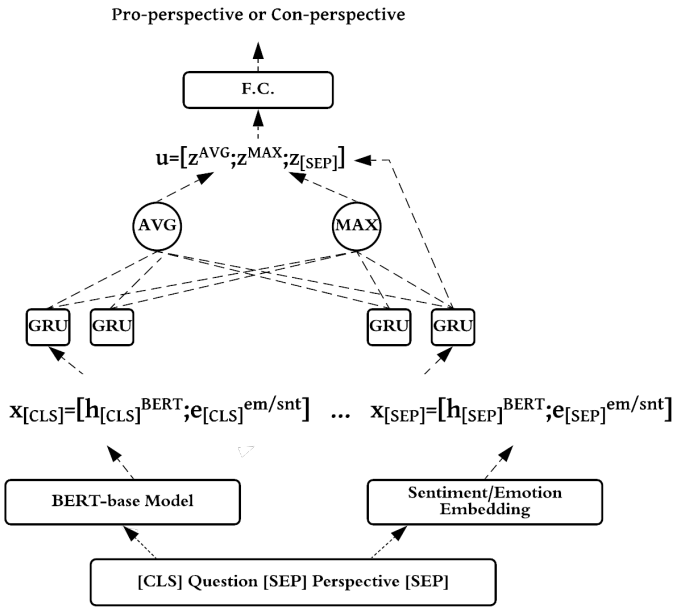


Figure 1: Stance detection architecture; snt: sentiment; em: emotion

HEALTH and MEDICINE 1- Should euthanasia or physician-assisted suicide be legal? 2- Is vaping with e-cigarettes safe?
EDUCATION 1-Should parents or other adults be able to ban books from schools and libraries? 2- Should public college be tuition-free?
POLITICS 1- Should recreational marijuana be legal? 2- Should more gun control laws be enacted?
SCIENCE and TECHNOLOGY 1- Is cell phone radiation safe? 2- Should net neutrality be restored?
ENTERTAINMENT and SPORTS 1- Are social networking sites good for our society? 2- Do violent video games contribute to youth violence?

Table 1: Procon dataset questions

overcome previous unidirectional language models (e.g. ULMFiT (Howard and Ruder, 2018)) or models trained on two independent directions (ELMo) (Peters et al., 2018) significantly. So, we build our baselines based on BERT architecture in two different ways: single and pair of inputs. A question and its related opinion are concatenated for single inputs. However, for input pairs, the question and the opinion are being separated with the BERT separator tag [SEP]. This approach has been used for question-answering applications (Devlin et al., 2019).

Opinion is connected with sentiment and emotion (Schneider and Dragut, 2015). Moreover, prior efforts show the successful employment of linguistic features, extracted with external tools, in neural networks for emotional cognition (Yang et al., 2017). So, we leverage sentiment and emotion information separately with BERT representations obtained from the last BERT-base layer to form the input of a shallow recurrent neural network. In the following, we provide the details.

- Employing sentiment: We analyze how the sentiment of sentences in proponents' and opponents' opinions can affect stance detection. Accordingly, we use a rule-based sentiment tool, VADER (Hutto and Gilbert, 2014), for obtaining the sentiment of a sentence. VADER translates its compound sentiment score, ranging from -1 to $+1$, into negative sentiment labels for scores $\leqslant -0.05$, positive labels for scores $\geqslant +0.05$, and neutral for the scores between -0.05 and $+0.05$.

  Here, we compute sentence-wise sentiment using VADER to let the model learn the flow of sentiment across the opinion. So, each token borrows the sentiment of its correspond-

Set	#of Topics	#of Words	#of Pro-perspectives	#of Con-perspectives	Total
train	417	127	2,140	2,125	4,265
dev	265	125	326	284	610
test	336	123	613	606	1,219

Table 2: Procon dataset statistics

ing sentence. Then, an embedding layer converts the discrete labels into d-dimensional vectors ($d = 768$) using a randomly initialized matrix $W^s_{3\times d}$; These representations are concatenated with BERT token embeddings to form the bidirectional Gated Recurrent Units ($\overleftrightarrow{\text{GRU}}$) input ($x_t$ for token t):

$$
\begin{aligned}
x_t &= [h_t^{\text{BERT}}; e_t^{\text{snt}}], \\
z_t &= \overleftrightarrow{\text{GRU}}(x_t), \\
u &= [\text{avg-pool}(Z); \text{max-pool}(Z); z_T], \\
y &= \text{softmax}(Wu + b)
\end{aligned}
$$

For an input sequence with T tokens, h_t^{BERT} is the hidden state of the last BERT-base layer corresponding to the input token at time t, e_t^{snt} is sentiment embedding of the token, [;] denotes concatenation operator, $Z = [z_i]_{i=1}^{T}$, and W, b are parameters of a fully connected layer.

Recall that our task is to identify the stance of long opinions; So, important information towards the final stance might be anywhere in the opinion. Because of that, we collect such information from the recurrent hidden states of all input tokens using max and average-pooling. Max-pooling returns a vector with maximum weights across all hidden states of input tokens for each dimension. In this way, the input tokens with higher weights will be engaged for stance prediction. Aside from that, the last hidden state of the recurrent network (z_T) is concatenated with the pooled information (u). Finally, a dense layer transforms vector u into the class dimension. Figure 1 shows the model architecture.

We refer to this model as VADER-Sent-GRU and report the experimental results in Section 6.

- Employing emotion: We take a similar approach to engage emotion information for stance detection using the NRC emotion lexicon (Mohammad and Turney, 2013). The Lexicon is collected by crowdsourcing and consists of English words with their eight basic emotions including anger, fear, anticipation, trust, surprise, sadness, joy, and disgust. So, the GRU input is a concatenation of BERT representation with emotion embedding (gained from a $9 \times d$ matrix with random initialization; one dimension is added for neutral emotion). Here, we use unidirectional $\overrightarrow{\text{GRU}}$ as it shows more stable results in our pilot experiments.

5 Experiments

In this section, we describe the corresponding baselines followed by the training setup.

5.1 Baselines

We use the following baselines utilized in opinion mining including sentiment analysis and stance detection:

- BERT (Devlin et al., 2019) followed by a non-linear transformation on a dense layer is used for downstream stance detection. Here, the whole network is fine-tuned and all 12 BERT-base layers' weights will be updated in back-propagation. The information is pooled from the final hidden state of the classification token ($h^{\text{BERT}}_{[\text{cls}]}$) after passing a fully connected layer with non-linear activation ($\tanh$). Then, a classifier layer shrinks the activations to a binary dimension.

$$
\begin{aligned}
x &= \tanh(W^p h^{\text{BERT}}_{[\text{cls}]} + b^p), \\
y &= W^c x + b^c
\end{aligned}
$$

  where $W^c, W^p, b^p,$ and b^c are the layers' parameters.

- $\text{BERT}_{\text{CONS}}$ is a BERT base model that considers two different inputs using a perspective and its respective claim (Popat et al., 2019). The first input is similar to BERT sentence model's, *[CLS] claim [SEP] perspective*

[SEP], and the second one is the sequence of *[CLS] claim [SEP]*. Each input will be given to the BERT model separately. The goal is to incorporate the consistency between the representation of the perspective and claim using cosine distance of the two inputs. Accordingly, the following loss ($loss_c$) is added to the regular cross-entropy loss of the BERT model:

$$loss_c = \begin{cases} 1 - cos(X^{[C]}, X^{[C;P]}), \text{y=pro} \\ max(0, cos(X^{[C]}, X^{[C;P]}), \text{y=con} \end{cases}$$

where $X^{[C]}$ and $X^{[C;P]}$ are the final hidden state representations corresponding to the [CLS] token of the BERT model for the specified input. In our experiments, we replace the underlying question of a perspective with the claim in the two input sequences.

- XML-CNN model consists of three convolution layers with kernel size$= (2, 4, 8)$. With a dynamic max-pooling layer, crucial information is extracted across the document. XML-CNN was able to beat most of its deep neural network baselines in six benchmark datasets (Liu et al., 2017). We use, BERT, Word2vec, and FastText (Mikolov et al., 2018) embeddings for input tokens.

- AWD-LSTM is a weight-dropped LSTM that deploys DropConnect on hidden-to-hidden weights as a form of recurrent regularization (Merity et al., 2017). Word2vec Embedding is used for its input.

We define the corresponding hidden states of the last BERT layer as BERT embedding/representation of input sequence for both single and pair of inputs mode.

5.2 Training

We develop our code based on the Hedwig[4] implementation and train the models on 30 epochs with batch size=8. We apply early stopping technique to avoid overfitting during training. Training is stopped after 5 consequent epochs of no improvement of the highest F1 score. We inspect the test set on the model with the best F1 score of development set and keep the settings for BERT the same as the *BERT-base-uncased* model. Adam optimizer with the learning rate of $2e-5$ (for BERT) and $2e-4$ (for other models) is used. We see a dramatic drop in BERT performance with some other learning rates. Scikit-learn (Pedregosa et al., 2011) library is employed for evaluation measures.

[4]https://github.com/castorini/hedwig

Model	P.	R.	F1
Pair of Input			
BERT	**76.49**	75.37	75.92*†
BERT$_{\text{CONS}}$	70.34	81.24	75.40
XML-CNN(BERT)	68.48	82.22	74.72
VADER-Sent-$\overleftrightarrow{\text{GRU}}$	69.14	**86.62**	**76.90**†
NRC-Emotion-$\overrightarrow{\text{GRU}}$	73.79	79.45	76.51*
Unary Input			
BERT	73.89	76.18	75.02
AWD-LSTM(Word2Vec)	65.93	73.25	69.40
XML-CNN(Word2Vec)	58.30	83.03	68.51
XML-CNN(FastText)	66.85	77.32	71.71
XML-CNN(BERT)	70.30	79.93	74.81
VADER-Sent-$\overleftrightarrow{\text{GRU}}$	66.36	82.38	73.51
NRC-Emotion-$\overrightarrow{\text{GRU}}$	68.46	83.20	75.11

Table 3: Evaluation results; P.:Precision, R.:Recall, *: p-value ≤ 0.001; †: p-value ≤ 0.0001

6 Results and Discussion

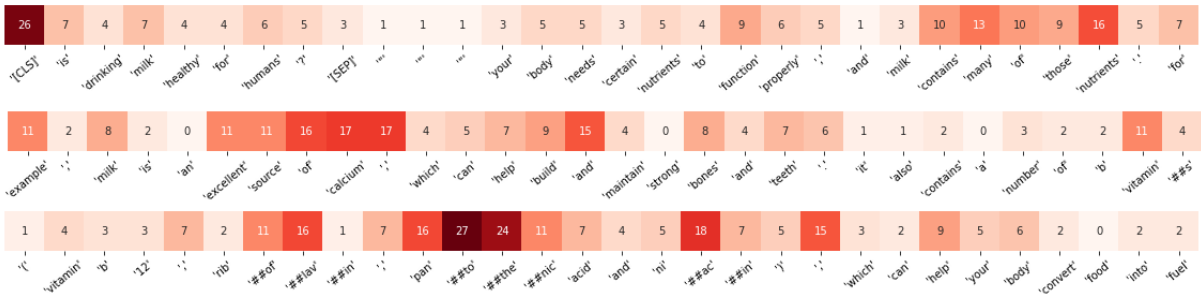
Experimental results are provided in Table 3. It was expected that fine-tuning BERT with a pair of input achieves a competitive performance among other baselines; but it shows that even with a shallow concatenation of the question and perspective (unary input), BERT can achieve consistent results. Moreover, models that take BERT representation in feature selection mode (without fine-tuning), e.g. XML-CNN(BERT), show better stance detection performance than other token embeddings.

We apply McNemar's test to measure whether the disagreement between the predictions of the two models is statistically significant.

Among the models with pairs of input, VADER-Sent-GRU gains the highest recall and F1 score. It indicates that the external knowledge gained from a massive corpus, fine-tuned on $20\times$ fewer parameters and enriched with sentiment information can compete with the original architecture (75.92 vs 76.90, $p < 0.0001$). As the model is significantly smaller, it trains faster and needs fewer resources for training. NRC-Emotion-GRU, highlighted in gray, achieves the second-highest F1 score among the models; It reveals that adding emotion information improves stance detection (75.92 vs 76.51,

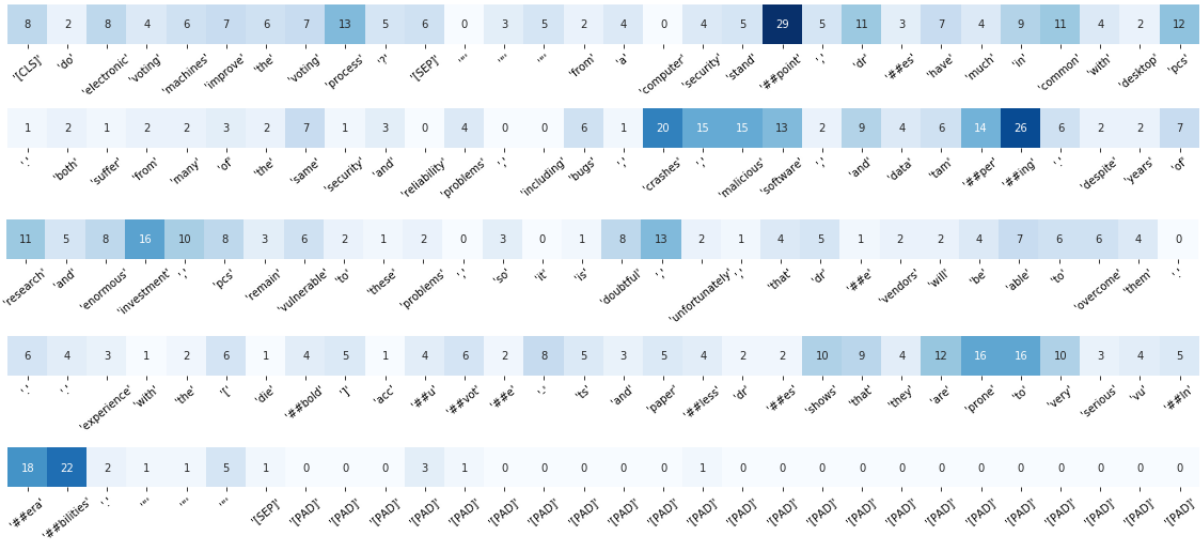
Model	P.	R.	F1
VADER-Sent-$\overleftrightarrow{\text{GRU}}$	69.14	86.62	76.90
$\overleftrightarrow{\text{GRU}}$	72.18	78.30	75.12 (**1.78** ↓)
NRC-Emotion-$\overrightarrow{\text{GRU}}$	73.79	79.45	76.51
$\overrightarrow{\text{GRU}}$	69.14	83.69	75.72 (**0.79** ↓)

Table 4: Effect of sentiment and emotion in our models with pair of input



Question :	Is drinking milk healthy for humans?
Top words:	nutrients, calcium, niacin, riboflavin, and pantothenic

Table 5: A pro-perspective



Question :	Do electronic voting machines improve the voting process?
Top words:	vulnerabilities, investment, standpoint, crashes, malicious software, and tampering

Table 6: A con-perspective

$p < 0.001$). However, employing sentiment information is more helpful than emotion in detecting the stance of opinions with compelling arguments (76.51 vs 76.90, $p < 0.0001$).

Unlike the superiority of $\text{BERT}_{\text{CONS}}$ over BERT reported in (Popat et al., 2019), we do not see a similar performance here. $\text{BERT}_{\text{CONS}}$ uses cosine similarity between the BERT representations of [claim] and [perspective; claim] in the loss function such that their representations become similar when per-

spective supports the claim and dissimilar when it opposes the claim. This method works for claims and perspectives of the Perspectrum dataset where the two input components are short sentences with $5-10$ words long. However, in our dataset, we have a *question* and its perspective that spans multiple sentences. So, forcing the model to make the BERT representations of [question] and [perspective; question] similar or dissimilar, according to the stance, harms the model training. Because the input components have different characteristics utilizing this method results in lower performance than the base model (BERT).

Next, we present some experiments to better understand the model's units.

6.1 Effect of Sentiment and Emotion

As stated in Section 4, our recurrent models (VADER-Sent-GRU and NRC-Emotion-GRU) employ sentiment and emotion information of tokens respectively. To see the effect of learning the flow of sentiment and emotion across an opinion, we lift their embeddings from the input of the models. So, $\overrightarrow{\text{GRU}}$ and $\overleftrightarrow{\text{GRU}}$ are unidirectional and bidirectional Gated Recurrent Units network respectively, followed by pooling and classification layers:

$$
\begin{aligned}
x_t &= h_t^{\text{BERT}}, \\
z_t &= \text{GRU}(x_t), \\
u &= [\text{avg-pool}(Z); \text{max-pool}(Z); z_T], \\
y &= \text{softmax}(Wu + b)
\end{aligned}
$$

Similarly, for an input sequence with T tokens, h_t^{BERT} is the hidden state of the last BERT layer corresponding to the input token at time t, $Z = [z_i]_{i=1}^{T}$, and W, b are parameters of a fully connected layer.

According to the results in Table 4, both precision and F1 score reduce for the model without emotion ($\overrightarrow{\text{GRU}}$); however, we see a reduction in recall and F1 in the model after lifting sentiment ($\overleftrightarrow{\text{GRU}}$) indicating that integrating sentence-wise sentiment and token-level emotion impact stance detection. We also provide the average sentiment score of the perspectives regarding five different questions in Figure 2. The figure shows the difference between the sentiment of the two stance classes in each issue resulting in a better stance classification. In the next part, we analyze the effect of pooling.

6.2 Pooling Explanation

In (Popat et al., 2019), authors find the most important phrases of input by removing phrases from the sequence and finding the ones with maximum effect on misclassification. In our model, we find the crucial information engaged in identifying the stance of a perspective using the max-pooling operation applied to the output sequence of recurrent neural networks (see Section 4). We hypothesize that the more a token is engaged in max-pooling, the more critical the token is for final stance prediction.

Tables 5 and 6 show the heatmap plots of two test instances. The number in each square is the engagement score, the frequency of the presence of a token in max-pooling operation. Darker colors show a higher frequency and indicate how the model identifies the stance across the perspective towards a question. The underlying question in Table 5 asks 'Is drinking milk healthy for humans?' According to its figure, we find sub-tokens of *nutrients*, *calcium*, *niacin*, *riboflavin*, and *pantothenic* with high scores. All of these words are positively aligned with the final (pro) stance; Specifically, the last three words are a type of Vitamin B. In another example in Table 6, the question is 'Do electronic voting machines improve the voting process?' Its corresponding heatmap displays sub-tokens of *vulnerabilities*, *investment*, *standpoint*, *crashes*, *malicious software*, and *tampering* with high scores; all of which are almost consistent with the perspective's (con) stance.

Similarly, we find the most important words/phrases, regarding their engagement score, for a few other examples of the test set that are correctly classified. The sub-tokens of these phrases have the highest frequency in max-pooling operation. We add (pro) or (con) at the end of each phrase list to indicate the stance of their respective perspective.

- Should students have to wear school uniforms? uniforms restrict students' freedom of expression (con)
- Are social networking sites good for our society? lead to stress and offline relationship (con)
- Should recreational marijuana be legal? legalization, odious occasion (con)

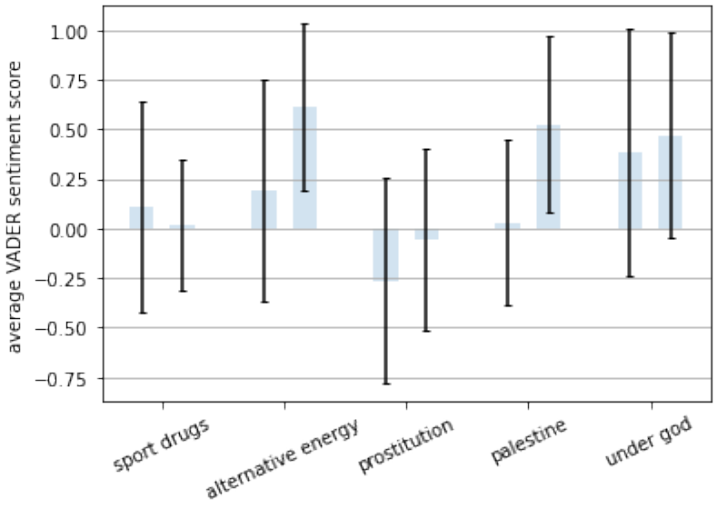


Figure 2: Average VADER sentiment scores across five different issues. In each issue the first bar belongs to proponents and the second bar belongs to the opponents

- What are the pros and cons of milk's effect on cancer? dairy consumption is linked with rising death rates from prostate cancer (con)
- Is human activity responsible for climate change? significant, because, (likely greater than 95 percent probability) (pro)
- Is obesity a disease? no question that obesity is a disease, blood sugar is not functioning properly, dysregulation, diabetes (pro)
- Is the death penalty immoral? anymore, failed policy (pro)

The above list shows that the stance-related phrases have been well identified by the model in the pooling step.

7 Conclusion

We propose a model that leverages BERT representation with sentiment or emotion information for stance detection. We create a new dataset for the perspectives that are as long as a paragraph covering a wide variety of contemporary topics. The experiments on our benchmark dataset highlight the effect of emotion and sentiment in stance prediction. The model can improve BERT base performance with significantly fewer parameters. We also explain the contribution of essential phrases of perspectives in detecting their stance using max-pooling operation.

Acknowledgments

This work is supported in part by the U.S. NSF grants 1838147 and 1838145. We also thank anonymous reviewers for their helpful feedback.

References

Isabelle Augenstein, Tim Rocktäschel, Andreas Vlachos, and Kalina Bontcheva. 2016. Stance detection with bidirectional conditional encoding. In *Proceedings of the 2016 Conference on EMNLP*, pages 876–885, Austin, Texas. ACL.

Roy Bar-Haim, Indrajit Bhattacharya, Francesco Dinuzzo, Amrita Saha, and Noam Slonim. 2017. Stance classification of context-dependent claims. In *Proceedings of the 15th Conference of EACL: Volume 1, Long Papers*, pages 251–261, Valencia, Spain. ACL.

Sihao Chen, Daniel Khashabi, Wenpeng Yin, Chris Callison-Burch, and Dan Roth. 2019. Seeing things from a different angle:discovering diverse perspectives about claims. In *NAACL*, pages 542–557, Minneapolis, Minnesota. ACL.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT)*, pages 4171–4186, Minneapolis, Minnesota. ACL.

Eduard C. Dragut, Clement T. Yu, A. Prasad Sistla, and Weiyi Meng. 2010. Construction of a sentimental word dictionary. In *CIKM*, pages 1761–1764.

Jiachen Du, Ruifeng Xu, Yulan He, and Lin Gui. 2017. Stance classification with target-specific neural attention networks. IJCAI.

Andreas Hanselowski, Avinesh PVS, Benjamin Schiller, Felix Caspelherr, Debanjan Chaudhuri, Christian M. Meyer, and Iryna Gurevych. 2018. A retrospective analysis of the fake news challenge stance-detection task. In *ACL*, pages 1859–1874, Santa Fe, New Mexico, USA. ACL.

Kazi Saidul Hasan and Vincent Ng. 2013. Stance classification of ideological debates: Data, models, features, and constraints. In *Proceedings of the Sixth IJCNLP*, pages 1348–1356, Nagoya, Japan. AFNLP.

Marjan Hosseinia, Eduard Dragut, and Arjun Mukherjee. 2019. Pro/con: Neural detection of stance in argumentative opinions. In *SBP-BRiMS*, pages 21–30.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of ACL*, pages 328–339, Melbourne, Australia. ACL.

Clayton J. Hutto and Eric Gilbert. 2014. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *ICWSM*.

Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. 2017. Deep learning for extreme multi-label text classification. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 115–124. ACM.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2017. Regularizing and optimizing lstm language models. *CoRR*, abs/1708.02182.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of LREC 2018*.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. SemEval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 31–41, San Diego, California. ACL.

Saif M. Mohammad and Peter D. Turney. 2013. Crowdsourcing a word-emotion association lexicon. 29(3):436–465.

Mitra Mohtarami, Ramy Baly, James Glass, Preslav Nakov, Lluís Màrquez, and Alessandro Moschitti. 2018. Automatic stance detection using end-to-end memory networks. In *Proceedings of the 2018 Conference of NAACL: Human Language Technologies*, pages 767–776, New Orleans, Louisiana. ACL.

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

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 NAACL: Human Language Technologies*, pages 2227–2237, New Orleans, Louisiana. ACL.

Kashyap Popat, Subhabrata Mukherjee, Andrew Yates, and Gerhard Weikum. 2019. STANCY: Stance classification based on consistency cues. In *Proceedings of the 2019 Conference on EMNLP-IJCNLP*, pages 6413–6418, Hong Kong, China. Association for Computational Linguistics.

Benjamin Riedel, Isabelle Augenstein, Georgios P. Spithourakis, and Sebastian Riedel. 2017. A simple but tough-to-beat baseline for the Fake News Challenge stance detection task. *CoRR*, abs/1707.03264.

Andrew T. Schneider and Eduard C. Dragut. 2015. Towards debugging sentiment lexicons. In *ACL*, pages 1024–1034.

Dhanya Sridhar, James Foulds, Bert Huang, Lise Getoor, and Marilyn Walker. 2015. Joint models of disagreement and stance in online debate. In *Proceedings of the 53rd ACL and the 7th IJCNLP*, pages 116–125, Beijing, China. ACL.

Marija Stanojevic, Jumanah Alshehri, Eduard C. Dragut, and Zoran Obradovic. 2019. Biased news data influence on classifying social media posts. In *NEwsIR@SIGIR*, volume 2411, pages 3–8.

Qingying Sun, Zhongqing Wang, Qiaoming Zhu, and Guodong Zhou. 2018. Stance detection with hierarchical attention network. In *ACL*, pages 2399–2409, Santa Fe, New Mexico, USA. ACL.

Marilyn Walker, Pranav Anand, Rob Abbott, and Ricky Grant. Stance classification using dialogic properties of persuasion.

Fan Yang, Arjun Mukherjee, and Eduard Dragut. 2017. Satirical news detection and analysis using attention mechanism and linguistic features. In *Proceedings of the 2017 Conference on EMNLP*, pages 1979–1989, Copenhagen, Denmark. ACL.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

Shanshan Zhang, Lihong He, Eduard Dragut, and Slobodan Vucetic. 2019. How to invest my time: Lessons from human-in-the-loop entity extraction. In *SIGKDD*, page 2305–2313.

Challenges in Emotion Style Transfer: An Exploration with a Lexical Substitution Pipeline

David Helbig, Enrica Troiano and **Roman Klinger**
Institut für Maschinelle Sprachverarbeitung
University of Stuttgart, Germany
{david.helbig,enrica.troiano,roman.klinger}@ims.uni-stuttgart.de

Abstract

We propose the task of emotion style transfer, which is particularly challenging, as emotions (here: anger, disgust, fear, joy, sadness, surprise) are on the fence between content and style. To understand the particular difficulties of this task, we design a transparent emotion style transfer pipeline based on three steps: (1) select the words that are promising to be substituted to change the emotion (with a brute-force approach and selection based on the attention mechanism of an emotion classifier), (2) find sets of words as candidates for substituting the words (based on lexical and distributional semantics), and (3) select the most promising combination of substitutions with an objective function which consists of components for content (based on BERT sentence embeddings), emotion (based on an emotion classifier), and fluency (based on a neural language model). This comparably straightforward setup enables us to explore the task and understand in what cases lexical substitution can vary the emotional load of texts, how changes in content and style interact and if they are at odds. We further evaluate our pipeline quantitatively in an automated and an annotation study based on Tweets and find, indeed, that simultaneous adjustments of content and emotion are conflicting objectives: as we show in a qualitative analysis motivated by Scherer's emotion component model, this is particularly the case for implicit emotion expressions based on cognitive appraisal or descriptions of bodily reactions.

1 Introduction

Humans are capable of saying the same thing in many ways. Careful lexical choices can re-shape a concept in different modes of presentation, giving it a humourous tone, for example, or some degree of formality, or a rap vibe. This type of linguistic creativity has recently been mirrored in the task of textual style transfer, where a stylistic variation is induced on an existing piece of text. The core idea is that texts have a content and a style, and that it is possible to keep the one while changing the other.

Past work on style transfer has targeted attributes (or styles) like sentiment (Dai et al., 2019) and tense (Hu et al., 2017), producing a rich literature on deep generative models that disentangle the content and the style of an input text, and subsequently condition generation towards a desired style (Fu et al., 2018; Shen et al., 2017; Prabhumoye et al., 2018). With this paper, we propose a non-binary style transfer setting, namely emotion style transfer, in which the target corresponds to one emotion (following Ekman's fundamental emotions of anger, fear, joy, surprise, sadness, and disgust). Further, this setting is particularly challenging as emotions are on the fence between content and style. To the best of our knowledge, this type of attribute has been explored only to some degree by the unpublished work by Smith et al. (2019), who transfer text towards 20 affect-related styles. Emotions received more attention in conditioned text generation (Ghosh et al., 2017; Huang et al., 2018; Song et al., 2019).

To explore the challenges of emotion style transfer (for which we depict an example in Figure 1), we develop a transparent pipeline based on lexical substitution (in contrast to a black-box neural encoder/decoder approach), in which we first (1) select those words that are promising to be changed to adapt the target style, (2) find candidates that may substitute these words, (3) select the best combination regarding content similarity to original

In (Anger):	This **soul-crushing drudgery plagues** him
Out (Joy):	This **fulfilling job motivates** him

Figure 1: An example of emotion transfer performed with lexical substitution.

Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pages 41–50
Vancouver, Canada, July 30 - August 4, 2017.
https://doi.org/10.18653/v1/P17

input, target style, and fluency. As we will see, this straight-forward approach is promising while it still enables to understand the changes to the text and their function.

Emotions are not only interesting from the point of view that they contribute to content and style. They are also a comparably well-investigated phenomenon with a rich literature in psychology. For instance, Scherer (2005) states that emotions consist of different components, namely a cognitive appraisal, bodily symptoms, a subjective feeling, expression, and action tendencies. Descriptions of all these components can be realized in natural language to communicate a specific private emotional state. We argue (and analyze based on examples later) that a report of a feeling ("*I am happy*") might be challenging in a different way than descriptions of bodily reactions (*"I am sweating"*) or events (*"My dog was overrun by a car"*).

With our white-box approach of style transfer and the evaluation on the novel task of emotion transfer, we address the following research questions: To what extent can lexical substitution modulate the emotional leaning of text? What is its limitation (e.g., by changing the emotion "style", does content change as well)? Our results show that the success of this approach, both in terms of style change and content preservation, depends on the strategies used for selection and substitution, and that emotion transfer is a viable task to address. Further, we see in a qualitative analysis that what an emotion classification model bases its decisions on might not be sufficient to guide a style transfer method. This becomes evident when we compare how transfer is realized across types of emotion expressions, corresponding to specific components of Scherer's model.

Our implementation is available at http://www.ims.uni-stuttgart.de/data/lexicalemotiontransfer.

2 Related Work

2.1 Emotion Analysis

In the field of psychology, the two main emotion traditions are categorical models and the strand that focuses on the continuous nature of humans' affect (Scherer, 2005). Emotions are grouped into categories corresponding to emotion terms, some of which are prototypical experiences shared across cultures. For Ekman (1992), they are anger, joy, surprise, disgust, fear and sadness; on top of these, Plutchik (2001) adds anticipation and trust. Posner et al. (2005), instead locates emotions along interval scales of affect components (valence, arousal, dominance).

These studies have also influenced computational approaches to emotions, whose preliminary requirement is to follow a specific conceptualization coming from psychology, in order to determine the number and type of emotion classes to research in language. Emotion analysis in natural language processing has mainly established itself as a classification task, aimed at assigning a text to the emotion it expresses (Alm et al., 2005). It has been conducted on a variety of corpora that encompass different types of annotations[1], based on one of the established emotion models mentioned above. Such studies also differ with respect to the textual genres they consider, ranging from from tweets (Mohammad et al., 2017; Klinger et al., 2018) to literary texts (Kim et al., 2017).

While emotion classification approaches have been used to guide controlled generation of text (Ghosh et al., 2017; Huang et al., 2018; Song et al., 2019), computationally modelling emotions has not yet been applied to style transfer. After describing a method to address such task, we analyse its performance by leveraging Scherer's component model: emotions are underlied by various dimensions of cognitive appraisal, which can be differently expressed in text and may pose different challenges for style transfer.

2.2 Style Transfer

Most of the recently published approaches to style transfer make use of artificial neural network architectures, in which some latent semantic representation is the backbone of the system. For instance, Prabhumoye et al. (2018) use neural back-translation to encode the content of text while reducing its stylistic properties, and later decoding it with a specific target style. Gong et al. (2019) evaluate paraphrases regarding their fluency, similarity to the input text and expression of a desired target style, and use this as feedback in a reinforcement learning approach. Li et al. (2018) combine rules with neural methods to explicitly encode attribute markers of the target style.

Such transfer methods have been applied to a variety of styles, including sentiment (Shen et al., 2017; Fu et al., 2018; Xu et al., 2018) and a num-

[1] A comprehensive list of available emotion datasets and annotation schemes can be found in Bostan and Klinger (2018).

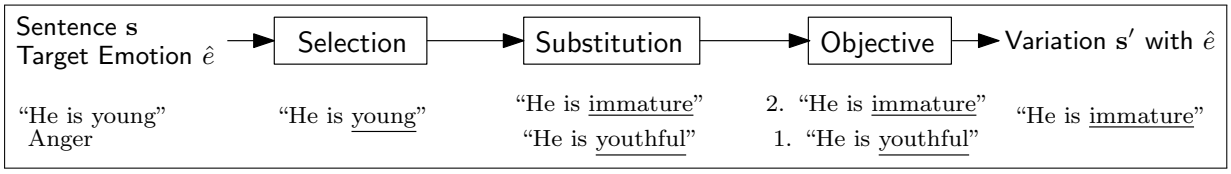


Figure 2: Pipeline model architecture. The selection module marks tokens to substitute, the substitution module retrieves candidates and perform substitution. The objective ranks and scores variations.

ber of affect-related variables (Smith et al., 2019). Other examples include text genres (Lee et al., 2019; Jhamtani et al., 2017), romanticism (Li et al., 2018), politeness/offensiveness and formality (Sennrich et al., 2016; Nogueira dos Santos et al., 2018; Wang et al., 2019).

One of the earliest methods that targets sentiment is proposed by Guerini et al. (2008), who change, add and delete sentiment-related words in a lexical substitution framework. Their strategy to retrieve candidate substitutes is informed by a thesaurus and an emotion dictionary: the first facilitates the extraction of substitutes standing in a specific semantic relation to the input words, the other allows to pick those words that have the desired valence score. Following this approach, Whitehead and Cavedon (2010) filter out ungrammatical expressions resulting from lexical substitution.

Like some works mentioned above, we adopt the view that emotions can be transfered by focusing on specific words, we use WordNet as a source of lexical substitutes, and we consider the three objectives of fluency, similarity and the presence of the target style. Moreover, we opt for a more interpretable solution than neural strategies, as we aim at pointing out what leads to a successful transfer, and what, on the contrary, prevents it.

2.3 Paraphrase Generation through Lexical Substitution

Lexical substitution received some attention independent of style transfer, as it is useful for a range of applications, like paraphrase generation and text summarisation (Dagan et al., 2006). This task, which was formulated by McCarthy and Navigli (2007) and implemented as part of the SemEval-2007 workshop, consists in finding lexical substitutes close in meaning to the original word, given its context within a sentence. The task has mainly been addressed using handcrafted and crowdsourced thesauri, such as WordNet, in order to retrieve lexical substitutes (Martinez et al., 2007; Sinha and Mihalcea, 2014; Kremer et al., 2014; Biemann, 2013). Moreover, it has been approached with distributional spaces, where the embeddings of the candidate substitutes of a target word can be found, and they can be ranked according to their similarity to the target embedding (Zhao et al., 2007; Hassan et al., 2007), as well as the similarity of their contextual information (Melamud et al., 2015)[2].

In the present paper, we follow a similar progression: we retrieve candidates for lexical substitution in WordNet; then, in our more advanced systems, we switch to embedding-based retrieval models.

3 Methods

Emotion transfer can be seen as a task in which a sentence $\mathbf{s}$ is paraphrased, and the result of this operation exhibits a different emotion than $\mathbf{s}$, specifically, a target emotion. We address emotion transfer with a pipeline in which each unit contributes to the creation of emotionally loaded paraphrases. The pipeline is shown in Figure 2. First is a *selection* component, which identifies the tokens in $\mathbf{s}$ that are to be changed. Then, the *substitution* component takes care of the actual substitution. It is responsible for finding candidate substitutes for the tokens that have been selected, producing paraphrases of the input sentence. Importantly, paraphrases are over-generated: at this stage of the pipeline, the output is likely to include sentences that do not express the target emotion. Paraphrases are then scored and re-ranked in the last, objective component, which picks up the "best" output.

3.1 Selection

This component identifies those tokens from a sentence $\mathbf{s} = t_1, \ldots t_n$ that will be substituted later, and groups them into *selections* $\mathcal{S} = \{S_i\}$, where each S_i consists of tokens, $S_i = \{t_i, \ldots, t_j\}$ $(1 \geq i, j \leq n)$. We experiment with two selection strategies, in which the maximal number of tokens

[2] A comparison of different context-aware models for lexical substitution can be found in Soler et al. (2019).

in one selection is p and the maximal number of selections is q ($p, q \in \mathbb{N}$).

Brute-Force. This baseline selection strategy picks each token separately, therefore, we obtain n selections, one for each token, i.e., $\mathcal{S} = \{\{t_1\}, \ldots, \{t_n\}\}$ ($p = 1, q = n$).

Attention-based. To pick words that are likely to influence the (current and target) emotion of a sentence, we exploit an emotion classification model to inform the selection strategy. We train a biLSTM with a self-attention mechanism (Baziotis et al., 2018) and then select those words with a high attention weight to be in the set of selections. To avoid a combinatorial explosion, we consider the k tokens with highest attention weights and add all possible combinations of up to p tokens. Therefore, $q = |\mathcal{S}| = \sum_{i=1}^{k} \binom{p}{i}$. As an example, possible selections in the sentence from Figure 1 for $k = 3, p = 2$ would be $S = \{\{\text{soul-crushing}\}, \{\text{drudgery}\}, \{\text{plagues}\}, \{\text{soul-crushing, drudgery}\}, \{\text{soul-crushing, plagues}\}, \{\text{drudgery, plagues}\}\}$.

3.2 Substitution

The selections $\mathcal{S}$ are then passed to the substitution model together with part-of-speech information. Two tasks are fulfilled by this component: substitution candidates are found for the tokens of each S_i, and the substitution is done by replacing those candidate tokens at position $i, \ldots, j$ in the input sentence $\mathbf{s}$. The next paragraphs detail our strategies for candidate retrieval. We compare a lexical semantics and two distributional semantics-based methods.

WordNet Retrieval. In the WordNet-based method (Fellbaum, 1998), we retrieve the synsets for the respective selected token with the assigned part of speech. Candidates for substitution are the neighboring synsets with the hyponym and hypernym relation (for verbs and nouns) and antonym and synonym relation (for adjectives).

Note that we do not perform word-sense disambiguation prior to retrieving the base synsets. Accordingly, the sense of the selected token in the context of the source sentence and the sense of some retrieved candidates may be different. This is in line with the design of the pipeline and we expect irrelevant forms to be penalised in the objective component.

Distributional Retrieval – Uninformed. In the "Distributional Retrieval – Uninformed" setting, we retrieve u substitution candidates based on the cosine similarity in a vector space. To build the vector space, we employ pre-trained word embeddings.[3] They are the same that are used for training the emotion classifier responsible for retrieving attention scores in the selection stage.

Distributional Retrieval – Informed. A disadvantage of the uniformed method mentioned before might be that the selected u substitutions for each token might not contain words with the targeted emotional orientation. In this approach, we slightly change the substitution selection process by first retrieving a list of u most similar tokens from the vector space. Based on this list, which is presumably of sufficient similarity to the selected token, we select those v relevant for the target emotion.

Let E be the set of emotion categories and $\hat{e} \in E$ the target emotion (with vector representation $\hat{\mathbf{e}}$). Further, let $\bar{\mathbf{e}}$ be the centroid of concepts associated with the respective emotion, as retrieved from the NRC emotion dictionary (Mohammad and Turney, 2013). From the list of semantically similar u candidates c for one token to be substituted, we select the v top scoring ones via

$$\text{score}(c, \hat{e}) = \cos(\hat{\mathbf{e}}, \mathbf{c}) - \frac{1}{|E| - 1} \sum_{\bar{e} \in E \setminus \hat{e}} \cos(\bar{\mathbf{e}}, \mathbf{c}) \,.$$

3.3 Objective

The set of candidate paraphrases produced at substitution time, based on the selections, are an overgeneration which might not be fluent, diverge from the original meaning, and might not contain the target emotion. To select those paraphrases which do not have such unwanted properties, we subselect those with the desired properties based on an objective function $f(\cdot)$ which consists of three components for fluency of the paraphrase $\mathbf{s}'$, semantic similarity between the original sentence $\mathbf{s}$ and the paraphrase $\mathbf{s}'$, and the target emotion $\hat{e}$ of the paraphrase, therefore

$$\text{f}(\mathbf{s}, \mathbf{s}', \hat{e}) = \lambda_1 \cdot \text{emo}(\mathbf{s}', \hat{e}) + \lambda_2 \cdot \text{sim}(\mathbf{s}, \mathbf{s}') + \lambda_3 \cdot \text{flu}(\mathbf{s}').$$

The paraphrase with the highest final score is selected as the result of the emotion transfer process ($\sum_i \lambda_i = 1$).

[3] 300 dimensional embeddings, available at https://github.com/cbaziotis/ntua-slp-semeval2018

Emotion Score. To obtain a score for the target emotion $\hat{e}$ we use an emotion classification model (the same as for the attention selection procedure) in which the last layer is a fully connected layer of size $|E|$ and the output layer is a softmax. Let g represent the classification model that takes a sequence of tokens $\mathbf{s}$ and an emotion e as inputs and produces the activation for e in the final layer. Therefore,

$$\text{emo}(\mathbf{s}', \hat{e}) = \frac{\exp(g(\mathbf{s}', \hat{e}))}{\sum_{e \in E} \exp(g(\mathbf{s}', e))}.$$

Similarity Score. To keep the semantic similarity as much as possible between the input sentence $\mathbf{s}$ and the candidate paraphrase $\mathbf{s}'$, we calculate the cosine similarity between the respective sentence embeddings, based on the pre-trained BERT model (Devlin et al., 2019), in the implementation provided by Wolf et al. (2019). We conceptualize BERT as a mapping function that takes a sequence of tokens $\mathbf{s}$ as input and produces a hidden vector representation for each token. The sentence embeddings $\mathbf{r}$ are obtained by averaging over all hidden vectors.[4] Therefore,

$$\text{sim}(\mathbf{s}, \mathbf{s}') = \cos(\mathbf{r}, \mathbf{r}').$$

Fluency Score. To avoid that tokens are substituted with words which do not fit in the context, we include a language model which scores the paraphrase $\mathbf{s}'$ (similar to Zhao et al., 2018). This model assesses the fluency by perplexity using GPT (Radford et al., 2018), an autoregressive neural language model based on the transformer architecture, which allows us to read the probability of the next token in a sentence given its history. We use a pretrained version of the model provided by Wolf et al. (2019). The perplexity as the average negative log probability over the tokens of our variation sentence $\mathbf{s}'$ is

$$\text{perplexity}(\mathbf{s}') = \frac{1}{n-1} \sum_{i}^{n-1} -\log(P(t_{i+1}|t_1, \ldots, t_i)).$$

Since we are dealing with negative log values, a low perplexity score indicates high probability and therefore high fluency. In order to obtain our final fluency score, we normalize the perplexity to the range $[0, 1]$ and reverse the polarity. To this end, we use the highest perplexity score ($\text{perplexity}_{\max}$) and lowest perplexity score ($\text{perplexity}_{\min}$) that we retrieve among all variation sentences created for our input sentence as scaling factors:

$$\text{flu}(\mathbf{s}') = \frac{\text{perplexity}(\mathbf{s}') - \text{perplexity}_{\max}}{\text{perplexity}_{\min} - \text{perplexity}_{\max}}$$

4 Experiments

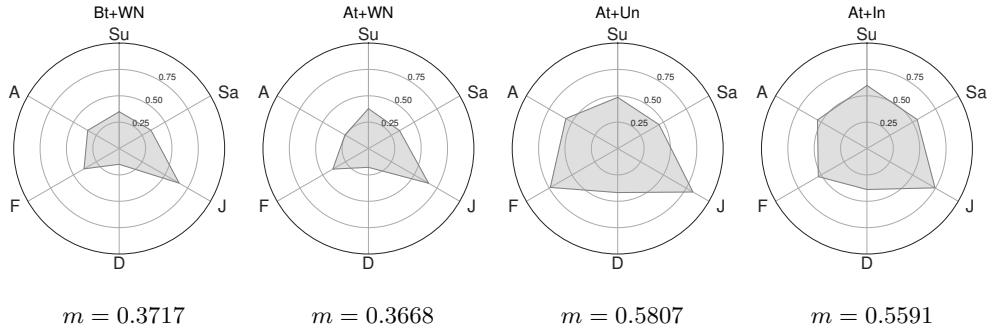
Having established the general pipeline, we move on to the question whether our strategies for selection and substitution actually produce variations with the desired emotion (RQ1). In addition, we examine the interaction between the emotion connotation of the paraphrases and their similarity to the inputs (RQ2). These questions are answered in an automatic and a human evaluation.

4.1 Setting

We instantiate and compare four model configurations for lexical substitution with different combinations of selection and substitution components. These are designed such that we can compare the selection procedure separately from the substitution component.

- **Bf+WN**: We select isolated words in the brute-force configuration and substitute those with the WordNet-based approach.
- **At+WN**: To compare if the attention mechanism is more powerful in finding relevant words to be substituted, we change the brute force selection to the attention-based method. Here, we consider the tokens with the $k = 2$ highest attention scores and combine them to selections with a maximum of $p = 2$ tokens in each selection.
- **At+Un**: We keep the attention mechanism for selection with $k = 2$ and $p = 2$, but vary the substitution component to select $u = 150$ candidates based on semantic similarity. As embedding space, we employ the same pre-trained embeddings we use for training the emotion classifier responsible for retrieving attention weights and calculating emotion scores. The number of variations created amounts to $\sum_{i=1}^{p} \binom{k}{i} u^i = 2 \cdot 150 + 1 \cdot 150^2 = 22800$.
- **At+In**: While the model configuration At+Un generates many possibly irrelevant variations, this model makes informed decisions on how to substitute: we keep the selection as in At+Un, but exchange the substitution method with the informed

[4] As recommended in the documentation of the implementation by Wolf et al. (2019) (https://huggingface.co/transformers/model_doc/bert.html, accessed on March 27, 2020), we do not use the reserved classification token [CLS] as a sentence embedding.



$m = 0.3717$ $m = 0.3668$ $m = 0.5807$ $m = 0.5591$

Figure 3: Automated evaluation results. Each radar plot shows the average emotion scores achieved by transferring 1,000 tweets to anger (A), disgust (D), fear (F), joy (J), sadness (Sa) and surprise (Su); m is the average over all emotions.

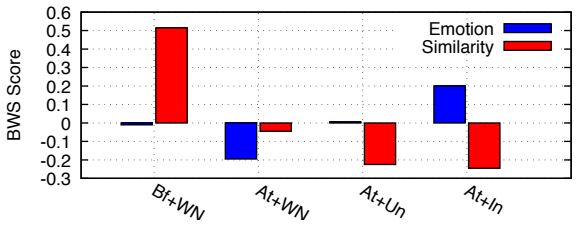


Figure 4: Results for the two human annotation trials, combined by model configuration.

strategy. Specifically, $u = 100$ candidates are found based on their semantic similarity to the token to be substituted, and among those, $v = 25$ tokens are subselected based on their emotion-informed score, leading to $\sum_{i=1}^{p} \binom{k}{i} v^i = 3 \cdot 25 + 3 \cdot 25^2 = 1950$ variations (with $k = 3$, $p = 2$). To inform this method about emotion in the embedding space, we use the NRC emotion dictionary (Mohammad and Turney, 2013).

Automatic Evaluation. The main goal of the automatic evaluation is to compare the potential of increasing the probability that the paraphrase contains the target emotion. To achieve that, we compare the four pipeline configurations, but only use the emotion score as the objective function to pick the best candidate. We use 1000 uniformly sampled Tweets from the corpus TEC (Mohammad, 2012). The emotion classification model used for scoring is trained on the same corpus using pre-trained Twitter embeddings provided by Baziotis et al. (2018).[5]. We use the attention scores obtained from this model for our attention-based selection method. As embedding space for the At+Un and At+In models, we use the same embeddings. As we transfer to the six emotions annotated in TEC, we obtain 6,000 paraphrases with At+Un and At+In and 5,904 with Bf+WN and At+WN (the latter due to non-English words which are not found in WordNet).

Human Evaluation. The goal of the human evaluation is to verify the automatic results (the potential of the selection and substitution components). Further, we compare the association of the paraphrase with the target emotion. To compare a basic setup and the most promising setup, we use $\text{emo}(\mathbf{s}', \hat{e})$ and $\text{sim}(\mathbf{s}, \mathbf{s}')$ for Bf+WN, At+WN, and At+Un and $\text{flu}(\mathbf{s}')$ in addition for At+In. This evaluation is based on 100 randomly sampled Tweets for which we ensure that they are single sentences from TEC. The annotation of emotion connotation and similarity to the original text is then setup as a best-worst-scaling experiment (Louviere et al., 2015), in which each of our two annotators is presented with one paraphrase for each of the four configurations, all for the same emotion (randomly chosen as well). Note that in contrast to best-worst scaling used for annotation as, e.g., in emotion intensity corpus creation (Mohammad et al., 2018), where textual instances are scored, here the instances change from quadruple to quadruple, but the originating configurations remain the same and receive the score. The agreement calculated with Spearman correlation of both annotators is $\rho = 1$ for the emotion connotation and $\rho = 0.8$ for semantic similarity.

4.2 Results

RQ1: Whats is the potential of emotion transfer with lexical substitution? We answer RQ1 by inspecting how likely the paraphrases are to contain the desired emotion and first turn to the automatic evaluation. Figure 3 shows the results. Each radar plot indicates the extent to which the paraphrases of each configuration express the tar-

[5] https://github.com/cbaziotis/ntua-slp-semeval2018

Text	Target
Input	surprises are great when the person is surprised !
Output for Sadness	*depresses* are great when the person is *disappointed* !
Input	love watching my daughter be so excited around christmas
Output for Anger	*detest* watching my daughter be so *annoyed* around christmas

Table 1: Examples of paraphrases produced with At+Inf for different target emotions, using all three components of the objective function.

get emotions. The average probability of the target emotion in the best paraphrases of Bf+WN is 0.3717, indicating that this method has a slightly higher potential than At+WN (0.3668); still, the shape of their plots is comparable. When we compare the substitution method while keeping the selection fixed (At+WN, At+Un, At+In), we see that the distributional methods show a clear increase (0.5807 and 0.5591 average target emotion probability).

In the manual evaluation, we see in Figure 4 (in blue) that the results are in line with the automatic evaluation. Instances originating from At+In are most often chosen as the best results, followed by At+Un and Bf+WN. At+WN scores the worst in human evaluation. Note that the best-worst-scaling results cannot directly be compared to automatic evaluation measures obtained with an automatic text classifier.

RQ2: Is semantic content preserved when changing the emotional orientation? We answer this research question based on the human annotation experiment, with the results in Figure 4. Contrary to the results on the transfer potential, Bf is judged as the most efficient selection strategy for content preservation, while At configurations are dispreferred. The ones based on distributional substitution appear to be worse compared to solutions leveraging WordNet. This shows that Bf provides a lower degree of freedom to the substitution component. The attention mechanism finds the relevant words to be substituted, but the annotators perceive these changes also as a change to the content.

To sum up, highest transfer potential is reached with a combination of attention-based selection, and distributional substitution. The fact that the latter surpasses WordNet-based retrieval may be traced back to the richness of embedding spaces, where substitution candidates can be found which have a higher semantic variability than those found in the thesaurus, and hence, have more varied emotional connotations. In addition, the distributional strategy performing better is the emotion-informed one (0.2 in Figure 4). This suggests that accessing emotion information during substitution is beneficial. The performance of this configuration is exemplified in Table 1, and further discussed in the qualitative analysis.

By comparing the two human trials, it emerges that no configuration excels in both emotion transfer and meaning preservation. In the second case, Attention-based configurations are largely downplayed by Bf+WN. Therefore, to tackle RQ2, the more a system changes emotions, the less it preserves content.

5 Analysis

We now turn to a more qualitative analysis of the results. Due to space restrictions, we show examples for the four pipeline configurations, all with the same objective function $\text{emo}(\cdot)+\text{sim}(\cdot)+\text{flu}(\cdot)$ and a comparison of the At+In model with different objective functions in supplementary material upon acceptance of this paper. Here, in Figure 2, we focus on a discussion of those cases which we consider particularly difficult, though common in everyday communication of emotions. In the selection of these examples, we follow the emotion component model of Scherer (2005) and use two examples, which correspond to a direct (explicit) communication of a subjective feeling (Ex, ID 1, 2), the description of a bodily reaction (BR, ID 3, 4), and a description of an event for which an emotion is developed based on a cognitive appraisal (Ap, ID 5, 6).

The examples which communicate an emotion directly are challenging because there is no other content available than the emotion that is described (ID 1, 2). The model has the choice to exchange two out of three words, and in nearly all cases, it choses to keep "i" and replaces the verb and the emotion word. While the latter is replaced appropriately, the verb is in most cases not substituted in a grammatically correct way. We see here that the emotion classification component in the objective function outrules the language model. This illustrates one fundamental issue with presumably all existing affect-related style transfer method: the original emotion is turned into the target emotion,

ID	Text	Type	Target
1	**I am happy**	Ex	
	i *fuck annoyed*		A
	i *dislike crabby*		D
	i *regret king*		F
	and am *happy*		J
	i am *bummed*		Sa
	i am *surprise*		Su
2	**I am sad**	Ex	
	i am *angrier*		A
	i *embarrassed disgusting*		D
	i *must lies*		F
	finally am *tiring*		J
	i *depressed* sad		Sa
	i *came realise*		Su
3	**Tears are running over my face**	BR	
	rage fuck running over my face		A
	puke are *puking* over my face		D
	shadows are *creeping* over my face		F
	gladness are running over my face		J
	depressed are *leaving* over my face		Sa
	squealed came running over my face		Su

ID	Text	Type	Target
4	**I was tembling**	BR	
	fuck irked trembling		A
	fatass reeks trembling		D
	i *hallucinated trembling*		F
	finally finally trembling		J
	bummed was *trembling*		Sa
	mom showed trembling		Su
5	**My son was standing close to the street**	Ap	
	my *fuck* was standing *annoyed* to the street		A
	my *molest* was *peeing* close to the street		D
	my *coward* was *creeping* close to the street		F
	my *yeshua* was *soaking* close to the street		J
	my *funeral* was *leaving* close to the street		Sa
	my *son* was standing *surprise* to the street		Su
6	**My grandmother died**	Ap	
	fckin grandmother *punched*		A
	ugh grandmother *farted*		D
	my *voldemort attack*		F
	my *family rededicated*		J
	cried grandmother died		Sa
	my *mama showed*		Su

Table 2: Challenging cases for different ways to communicate an internal emotion state. Inputs are in bold; all paraphrases are produced with At+Inf and all three components of the objective function. Ex: Explicit emotion mention, BR: Bodily reaction, Ap: Event appraisal.

but their intensities do not correspond.

In the examples which describe a bodily reaction (ID 3, 4), we see that the attention mechanism does not allow the words "over my face" or "trembling" to change. Instead, it finds the other words more likely to be substituted – the classifier is not informed about the meaning of "trembling" and "over my face". The substituted words make sense, but content and fluency are sacrificed again for the maximal emotion intensity available.

Similarly, the emotion classifier and therefore the associated attention mechanism do not find "close to the street" to be relevant to develop an emotion (ID 5). Instead, other words are exchanged to introduce the target emotion. These issues are mostly due to issues in the emotion classification module. Further, we see that the substitution and selection elements might have a higher chance to perform well if they considered phrases instead of isolated words.

We observe a lack of fluency in many of our output sentences, which we attribute to a dominance of the emotion classifier score. Adapting the weights of the scores in the objective might have potential, however, our findings might suggest that content, emotion and fluency are in conflict with each other – and that obtaining a particular emotion is only possible by sacrificing content similarity. Not doing so seems to lead to non-realistic utterances.

6 Conclusion & Future Work

With this paper, we introduced the task of emotion style transfer, which we have seen to be particularly difficult, on the one side due to being on the fence between content and style, and on the other side due to being a non-binary problem. Our quantitative analyses have shown that there is indeed a trade-of between content preservation and obtaining a target style and that emotion transfer is especially challenging when the text consists of descriptions of emotions in which the separation between content and style is not linguistically clear (as in "I am happy that X happened"). We propose that such test sentences based on descriptions of bodily reactions and event appraisal will be part of future test suits for emotion style transfer, in order to ensure that this task does not work well only on particular expressions of emotions.

We identified the challenge to find the right trade-of between fluency, target emotion, and content preservation. This is particularly challenging, as it would be desirable to separate the emotion intensity from our objective function. We therefore propose that intensity is handled as a fourth component in future work. This could be combined with a decoder as suggested by (Li et al., 2018). Finally, a larger-scale human evaluation should be carried out to clarify the contribution of each component.

References

Cecilia Ovesdotter Alm, Dan Roth, and Richard Sproat. 2005. Emotions from text: Machine learning for text-based emotion prediction. In *HLT-EMNLP*.

Christos Baziotis, Athanasiou Nikolaos, Alexandra Chronopoulou, Athanasia Kolovou, Georgios Paraskevopoulos, Nikolaos Ellinas, Shrikanth Narayanan, and Alexandros Potamianos. 2018. NTUA-SLP at SemEval-2018 task 1: Predicting affective content in tweets with deep attentive RNNs and transfer learning. In *SemEval*.

Chris Biemann. 2013. Creating a system for lexical substitutions from scratch using crowdsourcing. *Language Resources and Evaluation*, 47(1):97–122.

Laura Ana Maria Bostan and Roman Klinger. 2018. An analysis of annotated corpora for emotion classification in text. In *COLING*.

Ido Dagan, Oren Glickman, Alfio Gliozzo, Efrat Marmorshtein, and Carlo Strapparava. 2006. Direct word sense matching for lexical substitution. In *ACL-COLING*.

Ning Dai, Jianze Liang, Xipeng Qiu, and Xuanjing Huang. 2019. Style transformer: Unpaired text style transfer without disentangled latent representation. In *ACL*.

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

Paul Ekman. 1992. An argument for basic emotions. *Cognition and Emotion*, 6(3):169–200.

Christiane Fellbaum, editor. 1998. *WordNet: an electronic lexical database*. Language, speech, and communication. MIT Press, Cambridge, Mass.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style transfer in text: Exploration and evaluation. In *AAAI*.

Sayan Ghosh, Mathieu Chollet, Eugene Laksana, Louis-Philippe Morency, and Stefan Scherer. 2017. Affect-LM: A neural language model for customizable affective text generation. In *ACL*.

Hongyu Gong, Suma Bhat, Lingfei Wu, JinJun Xiong, and Wen-mei Hwu. 2019. Reinforcement learning based text style transfer without parallel training corpus. In *NAACL-HLT*, pages 3168–3180.

Marco Guerini, Oliviero Stock, and Carlo Strapparava. 2008. Valentino: A tool for valence shifting of natural language texts. In *LREC*.

Samer Hassan, Andras Csomai, Carmen Banea, Ravi Sinha, and Rada Mihalcea. 2007. UNT: SubFinder: Combining knowledge sources for automatic lexical substitution. In *SemEval*.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *ICML*.

Chenyang Huang, Osmar Zaïane, Amine Trabelsi, and Nouha Dziri. 2018. Automatic dialogue generation with expressed emotions. In *NAACL*.

Harsh Jhamtani, Varun Gangal, Eduard Hovy, and Eric Nyberg. 2017. Shakespearizing modern language using copy-enriched sequence to sequence models. In *Proceedings of the Workshop on Stylistic Variation*.

Evgeny Kim, Sebastian Padó, and Roman Klinger. 2017. Investigating the relationship between literary genres and emotional plot development. In *LaTeCH-CLfL*.

Roman Klinger, Orphée De Clercq, Saif Mohammad, and Alexandra Balahur. 2018. IEST: WASSA-2018 implicit emotions shared task. In *WASSA*.

Gerhard Kremer, Katrin Erk, Sebastian Padó, and Stefan Thater. 2014. What substitutes tell us - analysis of an "all-words" lexical substitution corpus. In *EACL*.

Joseph Lee, Ziang Xie, Cindy Wang, Max Drach, Dan Jurafsky, and Andrew Ng. 2019. Neural text style transfer via denoising and reranking. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 74–81.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, Retrieve, Generate: a simple approach to sentiment and style transfer. In *NAACL-HLT*.

Jordan J. Louviere, Terry N. Flynn, and A. A. J. Marley. 2015. *Best-worst scaling: theory, methods and applications*. Cambridge University Press, Cambridge, United Kingdom.

David Martinez, Su Nam Kim, and Timothy Baldwin. 2007. MELB-MKB: Lexical substitution system based on relatives in context. In *SemEval*.

Diana McCarthy and Roberto Navigli. 2007. SemEval-2007 task 10: English lexical substitution task. In *SemEval*.

Oren Melamud, Omer Levy, and Ido Dagan. 2015. A simple word embedding model for lexical substitution. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*.

Saif Mohammad. 2012. #emotional tweets. In **SEM*.

Saif Mohammad, Felipe Bravo-Marquez, Mohammad Salameh, and Svetlana Kiritchenko. 2018. SemEval-2018 task 1: Affect in tweets. In *SemEval*.

Saif M. Mohammad, Parinaz Sobhani, and Svetlana Kiritchenko. 2017. Stance and sentiment in tweets. *ACM Trans. Internet Technol.*, 17(3).

Saif M. Mohammad and Peter D. Turney. 2013. Crowdsourcing a word-emotion association lexicon. *Computational Intelligence*, 29(3):436–465.

Robert Plutchik. 2001. The nature of emotions: Human emotions have deep evolutionary roots, a fact that may explain their complexity and provide tools for clinical practice. *American Scientist*, 89(4):344–350.

Jonathan Posner, James A. Russell, and Bradley S. Peterson. 2005. The circumplex model of affect: an integrative approach to affective neuroscience, cognitive development, and psychopathology. *Development and Psychopathology*, 17(3):715–734.

Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W. Black. 2018. Style transfer through back-translation. In *ACL*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understandingby generative pre-training. Preprint. Retrieved from https://openai.com/blog/language-unsupervised/ [accessed on December 12, 2019].

Cicero Nogueira dos Santos, Igor Melnyk, and Inkit Padhi. 2018. Fighting offensive language on social media with unsupervised text style transfer. In *ACL*.

Klaus R. Scherer. 2005. What are emotions? And how can they be measured? *Social Science Information*, 44(4):695–729.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Controlling politeness in neural machine translation via side constraints. In *NAACL-HLT*.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *NIPS*.

Ravi Sinha and Rada Mihalcea. 2014. Explorations in lexical sample and all-words lexical substitution. *Natural Language Engineering*, 20(1):99–129.

Eric Michael Smith, Diana Gonzalez-Rico, Emily Dinan, and Y-Lan Boureau. 2019. Zero-shot fine-grained style transfer: Leveraging distributed continuous style representations to transfer to unseen styles. *arXiv preprint arXiv:1911.03914*.

Aina Garí Soler, Anne Cocos, Marianna Apidianaki, and Chris Callison-Burch. 2019. A comparison of context-sensitive models for lexical substitution. In *ICCS*, pages 271–282. Association for Computational Linguistics.

Zhenqiao Song, Xiaoqing Zheng, Lu Liu, Mu Xu, and Xuanjing Huang. 2019. Generating responses with a specific emotion in dialog. In *ACL*.

Yunli Wang, Yu Wu, Lili Mou, Zhoujun Li, and Wenhan Chao. 2019. Harnessing pre-trained neural networks with rules for formality style transfer. In *EMNLP-IJCNLP*.

Simon Whitehead and Lawrence Cavedon. 2010. Generating Shifting Sentiment for a Conversational Agent. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*.

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

Jingjing Xu, Xu Sun, Qi Zeng, Xiaodong Zhang, Xuancheng Ren, Houfeng Wang, and Wenjie Li. 2018. Unpaired sentiment-to-sentiment translation: A cycled reinforcement learning approach. In *ACL*.

Jake Zhao, Yoon Kim, Kelly Zhang, Alexander M Rush, and Yann LeCun. 2018. Adversarially regularized autoencoders. *ICML*.

Shiqi Zhao, Lin Zhao, Yu Zhang, Ting Liu, and Sheng Li. 2007. HIT: Web based scoring method for English lexical substitution. In *SemEval*.

Incorporating Uncertain Segmentation Information into Chinese NER for Social Media Text

Shengbin Jia[1,2], **Ling Ding**[1], **Xiaojun Chen**[1], **Shijia E**[2], **Yang Xiang**[1]
[1]Tongji University, Shanghai, China
{shengbinjia,dling,xiaojunchen,shxiangyang}@tongji.edu.cn
[2]Tencent, Shanghai, China
allene@tencent.com

Abstract

Chinese word segmentation is necessary to provide word-level information for Chinese named entity recognition (NER) systems. However, segmentation error propagation is a challenge for Chinese NER while processing colloquial data like social media text. In this paper, we propose a model (UIcwsNN) that specializes in identifying entities from Chinese social media text, especially by leveraging uncertain information of word segmentation. Such ambiguous information contains all the potential segmentation states of a sentence that provides a channel for the model to infer deep word-level characteristics. We propose a trilogy (i.e., Candidate Position Embedding ⇒ Position Selective Attention ⇒ Adaptive Word Convolution) to encode uncertain word segmentation information and acquire appropriate word-level representation. Experimental results on the social media corpus show that our model alleviates the segmentation error cascading trouble effectively, and achieves a significant performance improvement of 2% over previous state-of-the-art methods.

1 Introduction

Named entity recognition (NER) is a fundamental task for natural language processing and fulfills lots of downstream applications, such as semantic understanding of social media contents.

Chinese NER is often considered as a character-wise sequence labeling task since there are no natural delimiters between Chinese words (Liu et al., 2010; Li et al., 2014). But the word-level information is necessary for a Chinese NER system (Mao et al., 2008; Peng and Dredze, 2015; Zhang and Yang, 2018). Various segmentation features can be obtained from the Chinese word segmentation (CWS) procedures then used into a pipeline NER module (Peng and Dredze, 2015; He and Sun, 2017a; Zhu and Wang, 2019), or be co-trained by CWS-NER multi-task learning (Peng and Dredze, 2016; Cao et al., 2018).

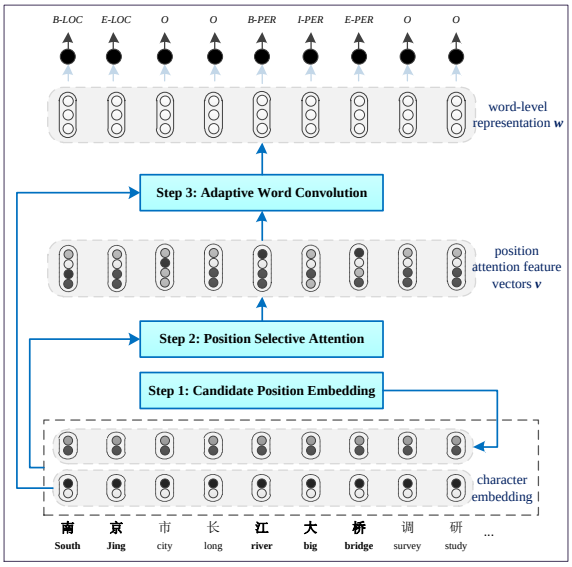


Figure 1: The architecture of our model. An interesting instance "南京市长江大桥调研(Daqiao Jiang, major of Nanjing City, is investigating)..." is represented, which is cited from (Zhang and Yang, 2018).

However, segmentation error propagation is a challenge for Chinese NER, when processing informal data like social media text (Duan et al., 2012). The CWS will produce more unreliable results on the social media text than on the formal data. Incorrectly segmented entity boundaries may lead to NER errors. Nevertheless, most existing extractors always assume that input segmentation information is affirmative and reliable without conscious error discrimination. That is, they acquiesce in that "The one supposed-reliable word segmentation output of a CWS module will be input into the NER module". Although the joint training way may improve the accuracy of word segmentations, the NER module still cannot recognize inevitable segmentation errors.

Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pages 51–60
Vancouver, Canada, July 30 - August 4, 2017.
https://doi.org/10.18653/v1/P17

To solve this problem, we design a model (UIcwsNN) that dedicates to identifying entities from Chinese social media text, by incorporating **U**ncertain **I**nformation of **C**hinese **W**ord **S**egmentation into a **N**eural **N**etwork. This kind of uncertain information reflects all the potential segmentation states of a sentence, not just the certain one that is supposed-reliable by the CWS module. Furthermore, we propose a trilogy to encode uncertain word segmentation information and acquire word-level representation, as shown in Figure 1.

In summary, the contributions of this paper are as follows:

- We embed candidate position information of characters into the model (in Section 3.1) to express the states of underlying word. And we design the Position Selective Attention (in Section 3.2) that enforces the model to focus on the appropriate positions while ignoring unreliable parts. The above operations provide a wealth of resources to allow the model to infer word-level deep characteristics, rather than bluntly impose segmentation information.

- We introduce the Adaptive Word Convolution (in Section 3.3), it dynamically provides word-level representation for the characters in specific positions, by encoding segmentations of different lengths. Hence our model can grasp useful word-level semantic information and alleviate the interference of segmentation error cascading.

- Experimental results on different datasets show that our model achieves significant performance improvements compared to baselines that use only character information. Especially, our model outperforms the previous state-of-the-art method by 2% on the social media.

2 Related Work

The NER on English has achieved promising performance by naturally integrating character information into word representations (Ma and Hovy, 2016; Peters et al., 2018; Yang et al., 2018; Yadav and Bethard, 2019; Li et al., 2020). However, Chinese NER is still underachieving because of the word segmentation problem. Unlike the English language, words in Chinese sentences are not separated by spaces, so that we cannot get Chinese words without pre-processed CWS. In particular, identifying entities on Chinese social media is harder than on other formal text since there is worse segmentation error propagation trouble. Existing methods payed little attention to this issue, and there were few entity recognition methods specifically for Chinese social media text (Peng and Dredze, 2015; He and Sun, 2017a,b).

As for the Chinese NER, existing methods could be classified as either word-wise or character-wise. The former one used words as the basic tagging unit (Ji and Grishman, 2005). Segmentation errors would be directly and inevitably entered into NER systems. The latter used characters as the basic tokens in the tagging process (Chen et al., 2006; Mao et al., 2008; Lu et al., 2016; Dong et al., 2016). Character-wise methods that outperformed word-wise methods for Chinese NER (Liu et al., 2010; Li et al., 2014).

There were two main ways to take word-level information into a character-wise model. One was to employ various segmentation information as feature vectors into a cascaded NER model. Chinese word segmentation was performed first before applying character sequence labeling (Guo et al., 2004; Mao et al., 2008; Zhu and Wang, 2019). The pre-processing segmentation features included character positional embedding (Peng and Dredze, 2015; He and Sun, 2017a,b), segmentation tags (Zhang and Yang, 2018; Zhu and Wang, 2019), word embedding (Peng and Dredze, 2015; Liu et al., 2019; E and Xiang, 2017) and so on. The other was to train NER and CWS tasks jointly to incorporate task-shared word boundary information from the CWS into the NER (Xu et al., 2013; Peng and Dredze, 2016; Cao et al., 2018). Although co-training might improve the validity of the word segmentation, the NER module still had no specific measures to avoid segmentation errors. The above existing methods suffered the potential issue of error propagation.

A few researchers tried to address the above defect. Luo and Yang (2016) used multiple word segmentation outputs as additional features to a NER model. However, they treated the segmentations equally without error discrimination. Liu et al. (2019) introduced four naive selection strategies to select words from the pre-prepared Lexicon for their model. However, these strategies did not consider the context of a sentence. Zhang

and Yang (2018) proposed a Lattice LSTM model that used the gated recurrent units to control the contribution of the potential words. However, as shown by Liu et al. (2019), the gate mechanism might cause the model to degenerate into a partial word-based model. Ding et al. (2019) and Gui et al. (2019) proposed the models with graph neural network based on the information that the gazetteers or lexicons offered. Obtaining large-scale, high-quality lexicons would be costly. They were dedicated to capturing the correct segmentation information but might not alleviate the interference of inappropriate segmentations.

It is worth mentioning that the above methods were not specifically aimed at social media. We propose a method to learn word-level representation by leveraging uncertain word segmentation information while considering the informal expression characteristics of social media text.

3 Methodology

Figure 1 illustrates the overall architecture of our model UIcwsNN. Given a sentence $S = \{c_1, c_2, \cdots, c_n\}$ as the sequence of characters, each character will be assigned a pre-prepared tag.

We use a Conditional random fields (CRF) layer to decode tags according to the outputs from the sequence encoder (Lample et al., 2016; Yang et al., 2018).

As for the sequence encoding, we use the convolution operation as our basic encoding unit. The colloquial social media text usually does not have normative grammar or syntax and presents semantics in fragmented form, for example, "有好多好多的话想对你说李巾凡想要瘦瘦瘦成李帆我是想切开云朵的心(Have many many words to say to you Jinfan Li wanna thin thin thin to Fan Li I am a heart that want to cut the cloud)". These properties will destroy the propagation of temporal semantic information that comes with the textual sequence. Therefore, the Convolutional neural network (CNN) is naturally suitable for encoding colloquial text because it specializes in capturing salient local features from a sequence.

More importantly, we use a trilogy to learn the word-level representation by incorporating uncertain information of Chinese text segmentation, as shown in the following details.

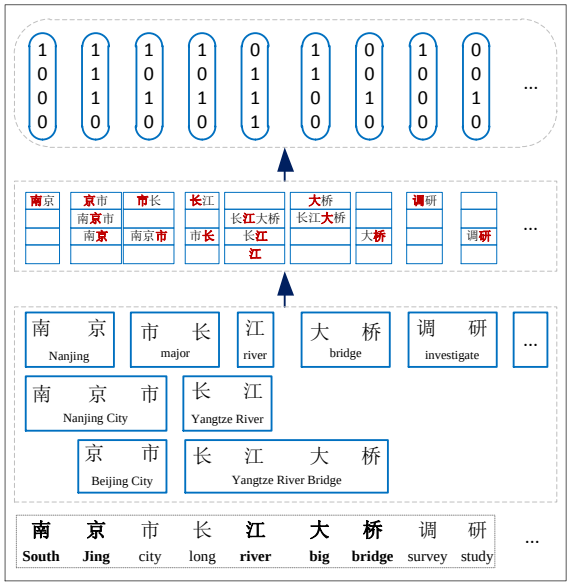


Figure 2: Create the candidate position embedding.

3.1 Step-1: Candidate Position Embedding

We design the candidate position embedding to represent candidate positions of each character in all potential words. It reflects the states of all underlying segmentation in a sentence.

We firstly scan all the potential words in the sentence that can be worded [1], so as to obtain as much meaningful segmentation states as possible. As shown in the bottom part of Figure 2, the instance can be segmented and obtained candidate segmentations: "南京(Nanjing), 京市(Jing City), 南京市(Nanjing City), 市长(major), 长江(Yangtze River), 江(river), 大桥(bridge), 长江大桥(Yangtze River Bridge), 调研(investigate), ...".

Next, we use a 4-dimensional vector $\mathbf{c}_i^{(p)}$ to embed candidate position information of a character, where each dimension indicates the positional candidate (i.e., Begin, Inside, End, Single) of a character in words. 1 if it exists, 0 otherwise. For example, as shown in middle and top parts of Figure 2, as "京(Jing)" being the begin of "京市(Beijing City)", the inside of "南京市(Nanjing City)", and the end of "南京(Nanjing)", the 1^{st}, 2^{nd} and 3^{rd} dimensions of the embedding of "京(Jing)" are 1, but the 4^{th} dimension is 0 (i.e., $[1, 1, 1, 0]$).

The correct segmentation sequence for the example should be "南京(Nanjing)/市长(major)/江大桥(Daqiao Jiang)/调研(is investigating)/...". However, the one certain segmentation output that

[1] We use the "Jieba", a popular python packages for the CWS. Its special function "cut_for_search()" can achieve this operation. (https://github.com/fxsjy/jieba)

is supposed-reliable by the above CWS tool is "南京市(Nanjing City)/长江大桥(Yangtze River Bridge)/调研(investigates)/...". The errors may cause that the entity "江大桥(Daqiao Jiang)" is not recognized. In contrast, the candidate position embedding should be a more reasonable representation for the Chinese sentence segmentation. It is flexible for a model to infer word-level characteristics.

3.2 Step-2: Position Selective Attention

There should be only one certain position for a character in the given sentence. We design the position selective attention over candidate positions. It enforces the model to focus on the most relevant positions while ignoring unreliable parts.

Each sequence S is projected to an attention matrix $\boldsymbol{A}$ that captures the semantics of position features interaction according to the contexts.

$$\boldsymbol{A} = \tanh(\boldsymbol{W}^{(a)}[\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_n]), \quad (1)$$

where $\boldsymbol{A}$ is a matrix of $n \times 4$, $\boldsymbol{W}$ is trainable parameters.

We apply a set of convolution operations that involve filters $\boldsymbol{W}^{(c)}$ and bias terms $b^{(c)}$ to the sequence to learn a representation $\boldsymbol{h}_i$ for character c_i.

$$\boldsymbol{h}_i = [\boldsymbol{h}_i^{l=2}; \boldsymbol{h}_i^{l=3}; \boldsymbol{h}_i^{l=4}; \boldsymbol{h}_i^{l=5}], \quad (2)$$

$$\boldsymbol{h}_i^l = \text{relu}(\boldsymbol{W}_l^{(c)}[\boldsymbol{x}_i, \cdots, \boldsymbol{x}_{i+l-1}] + b_l^{(c)}), \quad (3)$$

where $\mathbf{h}_i^l$ represents a feature that is generated from a window of length l started with c_i. The $\boldsymbol{x}_i$ is the combination of character embedding $\boldsymbol{c}_i^{(e)}$ and expanded candidate position embedding, as

$$\boldsymbol{x}_i = \boldsymbol{c}_i^{(e)} + \boldsymbol{W}^{(p)}\boldsymbol{c}_i^{(p)}, \quad (4)$$

where $\boldsymbol{c}_i^{(e)} \in \mathbb{R}^{d_e}$, $\boldsymbol{W}^{(p)} \in \mathbb{R}^{4d_p}$. To enhance the learning of the position information assisted by the character semantic information, we ensure $d_e \leqslant d_p$.

Given the matrix $\boldsymbol{A}$, we define

$$\boldsymbol{v}_i = \frac{\exp(\boldsymbol{A}_{i,j})}{\sum_{j=0}^{3}\exp(\boldsymbol{A}_{i,j})}, \quad (5)$$

to quantify the reliability of the j^{th} position with respect to the i^{th} character.

The position attention feature vectors $\boldsymbol{v}$ should assign higher attention values to the appropriate positions while minimizing the values of disturbing positions.

	南 South	京 Jing	市 city	长 long	江 river	大 big	桥 bridge	调 survey	研 study
$subw_{i-3:i}$									
$subw_{i-2:i}$							✓		
$subw_{i-1:i}$		✓		✓		✓			✓
$subw_i$									
$subw_{i:i+1}$	✓		✓			✓		✓	
$subw_{i:i+2}$					✓				
$subw_{i:i+3}$									

Figure 3: Display the tabulation of subwords. The red vertical lines identify correct word segmentations. The ✓ shows the subwords that fit each character.

3.3 Step-3: Adaptive Word Convolution

Based on the position selection of each character, the step-3 encodes word segmentations to obtain complete word-level semantics.

As for each character c_i, we expect to encode the segmentation that involves the c_i as its word-level representation. There is a challenge: The lengths of word segmentations are diverse, and the positions of characters located in segmentations are flexible. A single encoding structure is difficult to adapt to this situation. Therefore, we propose the adaptive word convolution.

When c_i is the k^{th} character of the word w, we design the word to consist of two parts, namely, the left subword and the right subword, in the form

$$\begin{aligned} & w_{m:m+h-1} \\ \Leftrightarrow\ & subw_{m:i} \oplus subw_{i:m+h-1} \\ \Leftrightarrow\ & subw_{(i-k):i} \oplus subw_{i:(i+h-1-k)}, \end{aligned} \quad (6)$$

where $1 \leqslant m \leqslant n$, $1 \leqslant h \leqslant 4$, [2] $m \leqslant i \leqslant m+h$, and $0 \leqslant k < h$, $\oplus$ denotes join operation. For the instance mentioned above, we expect to get the tabulation, as shown in Figure 3. For example, the "南(South)" is the first (i.e., $k = 0$) character of the word "南京"(Nanjing) (i.e. $i = m = 1$ and $h = 2$), we can use the left $subw_{1:1}$ and the right $subw_{1:2}$ to express the word $w_{1:2}$, and then as the word-level representation for the character "南(South)". Especially, we discard the $subw_{1:1}$ beacuse $subw_{1:2}$ contains it.

To model subwords automatically, we learn a feature map $\boldsymbol{F}$ ($n \times 7$) through a set of convolution operations with windows of different directions and

[2] In most cases, Chinese words are no longer than 4 characters.

different sizes, as

$$\boldsymbol{F} = \begin{bmatrix} \overleftarrow{\boldsymbol{sw}_1^3} & \overleftarrow{\boldsymbol{sw}_2^3} & \cdots & \overleftarrow{\boldsymbol{sw}_n^3} \\ \overleftarrow{\boldsymbol{sw}_1^2} & \overleftarrow{\boldsymbol{sw}_2^2} & \cdots & \overleftarrow{\boldsymbol{sw}_n^2} \\ \overleftarrow{\boldsymbol{sw}_1^1} & \overleftarrow{\boldsymbol{sw}_2^1} & \cdots & \overleftarrow{\boldsymbol{sw}_n^1} \\ \boldsymbol{sw}_1^0 & \boldsymbol{sw}_2^0 & \cdots & \boldsymbol{sw}_n^0 \\ \overrightarrow{\boldsymbol{sw}_1^1} & \overrightarrow{\boldsymbol{sw}_2^1} & \cdots & \overrightarrow{\boldsymbol{sw}_n^1} \\ \overrightarrow{\boldsymbol{sw}_1^2} & \overrightarrow{\boldsymbol{sw}_2^2} & \cdots & \overrightarrow{\boldsymbol{sw}_n^2} \\ \overrightarrow{\boldsymbol{sw}_1^3} & \overrightarrow{\boldsymbol{sw}_2^3} & \cdots & \overrightarrow{\boldsymbol{sw}_n^3} \end{bmatrix}, \quad (7)$$

$$\overleftarrow{\boldsymbol{sw}_i^k} = \text{relu}(\boldsymbol{W}_k^{(s)}[\boldsymbol{z}_{i-k}, \cdots, \boldsymbol{z}_i] + b_k^{(s)}), \quad (8)$$

$$\overrightarrow{\boldsymbol{sw}_i^k} = \text{relu}(\boldsymbol{W}_k^{(s')}[\boldsymbol{z}_i, \cdots, \boldsymbol{z}_{i+k}] + b_k^{(s')}), \quad (9)$$

$$\boldsymbol{z}_i = \boldsymbol{c}_i^{(e)} + \boldsymbol{W}^{(v)}\boldsymbol{v}_i, \quad (10)$$

where $\boldsymbol{W}^{(v)} \in \mathbb{R}^{d_v}$, the $\rightarrow$ indicates the windows sliding forward, whereas $\leftarrow$ shows the windows sliding backward.

Based on the candidate position distribution of characters learned from the step-2, our model can adaptively separate valid subwords from the $\boldsymbol{F}$ to learn the word-level representation $\boldsymbol{w}_i$, in detail,

$$\boldsymbol{w}_i = \sum_{f=0}^{6} \alpha_{if} \boldsymbol{F}_{i,f}, \quad (11)$$

$$\alpha_{if} = \frac{\exp(\text{g}(\boldsymbol{F}_{i,f}, \boldsymbol{v}_i))}{\sum_{f=0}^{6} \exp(\text{g}(\boldsymbol{F}_{i,f}, \boldsymbol{v}_i))}, \quad (12)$$

$$\text{g}(\boldsymbol{F}_i, \boldsymbol{v}_i) = \tanh(\boldsymbol{W}^{(\alpha)}[\boldsymbol{F}_i + \boldsymbol{W}^{(v)}\boldsymbol{v}_i]). \quad (13)$$

After performing the trilogy, the model can grasp useful word-level semantic information and avoid the trouble of segmentation error cascading.

4 Experiments

4.1 Settings

Datasets. We evaluate Chinese NER models on two popular datasets. The ***WeiboNER*** corpus (Peng and Dredze, 2015; He and Sun, 2017a), is drawn from Chinese social media. It contains 1,890 Sina Weibo messages annotated with four entity types ([PER]SON, [ORG]ANIZATION, [LOC]ATION, and [GEO]POLITICAL), including named entities (NAM) and nominal mentions (NOM). The ***MSRA*** dataset (Levow, 2006), is in the formal text domain. There are 50,729 annotated sentences with three entity types (PER, ORG, and LOC). We use the BIOES scheme (Begin, Inside, Outside, End, Single) to indicate the position of the token in an entity (Ratinov and Roth, 2009).

Evaluation. We measure the performance of models by regarding three complementary metrics, Precision (P), Recall (R), and F1-measure (F). Each experiment will be performed five times under different random seeds to reduce the volatility of models. Then we report the mean and standard deviation for each model.

Hyperparameters. The character embedding is pre-trained on the raw microblog text [3] by the word2vec [4], and its dimension is 100. As for the base model BiLSTM+CRF, we use hidden state size as 200 for a bidirectional LSTM. As for the base model CNNs+CRF, we use 100 filters with window length $\{2, 3, 4, 5\}$. We tune other parameters and set the learning rate as 0.001, dropout rate as 0.5. We randomly select 20% of the training set as a validation set. We train each model for a maximum of 120 epochs using Adam optimizer and stop training if the validation loss does not decrease for 20 consecutive epochs. Besides, we set $d_e = d_p = 100$ and $d_v = 25$. We also experiment with other settings and find that these are the most reasonable.

4.2 Results and Detailed Analysis

4.2.1 Ablation Study

To study the contribution of each component in our model, we conducted ablation experiments on the two datasets where we use the product of each step to decode tags. We display the results in Table 1 and draw the following conclusions.

The feature (CS) is generated from the one certain segmentation output that is supposed-reliable by the CWS tool Jieba, and it may not benefit the NER on social media text. Compared with the corresponding baseline, the feature (CS) impels the model to improve its performance on the MSRA dataset but to reduce performance on the WeiboNER corpus. There are more segmentation errors on social media text than on formal text so that the impact of error cascading is heavy for NER on social media.

On the WeiboNER dataset, the three steps exert

[3] http://www.nlpir.org/download/weibo.7z

[4] https://radimrehurek.com/gensim/models/word2vec.html

Table 1: Results of ablation experiments on the WeiboNER dataset and MSRA dataset. The base model is the CNNs+CRF.

Models	WeiboNER			MSRA		
	P	R	F ±std	P	R	F ±std
character embedding (baseline)	66.45	53.47	59.22 ±0.42	87.11	85.84	86.47 ±0.21
+ certain segmentation feature (CS)	68.41	51.82	58.92 ±0.54	90.37	88.06	89.20 ±0.12
+ candidate position embedding (CPE)	65.19	56.46	60.51 ±0.37	90.20	88.27	89.22 ±0.06
+ position selective attention (PSA)	68.50	55.31	61.13 ±0.49	90.34	89.08	89.71 ±0.22
+ adaptive word convolution (AWC)	67.37	57.61	62.07 ±0.61	89.87	90.54	90.20 ±0.24
base model + BERT	78.01	72.97	75.40 ±0.33	94.51	91.72	93.09 ±0.27
UIcwsNN + BERT	79.64	73.29	76.33 ±0.20	96.31	94.98	95.64 ±0.15

different capabilities for improving model performance. Compared with the baseline, the model with the step-1 (+CPE) yields 1.3% improvement in the F value, and its recall improves significantly by 3%, although the precision decreases 1.2%. After we continue with the step-2 (+ PSA), the F value further increases by 0.6%. In this scenario, both precision and recall are higher than the baseline. When the step-3 (+AWC) is completed, the F value further increases by 0.9%. In this scenario, the recall significantly improves by 4% with 0.9% improvement in precision, compared to the baseline.

Combining the results on the two different datasets, we find several consistent phenomena. Globally, the F values of the model keep increasing after each step. From a decomposition perspective, the step-2 (+PSA) is notable for improving the precision of the model. And the step-3 (+AWC) is significant for improving the recall. Therefore, the trilogy is complementary.

Our method has good robustness. On the two datasets from different domains, the uncertain information of word segmentations is always efficient, the trilogy (i.e., +CPE, +PSA, +AWC) is valuable. However, performance improvement on the WeiboNER dataset is more significant than on the MSRA dataset. In contrast with formal text, the social media text contains more word segmentation errors that better reflects the advantages of our method.

Finally, We verify the influence of the pre-trained language model BERT (Devlin et al., 2018) on our model. We optimize the BERT [5] to obtain the character embedding and train the model CNNs+CRF jointly, where its F value reaches 75% on the WeiboNER dataset. The BERT improves the entity recognition outcome dramatically since it uses large-scale external data to pre-train the contextual embedding. When we use our model UIcwsNN to replace the base model CNNs+CRF, the effect is improved by nearly 1%. It proves that our trilogy and the BERT are complementary. The BERT can provide high-quality character-level embedding to the model, and our method contributes word-level semantic information for the model. This conclusion can also be drawn from the results of the MSRA dataset.

[5] https://storage.googleapis.com/bert_models/2018_11_03/chinese_L-12_H-768_A-12.zip

Table 2: The F values of existing models on the WeiboNER dataset. * indicates that the model utilizes external lexicons. ° indicates that the model adopts joint learning. The previous models do not use the BERT, so we show the results of our model without BERT.

Models	NAM	NOM	Overall
(Peng and Dredze, 2015)°	51.96	61.05	56.05
(Peng and Dredze, 2016)°	55.28	62.97	58.99
(He and Sun, 2017a)	50.60	59.32	54.82
(He and Sun, 2017b)	54.50	62.17	58.23
(Zhang and Yang, 2018)*	53.04	62.25	58.79
(Cao et al., 2018)°	54.34	57.35	58.70
(Zhu and Wang, 2019)	55.38	62.98	59.31
(Liu et al., 2019)*	52.55	**67.41**	59.84
(Ding et al., 2019)*	-	-	59.50
(Gui et al., 2019)*	55.34	64.98	60.21
(Johnson et al., 2020)	55.70	62.80	59.50
BiLSTM+CRF	53.95	62.63	57.69
CNNs+CRF	55.07	62.97	59.22
Our model (UIcwsNN)	**57.58**	**65.97**	**62.07**

4.2.2 Comparison with Existing Methods

Table 2 represents the results of the WeiboNER dataset. Our model UIcwsNN significantly outperforms other models and achieves new state-of-the-

Table 3: The results of different models on the MSRA dataset. × indicates that the model uses the BERT.

Model	P	R	F
(Chen et al., 2006)	91.22	81.71	86.20
(Dong et al., 2016)	91.28	90.62	90.95
(Zhang and Yang, 2018)	93.57	92.79	93.18
(Zhu and Wang, 2019)	93.53	92.42	92.97
(Ding et al., 2019)	94.60	94.20	94.40
(Zhao et al., 2019)×	95.46	95.09	95.28
(Gong et al., 2019)×	95.26	**95.57**	95.42
(Johnson et al., 2020)	93.71	92.29	92.99
UIcwsNN	89.87	90.54	90.20
UIcwsNN + BERT×	**96.31**	**94.98**	**95.64**

art performance. The overall score of our model is generally more than 2% higher than the scores of other models. Many methods use lexicon instead of the CWS to provide extractors with external word-level information, but how to choose the appropriate words based on sentence contexts is their challenge. Besides, the approaches that jointly train NER and CWS tasks do not achieve desired results, because segmentation noises affect their effectiveness inevitably. Our model handles this trouble.

The CNN-based models achieve better performance compared to the model BiLSTM+CRF. Furthermore, most of the existing methods construct encoders based on recurrent neural networks or graph neural networks. Although they perform excellent results on the MSRA dataset, they do not achieve a significant improvement on the WeiboNER corpus. In addition to the word segmentation error propagation on social media, another important reason may be that the fragmented semantic expression of colloquial text limits their performance. In contrast, our CNN-based model plays a better advantage in capturing the fragmented semantics of colloquial text.

Results on the MSRA dataset are shown in Table 3. Our model UIcwsNN specializes in learning word-level representation, but rarely considers other-levels characteristics, such as long-distance temporal semantics. Therefore, it only achieves competitive performance on the formal text. But our model UIcwsNN+BERT realizes new state-of-the-art performance.

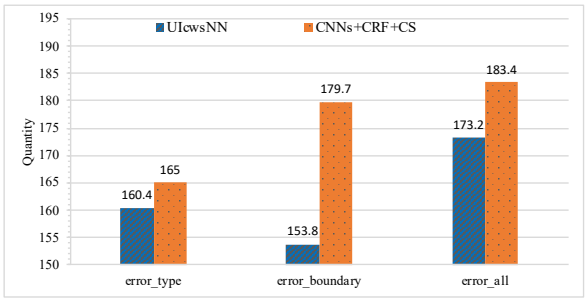


Figure 4: The statistics of the model output errors on the WeiboNER corpus. The model CNNs+CRF+CS uses the feature of the one supposed-reliable word segmentation output from the CWS tool Jieba.

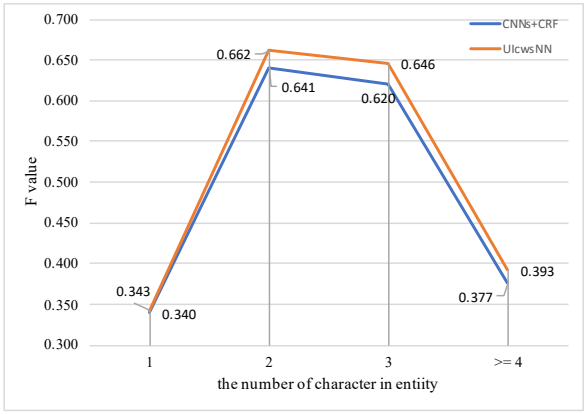


Figure 5: Performance of multi-character entities on the WeiboNER dataset. The base model CNNs+CRF only uses character embedding.

4.2.3 Error Analysis

We count the output errors of models and classify them into two categories [6]: type error and boundary error, as shown in Figure 4. The model CNNs+CRF+CS produces more boundary errors than type errors. However, our model UIcwsNN dramatically decreases the boundary error outputs (and the type errors are also reduced), so that the error distribution is reversed. That is, in model UIcwsNN, the proportion of boundary errors is smaller than that of type errors, but in model CNNs+CRF+CS, the opposite is true. This situation shows that word segmentation errors generated by the word segmentation tool seriously affect model performance, especially misleading the model to identify wrong entity boundaries. Our method can learn the word boundaries effectively, thereby alleviating the cascade of segmentation errors.

[6] If there are two kinds of errors on a predicted entity, the error will be counted twice.

Table 4: Testing examples with segmentation errors.

Case One	有人祝我早生贵[女]$_{PER.NOM}$真是无语啊 Someone wished me to have a precious daughter soon, I am so speechless
candidate segmentation	有人(someone), 祝(wish), 我(me), 早(soon), 早生(early birth), 生贵(precious), 贵(precious), 女(daughter), 女真(Nuzhen), 是(is), 真是(really), 无语(speechless), 啊(ah)
one certain segmentation	有人(someone), 祝(wish), 我(me), 早(soon), 生贵(precious), 女真(Nuzhen), 是(is), 无语(speechless), 啊(ah)
Case Two	刚刚获得了微博[准会员]$_{PER.NOM}$专属徽章，开心 I just got the exclusive badge for a weibo associate member, I am happy
candidate segmentation	刚刚(just now), 获得(get), 了(finish), 微博(weibo), 微博准(wei bo zhun), 准会(quasi), 会员(member), 专属(exclusive), 徽章(badge), 开心(happy)
one certain segmentation	刚刚(just now), 获得(get), 了(finish), 微博准(wei bo zhun), 会员(member), 专属(exclusive), 徽章(badge), 开心(happy)

4.2.4 Performance against Multi-character Entities

Figure 5 shows the performance of recognizing entities with different lengths $\{1, 2, 3, \geqslant 4\}$. According to statistics, entities with two or three characters account for more than 95% of the total number of entities. Both models give high F scores for entities of moderate lengths $\{2, 3\}$, but low performance for entities that are too short or too long. The reasons may be that entities with a single character or more than four characters are rare, resulting in model training inadequately. Our model UIcwsNN achieves better results than the base model CNNs+CRF when identifying entities of various lengths. In particular, as for entities with two or three characters, the model UIcwsNN yields more than 2% improvement. This situation implies that our model captures word-level semantic information by modeling the uncertain information of word segmentations so that it is good at recognizing multi-character entities.

4.2.5 Case Study

Table 4 shows several examples with word segmentation errors. When we use the one certain (supposed-reliable) segmentation sequence from the tool Jieba as the word-level feature for the model CNNs+CRF+CS, the segmentation errors "女真'(Nuzhen)" and "微博准(wei bo zhun)" lead to the misjudgments of the entities "女(daughter)" and "准会员(associate member)", respectively. Our model UIcwsNN can extract these entities. The uncertain character positions can provide our model with rich word-level information. Then, we use the position selective attention to support the model to learn appropriate segmentation states.

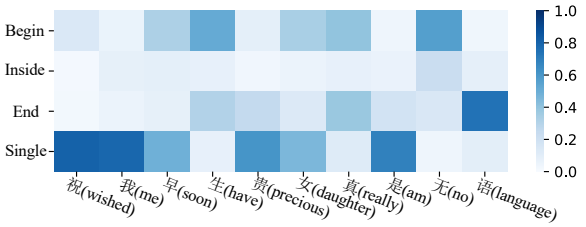


Figure 6: Visualization of position attention values $\boldsymbol{v}$ obtained from the position selective attention.

The visualization of the first case in Figure 6 shows that our model can assign higher attention values to the appropriate positions while mitigating error interferences.

5 Conclusion

Named entity recognization is an urgent task for semantic understanding of social media content. As for the Chinese NER, Chinese word segmentation error propagation is prominent since there is much colloquial text in social media. In this paper, we explore a trilogy to leverage the uncertain information of word segmentation to avoid the interference of segmentation errors. The step-1 utilizes the Candidate Position Embedding to present the potential segmentation states of a sentence; The step-2 employs the Position Selective Attention to capture appropriate segmentation states while ignoring unreliable parts; The step-3 uses the Adaptive Word Convolution to encode word-level representation dynamically. We analyze the performance of each component of the model and discuss the relationship between the model and related factors such as segmentation error, BERT, and entity length. Experiment results on different datasets

show that our model achieves new state-of-the-art performance. It demonstrates that our method has an excellent ability to capture word-level semantics and can alleviate the segmentation error cascading trouble effectively. In future work, we hope that the model can get rid of the word segmentation tool, instead, learn the candidate position informationn autonomously. We will release the source code when the paper is openly available.

Acknowledgments

This work is supported by the National Key Research and Development Project of China under Grant no. 2019YFB1704402, the 2019 Tencent Marketing Solution Rhino-Bird Focused Research Program, and the 2020 Tencent Rhino-Bird Elite Training Program.

References

Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, and Shengping Liu. 2018. Adversarial transfer learning for chinese named entity recognition with self-attention mechanism. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 182–192.

Aitao Chen, Fuchun Peng, Roy Shan, and Gordon Sun. 2006. Chinese named entity recognition with conditional probabilistic models. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 173–176.

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*.

Ruixue Ding, Pengjun Xie, Xiaoyan Zhang, Wei Lu, Linlin Li, and Luo Si. 2019. A neural multi-digraph model for chinese NER with gazetteers. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019*, pages 1462–1467.

Chuanhai Dong, Jiajun Zhang, Chengqing Zong, Masanori Hattori, and Hui Di. 2016. Character-based lstm-crf with radical-level features for chinese named entity recognition. In *Natural Language Understanding and Intelligent Applications*, pages 239–250. Springer.

Huiming Duan, Zhifang Sui, Ye Tian, and Wenjie Li. 2012. The cips-sighan clp 2012 chineseword segmentation onmicroblog corpora bakeoff. In *Proceedings of the second CIPS-SIGHAN joint conference on Chinese language processing*, pages 35–40.

Shijia E and Yang Xiang. 2017. Chinese named entity recognition with character-word mixed embedding. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 2055–2058. ACM.

Cheng Gong, Jiuyang Tang, Shengwei Zhou, Zepeng Hao, and Jun Wang. 2019. Chinese named entity recognition with bert. *DEStech Transactions on Computer Science and Engineering*, (cisnrc).

Tao Gui, Yicheng Zou, Qi Zhang, Minlong Peng, Jinlan Fu, Zhongyu Wei, and Xuanjing Huang. 2019. A lexicon-based graph neural network for chinese ner. 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 1039–1049.

Honglei Guo, Jianmin Jiang, Gang Hu, and Tong Zhang. 2004. Chinese named entity recognition based on multilevel linguistic features. In *International Conference on Natural Language Processing*, pages 90–99. Springer.

Hangfeng He and Xu Sun. 2017a. F-score driven max margin neural network for named entity recognition in chinese social media. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 713–718.

Hangfeng He and Xu Sun. 2017b. A unified model for cross-domain and semi-supervised named entity recognition in chinese social media. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Heng Ji and Ralph Grishman. 2005. Improving name tagging by reference resolution and relation detection. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 411–418.

Shardrom Johnson, Sherlock Shen, and Yuanchen Liu. 2020. Cwpc_biatt: Character–word–position combined bilstm-attention for chinese named entity recognition. *Information*, 11(1):45.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of NAACL-HLT*, pages 260–270.

Gina-Anne Levow. 2006. The third international chinese language processing bakeoff: Word segmentation and named entity recognition. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 108–117.

Haibo Li, Masato Hagiwara, Qi Li, and Heng Ji. 2014. Comparison of the impact of word segmentation on name tagging for chinese and japanese. In *LREC*, pages 2532–2536.

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*.

Wei Liu, Tongge Xu, Qinghua Xu, Jiayu Song, and Yueran Zu. 2019. An encoding strategy based word-character LSTM for Chinese NER. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2379–2389.

Zhangxun Liu, Conghui Zhu, and Tiejun Zhao. 2010. Chinese named entity recognition with a sequence labeling approach: based on characters, or based on words? In *International Conference on Intelligent Computing*, pages 634–640. Springer.

Yanan Lu, Yue Zhang, and Dong-Hong Ji. 2016. Multi-prototype chinese character embedding. In *LREC*.

Wencan Luo and Fan Yang. 2016. An empirical study of automatic chinese word segmentation for spoken language understanding and named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 238–248.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1064–1074.

Xinnian Mao, Yuan Dong, Saike He, Sencheng Bao, and Haila Wang. 2008. Chinese word segmentation and named entity recognition based on conditional random fields. In *Proceedings of the Sixth SIGHAN Workshop on Chinese Language Processing*.

Nanyun Peng and Mark Dredze. 2015. Named entity recognition for chinese social media with jointly trained embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 548–554.

Nanyun Peng and Mark Dredze. 2016. Improving named entity recognition for chinese social media with word segmentation representation learning. In *The 54th Annual Meeting of the Association for Computational Linguistics*, page 149.

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.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155.

Yan Xu, Yining Wang, Tianren Liu, Jiahua Liu, Yubo Fan, Yi Qian, Junichi Tsujii, and Eric I Chang. 2013. Joint segmentation and named entity recognition using dual decomposition in chinese discharge summaries. *Journal of the American Medical Informatics Association*, 21(e1):e84–e92.

Vikas Yadav and Steven Bethard. 2019. A survey on recent advances in named entity recognition from deep learning models. *arXiv preprint arXiv:1910.11470*.

Jie Yang, Shuailong Liang, and Yue Zhang. 2018. Design challenges and misconceptions in neural sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3879–3889.

Yue Zhang and Jie Yang. 2018. Chinese ner using lattice lstm. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1554–1564.

H. Zhao, M. Xu, and J. Cao. 2019. Pre-trained language model transfer on chinese named entity recognition. In *2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pages 2150–2155.

Yuying Zhu and Guoxin Wang. 2019. Can-ner: Convolutional attention network for chinese named entity recognition. 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 3384–3393.

Multi-Task Supervised Pretraining for Neural Domain Adaptation

Sara Meftah[◇], **Nasredine Semmar**[◇], **Mohamed Ayoub Tahiri**[◇]
Youssef Tamaazousti[∓], **Hassane Essafi**[◇], **Fatiha Sadat**[+]
[◇]CEA, LIST, Laboratoire Analyse Sémantique Texte et Image, France
[∓]MIT, CSAIL, USA
[+]UQÀM, Montréal, Canada
{firstname.lastname}@cea.fr, ytamaaz@mit.edu, sadat.fatiha@uqam.ca

Abstract

Two main transfer learning approaches are used in recent work in NLP to improve neural networks performance for under-resourced domains in terms of annotated data. 1) *Multi-task learning* consists in training the task of interest with related tasks to exploit their underlying similarities. 2) *Mono-task pretraining*, where target model's parameters are pretrained on large-scale labelled source domain and then fine-tuned on labelled data from the target domain (the domain of interest). In this paper, we propose a new approach that takes advantage of both approaches by learning a hierarchical model trained across multiple tasks from the source domain, then fine-tuned on multiple tasks from the target domain. Our experiments on four NLP tasks applied to social media texts show that our proposed method leads to significant improvements compared to both approaches.

1 Introduction

Deep learning approaches are powerful when dealing with large amounts of annotated data. However, these are only available for a few languages and domains due to the cost of the manual annotation (Duong, 2017). Particularly, despite the valuable importance of Social Media's content for a variety of applications (*e.g.* public security, health monitoring, or trends highlight), this large domain is still poor in terms of annotated data. Furthermore, to build systems for such applications, the machine might understand many low and high-level tasks. Therefore, a model, able to recognise as many linguistic properties as possible from a given sentence, is required.

Many attempts have been done to exhibit the performance of NLP models in low-resource scenarios. Particularly, two dominant approaches of neural Transfer Learning (TL) (Pan and Yang, 2010) are used in the State-Of-The-Art (SOTA): 1) *Mono-Task Pretraining (MTP)*: Sequential Transfer Learning (STL) (Ruder, 2019), performed in two stages: *pretraining* on a rich source-domain on enough training examples and then, *fine-tuning* on the available few target-domain examples. This approach has proved to be powerful in many NLP tasks, outperforming the classic supervised learning paradigm, because it takes benefit from pre-learned knowledge. And 2) *Multi-Task Learning (MTL)* (Caruana, 1997), that showed many benefits in several NLP tasks and applications, consists of training different tasks simultaneously, leveraging learned knowledge from related problems and resulting richer representations with higher generalisation (Collobert and Weston, 2008).

We introduce in this paper a novel method, that we call **Mu**lti-**T**ask **S**upervised **P**re-training and **Ad**aptation (**MuTSPad**), which unifies both approaches discussed above. MuTSPad takes benefit from both, by learning a hierarchical multi-task model trained across multiple tasks from the source-domain, and further fine-tuned on multiple tasks from the target-domain. Hence, in addition to diverse linguistic properties learned from various supervised NLP tasks, MuTSPad takes advantage of the pre-learned knowledge from the high-resource source-domain.

We demonstrate the effectiveness of our approach on domain adaptation from the high-resource News-domain to the low-resourced Tweets-domain. We carry out experiments on four NLP tasks, from the low-level Part-of-Speech tagging (POS) and Chunking (CK) to the higher-level Named Entity Recognition (NER) and Dependency Parsing (DP). MuTSPad exhibits significantly better performance than both TL approaches and is highly competitive compared to best SOTA methods.

Furthermore, to the best of our knowledge, there are no available common datasets containing annotations for all the above-mentioned tasks, nei-

Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pages 61–71
Vancouver, Canada, July 30 - August 4, 2017.
https://doi.org/10.18653/v1/P17

ther for the News-domain or the Tweets-domain. Though, many early works had highlighted the intricacy of multi-task training from heterogeneous datasets (Subramanian et al., 2018). Thus, we propose to build multi-task datasets for the News and Tweets domains, by unifying the aforementioned task-independent datasets.

2 Related Work

Our work is related to two lines of research, Sequential Transfer Learning and Multi-Task Learning. In the following, we briefly present the SOTA of each one. Then, we discuss some papers from the literature with a loosely close idea to multi-task pretraining and fine-tuning.

Sequential Transfer Learning (STL) is a TL setting performed in two stages: *Pretraining* and *Adaptation*. The purpose behind using STL techniques for NLP can be divided into two main research areas, "universal representations" and "domain adaptation". The former aims to build neural features transferable and beneficial to a wide range of NLP tasks and domains. *e.g.* ELMo (Peters et al., 2018), BERT (Devlin et al., 2019), etc. The second aims to harness the knowledge represented in features learned on a source domain (high-resourced in most cases) to improve learning a target domain (low-resourced in most cases) (Zennaki et al., 2016, 2019). The source and the target problems may differ on the task, the language or the domain. For instance, cross-lingual adaptation has been explored for sentiment analysis (Chen et al., 2016) and cross-domain adaptation has been applied for POS models adaptation from News to Tweets domain (Gui et al., 2017; Meftah et al., 2017, 2018). Our research falls into the second research area, since we aim, as the last two works, to transfer the knowledge learned when training on the News-domain to improve the Tweets-domain's training.

Multi-Task Learning (MTL) consists in a joint learning of related tasks and thus leverages training signals generated by different tasks (Caruana, 1997). The advantage of using MTL over independent task learning has been shown in many NLP tasks and applications (Lin et al., 2018). The processing (training) order of examples from different tasks (datasets), also called *Scheduling*, is particularly studied in the literature. This could be implicit or explicit (Jean et al., 2018). The formal include, for instance, affecting different learning rates for task-specific parameters. While the second, modify the importance of each task statically or dynamically. *e.g.* Kiperwasser and Ballesteros (2018) proposed variable schedules that increasingly favour the principal task over batches and Jean et al. (2018) proposed adaptive schedules that vary according to the validation performance of each task during training.

Multi-Task Pretraining and Fine-tuning: Multi-task pretraining has been especially explored for learning universal representations (Conneau et al., 2017; Ahmad et al., 2018). Multi-task fine-tuning was recently explored to fine-tune BERT pre-trained model in a multi-task fashion on multiple tasks (Liu et al., 2019). Furthermore, in term of using multi-task features for domain adaptation, Søgaard and Goldberg (2016) showed the benefit of multi-task learning for domain adaptation from News-domain to Weblogs-domain for CK task, when disposing CK's supervision only for the source-domain, and lower-level POS supervision for the target-domain. Finally, in terms of unifying multi-task learning and fine-tuning, Kiperwasser and Ballesteros (2018) proposed to improve machine translation with the help of POS and DEP tasks by scheduling tasks during training, starting with multi-tasking of the principal task with auxiliary lower-level tasks (POS and DEP), and as the training graduates, the model trains only to the main task. However, to the best of our knowledge, performing pretraining and fine-tuning on multi-task models for domain adaptation has not been explored in the literature.

3 Model Architecture

3.1 Sequence Labelling Architecture

Regarding the exact architecture of each task, POS, CK and NER tasks are Sequence Labelling (SL) tasks. Given an input sentence of n successive tokens $[w_1, \ldots, w_n]$, SL predicts the tag $c_i \in \mathcal{C}$ of every w_i, with $\mathcal{C}$ being the tag-set.

We followed the literature (Ma and Hovy, 2016; Yang et al., 2018) and used a common SL architecture, including three main components: (i) a Word Representation Extractor (**WRE**), (ii) a Features Extractor (**FE**) and (iii) a Classifier (**Cl**). **WRE** computes, for each token w_i, a word and a character-level biLSTMs encoder-based embeddings (respectively, $\mathbf{we_i}$=$\mathbf{WE}(w_i)$ and $\mathbf{ce_i}$=$\mathbf{CE}(w_i)$), and concatenates them to get a final representation $\mathbf{x_i}$=($\mathbf{we_i}$,$\mathbf{ce_i}$). **WRE**'s out-

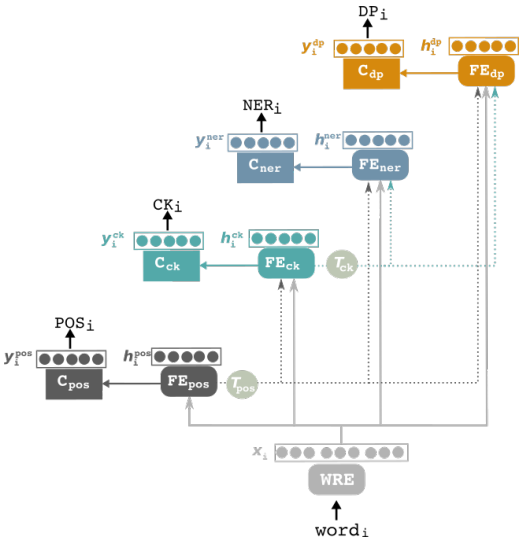


Figure 1: Illustrative scheme of our Hierarchical multi-task architecture.

puts $[x_1, \ldots, x_n]$ are fed into the $\mathbf{FE}$ that outputs a context sensitive representation for each token, consisting of a single biLSTMs layer which iteratively passes through the sentence in both directions. Finally, $\mathbf{Cl}$ consists of a fully-connected layer (denoted Ψ) that classifies every given x_i following:

$$\hat{y}_{w_i} = (\mathbf{C} \circ \mathbf{FE} \circ \mathbf{WRE})(w_i). \quad (1)$$

3.2 Dependency Parsing Architecture

For the DP branch, a similar procedure is applied, except that, compared to previous tasks, DP is a a harder problem and thus requires a more complex model. We followed Qi et al. (2018) and used their "neural arc-factored graph-based dependency parser", which is based on the "Deep bi-Affine parser" (Dozat and Manning, 2016)). Indeed, given an input sentence of n successive tokens $[w_1, \ldots, w_n]$, the goal of DP is two folds: 1) identifying, for each w_i, its head $w_j \in S$. The couple of tokens w_i and w_j are called the dependent and the head, respectively. Then, 2) predicting the dependency syntactic relation's class $r_j^i \in \mathcal{R}_{dp}$ relating each dependent-head pair, where $\mathcal{R}_{dp}$ being the dependency-relations set. More precisely, for each token w_i we predict its out-going labelled arc (w_i, w_j, r_j^i). Thus, constructing a syntactic tree structure of the sentence, where words are treated as nodes in a graph, connected by labelled directed arcs.

Hence, as in SL models, the DP architecture is composed of a $\mathbf{WRE}$ followed by a $\mathbf{FE}^{dp}$ and a $\mathbf{Cl}^{dp}$. Except that $\mathbf{Cl}^{dp}$ consists of *four* classifiers, producing four distinct vectors for representing the word: (i) as a dependent seeking its head; (ii), as a head seeking all its dependants; (iii), as a dependent deciding on its relation; and (iv), as a head deciding on the labels of its dependants. These representations are then passed to biAffine softmax classifiers.

3.3 Hierarchical Multi-Task Architecture

As mentioned above, POS, CK, NER and DP are the four tasks considered in this work. As we aim to learn a multi-task model where the four tasks are learned jointly, the architecture of our model contains a common branch as well as four exits, one per task. Also, as the tasks are hierarchically related to each other, we adopted a *hierarchical* architecture (similar to Hashimoto et al. (2017) and Sanh et al. (2019)). More specifically, we organised the four tasks from low-level to high-level, with each task being fed with a shared word embedding as well as the outputs of all the lower tasks. To construct that hierarchy of tasks, we followed some linguistic hints from the literature. Indeed, many works have shown that POS improves CK (Yang et al., 2017; Ruder12 et al., 2019); NER benefits from POS (Meftah and Semmar, 2018; Ruder, 2019) and CK (Collobert and Weston, 2008); and DP profits from POS and CK (Hashimoto et al., 2017). In simple terms, POS and CK are considered as "universal helpers" (Changpinyo et al., 2018). Thus, based on these linguistic hierarchy observations, we feed POS features to CK; then POS and CK features to both NER and DP.

An illustration of our multi-task hierarchical model is given in Fig.1. More specifically, $\mathbf{WRE}$ is shared across all tasks, its output (namely $\mathbf{x_i} = \mathbf{WRE}^{shared}(w_i)$) is fed to all branches. The lower component of the POS tagging branch ($\mathbf{FE^{pos}}$) is fed with the shared embedding and after processing, it outputs BiLSTMs features $\mathbf{h_i^{pos}}$. This is then fed into the POS classifier $\mathbf{C^{pos}}$ to calculate predictions through:

$$\hat{y}_{w_i}^{pos} = (\mathbf{C^{pos}} \circ \mathbf{FE^{pos}})(\mathbf{x_i}) \quad (2)$$

These POS features ($\mathbf{h_i^{pos}}$) as well as the shared embeddings $\mathbf{x_i}$ are then fed to the CK branch that outputs a probability distribution for the CK tag-set as follows:

$$\hat{y}_{w_i}^{ck} = (\mathbf{C^{ck}} \circ \mathbf{FE^{ck}})(\mathbf{x_i}, \mathbf{T}^{pos}(\mathbf{h_i^{pos}})) \quad (3)$$

Note that, rather than directly using $\mathbf{FE}^{pos}$'s output, we first reduce its dimensionality by applying

a learnable FC layer transformation denoted $\mathbf{T}^{pos}$.

In the same vein, following our hierarchy, the shared embedding, plus the output features of POS ($\mathbf{h_i^{pos}}$) as well as the output features of CK ($\mathbf{h_i^{ck}}$) are fed to the NER branch that outputs one class probability per named entity. Formally, this is computed using:

$$\hat{y}_{w_i}^{ner} = (\mathbf{C^{ner}} \circ \mathbf{FE^{ner}})(\mathbf{x_i}, \mathbf{T}^{pos}(\mathbf{h_i^{pos}}), \mathbf{T}^{ck}(\mathbf{h_i^{ck}})) \quad (4)$$

For DP, similarly to the NER branch, the shared embedding, plus the output features of POS as well as the output features of CK are fed to $\mathbf{FE}^{dp}$, followed by $\mathbf{C}^{dp}$ which outputs:

$$\hat{y}_{w_i}^{dp} = (\mathbf{C^{dp}} \circ \mathbf{FE^{dp}})(\mathbf{x_i}, \mathbf{T}^{pos}(\mathbf{h_i^{pos}}), \mathbf{T}^{ck}(\mathbf{h_i^{ck}})) \quad (5)$$

4 Our Approach

In low-resources scenarios, two approaches are commonly used to alleviate the lack of training data:

1. "Mono-Task Pre-training": pre-training the neural model (supervisedly or unsupervisedly) on a rich source-task before updating its weights on the task of interest (target task);

2. "Multi-task learning": training the task of interest jointly with other auxiliary tasks with labelled data that might force the network to learn useful features.

The first approach is known to work very well since a while, and yielded impressive results in recent years but the last approach seems to struggle as it still faces the problem of annotated data scarcity.

In this paper, to make sure that we learn useful features that are relevant for the tasks of our interest, we propose an approach that combines pre-training and multi-task learning, and thus takes benefits from the rich source-domain, and especially all its available annotated data and tasks. We called our method **Mu**lti-**T**ask **S**upervised **P**re-training and **Ad**aptation (**MuTSPad**). This is described in details in Sec. 4.1, before we describe (in Sec.4.2) how we alleviated the problem of datasets heterogeneity in multi-task learning (*i.e.*, only mono-task labelled datasets are available).

4.1 MuTSPad: Multi-Task Supervised Pretraining and Adaptation

MuTSPad roughly consists in pretraining on a large annotated *multi-task source* dataset and then fine-tuning it on the *multi-task target* dataset, that corresponds to that task of interest. As supervised and unsupervised pretraining, MuTSPad alleviates the lack of annotated data by taking benefit from rich source-domains. However, compared to them it does the pre-training on *multiple* tasks, and not only one. This brings even more real supervision to the network and thus gives more chance to end up with more features. Also important, as source-domains are usually richer than target-domains, we might always find source-datasets that are labelled exactly with all the tasks we want to solve in the target-domain. This enforces the network to learn only features that might be relevant for our tasks of interest, and thus avoid filling up the network with irrelevant features.

More specifically, we considered four commonly used tasks in this work, POS, CK, NER and DP. In classical multi-task scenario all sentences have all needed tasks annotation (i.e. an annotation dataset for each task). However, in our case we have two datasets, the first for News domain and the second for Tweets domain, both are heterogeneous, having one task-annotation per sentence. In contrary to the classical multi-task scenario that assumes having one dataset where words and sentences are labelled for all the tasks, in reality, this is very hard to encounter. Thus, let us consider a more realistic scenario: a set of datasets is given, with each dataset being labelled to only one task. For instance, one dataset is labelled with POS, the other with NER. Note that, though the first is labelled with POS only, it might certainly contain named entities. For this reason, we call this scenario "Heterogeneous multi-task learning". The difficulties of learning in such a scenario are described in details in Sec. 4.2.

Back to MuTSPad, let us assume a set of tasks $\mathcal{T}$, and one dataset $\mathcal{D}_i$ per task $\mathcal{T}_i$. A source multitask model $\mathcal{M}_s$ (described in Fig.1) is first trained on these heterogeneous *source*-datasets $\mathcal{D}^S$. And the set of all parameters learned for each task $\mathcal{T}_i$ as well as task-agnostic ones, are then used to initialise the target multi-task model, denoted $\mathcal{M}_t$. All the weights of this last $\mathcal{M}_t$ are then adapted on the set of target-datasets $\mathcal{D}^T$. Note that, though the sets of target-tasks $\mathcal{T}^T$ and source-tasks $\mathcal{T}^S$ might

be the same, their label-sets might differ, thus as in classical fine-tuning, the weights of each task-classifier are randomly initialised. However, for the labels that might be the same, we initialise the weights of the target task-classifiers with those pre-trained on the source-domain.

In terms of loss functions, as in classical multi-task learning, we minimise the weighted sum of each task loss:

$$\mathcal{L} = \frac{\sum_{j=1}^{j=N} \alpha_{task_j} \times \mathcal{L}_{task_j}}{N} \quad (6)$$

Where α_{task_j} represents task-weight and N is tasks number. As we used a hierarchical model for reasons mentioned in Sec. 3.3, we propose to focus the early-stage training on low-level tasks and progressively increase the focus on higher-level ones (along the same line of thought of Kiperwasser and Ballesteros (2018)). However, unlike (Kiperwasser and Ballesteros, 2018) who modified tasks sampling during training, we propose to tune loss calculation minimisation. During training, tasks weights α_{pos}, α_{ck}, α_{ner} and α_{dp} are respectively set up to: $[1, 0.5, 0.25, 0.25]$, and doubled at each epoch until $\alpha = 1$.

4.2 Heterogeneous Multi-Task Learning

As mentioned in the previous section, we mostly face the heterogeneous multi-task learning scenario, where only one task-labels might be assigned to a dataset. In that case, the classical multi-task learning approach is not directly applicable, thus we propose to use the "Scheduling process" (Zaremoodi et al., 2018; Lu et al., 2019) (described in the following paragraph). However, since training with different datasets for each task remains difficult (Subramanian et al., 2018)) we proposed "Dataset Unification" a much simpler and easy to learn method for that scenario.

4.2.1 Tasks Scheduling Procedure

To deal with this heterogeneous aspect, we first use simple frozen uniform scheduling, which we call "one task per batch", similar to (Zaremoodi et al., 2018) and (Lu et al., 2019), where at each iteration of the training process, the task to train is selected randomly. Specifically, the base steps of "one task per mini-batch" scheduling process are as follows: 1) picking a mini-batch of samples from only one particular task and 2) updating *only* the parameters corresponding to the selected task, as well as the subsequent tasks (including the task-agnostic parameters). Thus, at every step only one task is trained. We successively pick all the tasks following a constant ordering strategy "from lower-level to higher-level tasks" (Hashimoto et al., 2017): POS then CK then NER then DP. Thus, every 4 steps, the model sees all the tasks once and learns their corresponding parameters once.

4.2.2 Datasets Unification

To overcome the intricacy of "tasks scheduling process", we propose to construct a *unified dataset* by combining several sources of independent textual annotations. Furthermore, since we are interested in benefiting from pretraining and fine-tuning, we apply unification process on both, source and target-domains.

These datasets contain samples of a broad range of heterogeneous annotations in a variety of contexts (initially sentences are labelled only with one task rather than all), making the multi-task training challenging. Thus, to circumvent this problem, we propose to *unify* the Twitter-domain datasets to form a unified Tweets dataset that we call **TweetAll**. We do the same with standard News-domain datasets to form a unified multi-task dataset that we name **EnglishAll**. Concretely, we enrich the gold annotations of each task with an automatic annotation by applying on its training-set our baseline Mono-Task Learning model of the other 3 tasks. In the end, we obtain two unified datasets one for Tweet (**TweetAll**) and one for English (**EnglishAll**). Thus, in both datasets each sentence is labelled with all tasks (one label is the initial manual annotation and three are generated automatically). Consequently, using our unified datasets brings us to the classical multi-task scenario, where each sentence is annotated with all tasks, thus at each iteration, all tasks are learned and thus all multi-task model's parameters are updated.

5 Experiments

5.1 Domains, Tasks and Datasets

As mentioned above, we conducted experiments on four tasks: two low-level tasks (**POS** and **CK**) and two higher-level ones: (**NER** and **DP**). In terms of domain, data and annotations for the **source-datasets**, we used the *standard English domain* and chose the following datasets: The *WSJ* part of Penn-Tree-Bank (PTB) (Marcus et al., 1993) for POS, annotated with the PTB tag-set; *CONLL2003* for

Task	Classes	Sources	Eval. Metrics	Splits (train - val - test)
POS: POS Tagging	36	WSJ	Top-1 Acc.	912,344 - 131,768 - 129,654
CK: Chunking	22	CONLL-2000	Top-1 Exact-match F1.	211,727 - n/a - 47,377
NER: Named Entity Recognition	4	CONLL-2003	Top-1 Exact-match F1.	203,621 - 51,362 - 46,435
DP: Dependency Parsing	51	UD-English-EWT	Top-1 LAS.	204,585 - 25,148 - 25,096
POS: POS Tagging	17	TweeBank	Top-1 Acc.	24,753 - 11,742 - 19,112
CK: Chunking	18	TChunk	Top-1 Exact-match F1.	10,652 - 2,242 - 2,291
NER: Named Entity Recognition	6	WNUT	Top-1 Exact-match F1.	62,729 - 15,734 - 23,394
DP: Dependency Parsing	51	TweeBank	Top-1 LAS.	24,753 - 11,742 - 19,112

Table 1: Statistics of the datasets we used to train our multi-task learning models. **Top**: datasets of the source domain (called "EnglishAll"). **Bottom**: datasets of the target domain (called "TweetAll").

NER (Sang and De Meulder, 2003); *CONLL2000* (Sang et al., 2000) for CK; and finally *UD-English-EWT* (Nivre et al., 2016) for DP.

In the same vein, for the **target-datasets**, we used the *Tweets domain* and the following datasets: the recent *TweeBank* (Liu et al., 2018) for POS, annotated with the PTB universal tag-set; *WNUT-17* from emerging entity detection shared task (Derczynski et al., 2017) for NER; **TChunk** (Ritter et al., 2011) for CK; and the data annotated with Universal dependency relations in the *TweeBank* dataset for DP[1]. Detailed statistics of all the datasets are summarised in Table 1.

To evaluate our models, we use the accuracy (acc) for POS, Exact-match F1[2] (Li et al., 2020) for NER and CK and labelled attachment score (LAS) (Nivre et al., 2004).

5.2 Comparison methods

5.2.1 Baselines

We compare our method to multiple baselines, that we separate into 4 categories according to the pre-training method.

Without Pretraining: Training from scratch on the Tweets (target-domain) datasets.

- **Mono-Task Learning**: an independent training of our mono-tasks models (one model per task) on every target task separately.
- **Multi-Task Learning**: a *joint training* of our multi-task model described in Sec.3.3 (one model for all the tasks) on all the tasks from target-domain.

Unsupervised pretraining: we replace the **WRE** component in *Mono-Task Learning* by the pre-trained model[3] ELMo (Embeddings from Language Models) (Peters et al., 2018), consisting of a CNNs-based character-level representations followed by a 2-layer LSTMs. Thus, **ELMo** with the randomly initialised **FE** and **Cl** are further trained on the target-domain tasks. Specifically, we run experiments with two ELMo models: 1) $\mathbf{ELMo}^{small}$: the small pre-trained model (13.6M parameters) on 1 billion word benchmark. 2) $\mathbf{ELMo}^{large}$: the big pre-trained model (93.6M parameters) on 5.5 billion word benchmark.

Supervised pretraining on the source-domain of the network on each task independently then fine-tuning *on the same task* in the Tweets domain. This method is called **Mono-Task Pre-Training**. A variant of it is marked with *, and consists of just pretraining, *i.e.*, without fine-tuning. Note that this variant is possible only when target dataset has the same tagset as the source dataset.

Adversarial pretraining (Ganin et al., 2016; Gui et al., 2017) is particularly used for domain adaptation, that aims to reduce the shift between the source and target domains at the pretraining stage. Precisely, in parallel to task's objective trained on supervised annotations from the source domain, an adversarial objective with respect to a domain discriminator is trained on unsupervised target data to minimise the distance between source and target representations. Followed by a fine-tuning *on the same task* in the Tweets domain.

5.2.2 State-Of-The-Art (SOTA)

We compare our approach to the best SOTA performances for each task:

- **BiAffine** (Dozat and Manning, 2016): we report the LAS score for DP reported by Liu et al. (2018). Note that, in addition to word-level and character embeddings, which we use in our model to represent words, they use predicted POS labels and lemmas as input.
- **Flairs** (Akbik et al., 2019): For NER, using

[1]Note that TweeBank dataset is already anonymised. For TChuck and WNUT datasets, we used simple rules to anonymise usernames and URLs.

[2]SeqEval package were used to calculate F1 metric.

[3]https://allennlp.org/elmo

Method	PreTraining	POS (acc)	DP (LAS)	NER (F1)	CK (F1)	mNRG
SOTA - BiAffine(Dozat et al., 2017)	n/a	n/a	77.7	n/a	n/a	n/a
SOTA - PretRand (Meftah et al., 2019)	n/a	94.95	n/a	n/a	n/a	n/a
SOTA - Flairs (Akbik et al., 2019)	n/a	n/a	n/a	49.59	n/a	n/a
SOTA - MDMT (Mishra, 2019)	n/a	92.44	n/a	49.86	87.85	n/a
SOTA - DA (LSTM) (Gu and Yu, 2020)	n/a	n/a	n/a	n/a	84.58	n/a
SOTA - DA (BERT$_{BASE}$) (Gu and Yu, 2020)	n/a	n/a	n/a	n/a	87.03	n/a
SOTA - DA (BERT$_{LARGE}$) (Gu and Yu, 2020)	n/a	n/a	n/a	n/a	87.53	n/a
Best SOTA	n/a	94.95	77.7	49.86	87.85	n/a
Mono-task Learning	none	91.58	67.48	36.75	80.26	0.0
Multi-Task Learning		91.98	71.16	38.98	81.66	5.9
ELMo^{small}	Unsupervised	92.51	69.12	41.57	84.28	9.3
ELMo^{large}		94.02	69.76	**44.95**	85.56	19.9
Mono-Task Pre-Training*	Supervised	n/a	76.92	n/a	70.16	n/a
Mono-Task Pre-Training		93.33	78.21	41.25	84.64	21.5
Adversarial Pre-Training	Adversarial	93.47	77.49	41.68	84.75	22.6
MuTSPad (best)	MultiTask, Sup.	**94.53**	**80.12**	43.34	**85.77**	**31.5**

Table 2: **Results on the TweetAll test-sets for the four tasks**. On the second column, we describe the pretraining type (none, supervised, unsupervised, adversarial and our multi-task supervised). The last column (mNRG that states for median Normalised Relative Gain) aggregates relevantly the scores of the methods across tasks.

a BiLSTM-CRF sequence labelling architecture, fed with Pooled Contextual Embeddings, pre-trained on character-level language models.

- **PretRand** (Meftah et al., 2019): a neural model based on a transfer learning approach, it improves *Mono-Task Pre-Training* baseline by reducing the bias occurring in the pre-trained neurons.

- **Multi-dataset-multi-task (MDMT)** (Mishra, 2019): multi-task learning, based on pre-trained ELMo embeddings, of 4 NLP tasks: POS, CK, super sense tagging and NER, on 20 Tweets datasets 7 POS, 10 NER, 1 CK, and 2 super sense–tagged datasets.

- **Data Annealing (DA)** (Gu and Yu, 2020): a fine-tuning approach similar to *Mono-Task Pre-Training* baseline, but the passage from pretraining to fine-tuning is performed gradually, *i.e.* the training starts with only formal text data (News) at first; then, the proportion of the informal text data (Tweets) is gradually increased during the training process.

5.3 Implementation details

The hyper-parameters (HP) we used are as follows. For **The task-agnostic WRE:** The dimensions of character embedding = 50, hidden states of the character-level biLSTM = 100 and word-level embeddings (updated during training) = 300 (these latter are pre-loaded from Glove pre-trained vectors (Pennington et al., 2014) and fine-tuned during training). For **Sequence labelling branches**: we use a single-layer biLSTM (token-level feature extractor), with dimension = 200. **DP branch HP**: we follow Stanford parser' (//github.com/stanfordnlp/stanfordnlp) HP configuration. **Global HP:** In all experiments, SGD was used for training with early stopping and mini-batches were set to 16 sentences.

5.4 Results

5.4.1 Comparison to SOTA & baselines

Our experimental results are reported in Table 2. Clearly, MuTSPad strongly outperforms baselines, and is very competitive with the best SOTA results. We detail our main findings below:

Multi-Task Learning baseline enhances the performances of all tasks compared to *mono-task learning*. Obviously, it is most benefactor for DP by ∼3.5% since DP is highly influenced by POS labels, while it is least benefactor for POS by ∼0.5%.

Unsupervised pretraining: Clearly, incorporating pre-trained ELMo representations performs better compared to *mono-task learning*. Particularly for NER task with ∼+8% by **ELMo$^{\mathbf{large}}$**. We also found that it improves the other tasks but not with the same order of improvement as for NER, which we mainly attribute to the fact that contextual representations pre-trained on language modelling capture more semantic features. Particularly, we find that DP gains the least from ELMo compared to the other syntactic tasks.

Comparing MuTSPad to baselines: MuTSPad outperforms both TL methods, Multi-Task Learning and Mono-Task Fine-Tuning, on all data-sets, by ∼+26 and ∼+10, respectively, on mNRG (median

Method	POS	DP	NER	CK
w/o unif.	94.08	79.17	**43.34**	84.87
w/ source unif.	94.36	79.67	43.21	**85.77**
w/ source+target unif.	**94.53**	**80.12**	40.65	85.71

Table 3: Impact of Datasets Unification on MuTSPad.

Normalised Relative Gain; a well suited metric for multi-task (Tamaazousti et al., 2019)). Compared to unsupervised pretraining, we can observe that MuTSPad outperforms ELMo on POS, CK and DP, where $\mathbf{Elmo}^{\mathbf{large}}$ brought higher performances for NER. Note that ELMo is complementary to our approach, hence, we expect higher performances when incorporating $\mathbf{Elmo}^{\mathbf{large}}$ to MuTSPad.

Comparing MuTSPad to SOTA: Our score on DP is about $\sim$2.5% higher than best SOTA scores. For POS, CK and NER experiments, we achieve lower scores than SOTA. It is noteworthy that, first, contrary to our approach, all these methods are mono-task models (except MDMT), *i.e.*, unable to solve other tasks. Second, NER and CK best SOTA used pretrained contextualised representations that harness the performance, namely, Flais embeddings by Akbik et al. (2019), ELMo by Mishra (2019) and BERT by Gu and Yu (2020).

5.4.2 Impact of Datasets Unification

We report in Tab.3 MuTSPad's results:

1) **w/o unif.** : training on independent datasets, using the "one batch per task" scheduling rule. on both stages, pretraining and fine-tuning

2) **w/ source unif.** : In the pretraining stage, training is performed on unified EnglishAll dataset. While in fine-tuning, training is performed on independent datasets.

3) **w/ source+target unif.** : In both pretraining and finetuning stages, training is performed on unified EnglishAll and TwitterAll datasets, respectively.

Clearly, pretraining on unified source datasets (w/source unif) slightly improved performances on all tasks. Nevertheless, finetuning on unified target datasets (w/source+target unif) is beneficial only for POS and DP tasks, while it strongly hurts NER's performance. We mainly attribute this to the NER's "Mono-task learning" model's low performance on Tweets leading to noisy NER automatic predictions. It is noteworthy that, using unified datasets is easier to train, making training convergence faster.

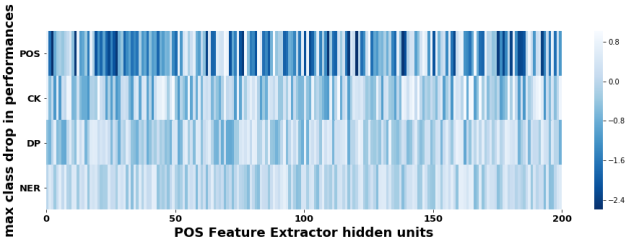


Figure 2: Maximum drop on class-score for each task when ablating individual units from the POS Feature Extractor output ($\mathbf{h}^{\mathbf{pos}}$). Dark/light blue: high/low drop. One can see that it is the POS task that is most impacted by the POS units.

5.5 Low-Level Tasks Importance Analysis

In this section, we investigate how low-level tasks impact high-level tasks in our hierarchical multi-task model (See Fig.1). Specifically, we focus on the impact of $\mathbf{h}^{\mathbf{pos}}$, the representation encoded by the POS task, for CK, NER and DP tasks.

For this purpose, we quantify the importance of $\mathbf{h}^{\mathbf{pos}}$ individual units for POS, CK, NER and DP performances. Assuming that ablating the most important units for a task should bring higher drop in performance compared to the least important units, we perform an individual ablation (also called pruning) of $\mathbf{h}^{\mathbf{pos}}$ units (neurons), as in (Zhou et al., 2018; Dalvi et al., 2019).

Given the already trained target multi-task model $\mathcal{M}_t$, we set the relating weights of each $unit_i$ from $\mathbf{h}^{\mathbf{pos}}$ to zero, *i.e.* $\mathbf{T}^{\mathbf{pos}}$ weights for CK, NER and DP; and $\mathbf{Cl}^{\mathbf{pos}}$ weights for POS. Hence, the ablated unit will not contribute to the final prediction for any input word. Then, with one unit ablated at a time, we launch the inference on each-task's dev-set, then compute the resulting score-drop for each class, leading to a matrix per task $\mathbf{A}^{task} \in \mathbf{M_{d,m}}(\mathbb{R})$, where d is $\mathbf{h}^{\mathbf{pos}}$'s dimension and m is the number of task' classes. This matrix can be summarised in a *max-class-score-drop* vector $\mathbf{v}^{\mathbf{task}} \in \mathbb{R}^d$, where each $\mathbf{v}_{\mathbf{i}}^{\mathbf{task}}$ from the vector represents the max class score drop of $unit_i$ from $\mathbf{h}^{\mathbf{pos}}$.

Applying this method, for POS, CK, NER and DP, leads to 4 *max-class-score-drop* vectors, one for each task, $\mathbf{v_{pos}}$, $\mathbf{v_{ck}}$, $\mathbf{v_{ner}}$ and $\mathbf{v_{dp}}$, that we plot in Fig.4 (one vector per line). We observe high values of *max class score drop* for POS compared to the remaining tasks. First, since $\mathbf{h}^{\mathbf{pos}}$'s units are more important for POS tagging than all other tasks. And, second, $\mathbf{h}^{\mathbf{pos}}$'s units are directly used for prediction for POS while transformed through

Task	Class	Unit	Top-10 activations
CK	B-INTJ	POS-Unit-112	:); awwwwwwww; uggghh; Omg; lol; hahahaha; WELL; Nope; LOL; No
	B-ADJP	POS-Unit-99	rapidely; Deeply; fine; more; hardly; particulary; slower; guilty; loose; entirely
DP	auxiliary	POS-Unit-47	do; can; was; ca; can; 's; would; have; ame; Wo
	discourse	POS-Unit-112	Hhhahahh; no; lmao; sorry; omg; hey; lol; yea; haha; please
NER	B-location	POS-Unit-35	North; Ireland; Italy; Kelly; Qatar; in; southafrica; new; over; Wellington
	B-person	POS-Unit-115	Trilarion; Jo; Watson; Hanzo; Abrikosov; Lily; jellombooty; theguest; Professor

Table 4: Top-10 words activating positively (red) or negatively (blue) (Since LSTMs generate positive and negative activations) some units from $\mathbf{h}^{\mathbf{pos}}$ that are the most important for different classes from CK, DP and NER

several layers for the other tasks. Furthermore, we can also observe that $\mathbf{h}^{\mathbf{pos}}$'s units are more important for CK and DP compared to NER.

Moreover, we attempt to peek inside some units from $\mathbf{h}^{\mathbf{pos}}$, the ablation thereof begets the higher drop in CK, DP and NER classes-scores. Specifically, we report in Tab.4 the top-10 words activating some of these units, as in (Kádár et al., 2017; Mefta h et al., 2019). Expectedly, we found that some of POS' units are firing, and thus specialised, on patterns that are beneficial for higher-level tasks. For instance, Unit-99, specialised on adjectives ending with the suffix "ly", is highly important for the CK class "B-ADJP" (*beginning of adjectival phrase*). Also, Unit-115, is firing on persons names, a valuable pattern for "B-person" class of NER. Interestingly, we found some units that are beneficial for multiple tasks, *e.g.* Unit-112, which is specific for interjections, is also important for both "discourse" class for DP and "B-INTJ" (*beginning of an interjection phrase*) for CK.

6 Conclusion

In this research, we have proposed **MuTSPad**, a new approach based on Transfer Learning (TL) for domain adaptation with three main contributions: 1) Consolidating two TL's approaches, sequential transfer learning and multi-task learning, by pretraining on resource-rich domain and fine-tuning on low-resourced domain in a multi-task fashion; 2) Unifying independent datasets to overcome the intricacy of multi-task training from different datasets; and 3) Conducting a set of individual units ablation, refining our understanding on how individual neurons lower-level tasks impact high-level tasks. We showed through empirical results on domain adaptation from *News* to *Tweets* that the proposed method *MuTSPad* allows a simultaneous benefit from similarities between domains and tasks, yielding better transfer learning performances on four NLP tasks, and outperforming the state-of-the-art on Dependency Parsing task.

This study leaves several important open directions for future work. First, we should explore soft multi-task architectures. Second, we expect to explore the combination of supervised and unsupervised multi-tasking. We also plan to explore the benefit of **MuTSPad**'s learned representations for higher-level NLP applications such as machine translation and sentiment analysis.

References

Wasi Uddin Ahmad, Xueying Bai, Zhechao Huang, Chao Jiang, Nanyun Peng, and Kai-Wei Chang. 2018. Multi-task learning for universal sentence representations: What syntactic and semantic information is captured? *arXiv preprint arXiv:1804.07911*.

Alan Akbik, Tanja Bergmann, and Roland Vollgraf. 2019. Pooled contextualized embeddings for named entity recognition.

R Caruana. 1997. Multitask learning [phd dissertation]. school of computer science.

Soravit Changpinyo, Hexiang Hu, and Fei Sha. 2018. Multi-task learning for sequence tagging: An empirical study. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2965–2977.

Xilun Chen, Yu Sun, Ben Athiwaratkun, Claire Cardie, and Kilian Weinberger. 2016. Adversarial deep averaging networks for cross-lingual sentiment classification. *arXiv preprint arXiv:1606.01614*.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, Anthony Bau, and James Glass. 2019. What is one grain of sand in the desert? analyzing individual neurons in deep nlp models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6309–6317.

Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. Results of the wnut2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147.

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.

Timothy Dozat and Christopher D Manning. 2016. Deep biaffine attention for neural dependency parsing. *arXiv preprint arXiv:1611.01734*.

Timothy Dozat, Peng Qi, and Christopher D Manning. 2017. Stanford's graph-based neural dependency parser at the conll 2017 shared task. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 20–30.

Long Duong. 2017. *Natural language processing for resource-poor languages*. Ph.D. thesis.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030.

Jing Gu and Zhou Yu. 2020. Data annealing for informal language understanding tasks. *arXiv preprint arXiv:2004.13833*.

Tao Gui, Qi Zhang, Haoran Huang, Minlong Peng, and Xuanjing Huang. 2017. Part-of-speech tagging for twitter with adversarial neural networks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2411–2420.

Kazuma Hashimoto, Yoshimasa Tsuruoka, Richard Socher, et al. 2017. A joint many-task model: Growing a neural network for multiple nlp tasks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1923–1933.

Sébastien Jean, Orhan Firat, and Melvin Johnson. 2018. Adaptive scheduling for multi-task learning.

Akos Kádár, Grzegorz Chrupała, and Afra Alishahi. 2017. Representation of linguistic form and function in recurrent neural networks. *Computational Linguistics*, 43(4):761–780.

Eliyahu Kiperwasser and Miguel Ballesteros. 2018. Scheduled multi-task learning: From syntax to translation. *Transactions of the Association of Computational Linguistics*, 6:225–240.

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*.

Ying Lin, Shengqi Yang, Veselin Stoyanov, and Heng Ji. 2018. A multi-lingual multi-task architecture for low-resource sequence labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 799–809, Melbourne, Australia. Association for Computational Linguistics.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*.

Yijia Liu, Yi Zhu, Wanxiang Che, Bing Qin, Nathan Schneider, and Noah A Smith. 2018. Parsing tweets into universal dependencies. In *NAACL*, volume 1, pages 965–975.

Peng Lu, Ting Bai, and Philippe Langlais. 2019. Sc-lstm: Learning task-specific representations in multi-task learning for sequence labeling. 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 2396–2406.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*.

Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank.

Sara Meftah and Nasredine Semmar. 2018. A neural network model for part-of-speech tagging of social media texts. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.

Sara Meftah, Nasredine Semmar, Fatiha Sadat, and Stephan Raaijmakers. 2018. Using neural transfer learning for morpho-syntactic tagging of south-slavic languages tweets. In *Proceedings of the Fifth Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial 2018)*, pages 235–243.

Sara Meftah, Nasredine Semmar, Othman Zennaki, and Fatiha Sadat. 2017. Using transfer learning in part-of-speech tagging of english tweets.

Sara Meftah, Youssef Tamaazousti, Nasredine Semmar, Hassane Essafi, and Fatiha Sadat. 2019. Joint learning of pre-trained and random units for domain adaptation in part-of-speech tagging. 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 4107–4112.

Shubhanshu Mishra. 2019. Multi-dataset-multi-task neural sequence tagging for information extraction from tweets. In *Proceedings of the 30th ACM Conference on Hypertext and Social Media*, pages 283–284. ACM.

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 1659–1666.

Joakim Nivre, Johan Hall, and Jens Nilsson. 2004. Memory-based dependency parsing. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, pages 49–56.

Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359.

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.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL*, volume 1, pages 2227–2237.

Peng Qi, Timothy Dozat, Yuhao Zhang, and Christopher D. Manning. 2018. Universal dependency parsing from scratch. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 160–170, Brussels, Belgium. Association for Computational Linguistics.

Alan Ritter, Sam Clark, Oren Etzioni, et al. 2011. Named entity recognition in tweets: an experimental study. In *EMNLP*, pages 1524–1534. Association for Computational Linguistics.

Sebastian Ruder. 2019. *Neural Transfer Learning for Natural Language Processing*. Ph.D. thesis, NATIONAL UNIVERSITY OF IRELAND, GALWAY.

Sebastian Ruder12, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. 2019. Latent multi-task architecture learning.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

Tjong Kim Sang, F Erik, and Sabine Buchholz. 2000. Introduction to the conll-2000 shared task: chunking. In *Proceedings of CoNLL-2000, Lisbon, Portugal*, pages 127–132.

Victor Sanh, Thomas Wolf, and Sebastian Ruder. 2019. A hierarchical multi-task approach for learning embeddings from semantic tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6949–6956.

Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 231–235.

Sandeep Subramanian, Adam Trischler, Yoshua Bengio, and Christopher J Pal. 2018. Learning general purpose distributed sentence representations via large scale multi-task learning. *ICLR*.

Youssef Tamaazousti, Hervé Le Borgne, Céline Hudelot, Mohamed El Amine Seddik, and Mohamed Tamaazousti. 2019. Learning more universal representations for transfer-learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Jie Yang, Shuailong Liang, and Yue Zhang. 2018. Design challenges and misconceptions in neural sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*.

Zhilin Yang, Ruslan Salakhutdinov, and William W Cohen. 2017. Transfer learning for sequence tagging with hierarchical recurrent networks. *arXiv preprint arXiv:1703.06345*.

Poorya Zaremoodi, Wray Buntine, and Gholamreza Haffari. 2018. Adaptive knowledge sharing in multi-task learning: Improving low-resource neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 656–661.

Othman Zennaki, Nasredine Semmar, and Laurent Besacier. 2016. Inducing multilingual text analysis tools using bidirectional recurrent neural networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 450–460.

Othman Zennaki, Nasredine Semmar, and Laurent Besacier. 2019. A neural approach for inducing multilingual resources and natural language processing tools for low-resource languages. *Natural Language Engineering*, 25(1):43–67.

Bolei Zhou, Yiyou Sun, David Bau, and Antonio Torralba. 2018. Revisiting the importance of individual units in cnns via ablation. *arXiv preprint arXiv:1806.02891*.

Author Index