nlpUP at SemEval-2020 Task 12 : A Blazing Fast System for Offensive Language Detection

Ehab Hamdy, Jelena Mitrović, Michael Granitzer
Faculty of Computer Science and Mathematics, University of Passau, Germany
hussei03@ads.uni-passau.de
{jelena.mitrovic|michael.granitzer}@uni-passau.de

Abstract

In this paper, we introduce our submission for the SemEval Task 12, sub-tasks A and B for offensive language identification and categorization in English tweets. This year the dataset for Task A is significantly larger than in the previous year. Therefore, we have adapted the BlazingText algorithm to extract embedding representation and classify texts after filtering and sanitizing the dataset according to the conventional text patterns on social media. We have gained both advantages of a speedy training process and obtained a good F1 score of 90.88% on the test set. For sub-task B, we opted to fine-tune a Bidirectional Encoder Representation from a Transformer (BERT) to accommodate the limited data for categorizing offensive tweets. We have achieved an F1 score of only 56.86%, but after experimenting with various label assignment thresholds in the pre-processing steps, the F1 score improved to 64%.

1 Introduction

In the era of social media and communications, it is easier than ever to freely express opinions on a plethora of topics. This openness creates a proliferation of useful information for productivity and making the right decisions. However, it also brings up opportunities for harsh discussions that can easily reach uncivilized, hateful, offensive or toxic levels (Shaw, 2011). Forms of offensive language are many-fold and there is no concensus in the terminology. In an effort to unify the terminology used by the NLP community working on the offensive language detection tasks (Caselli et al., 2020) proposed the AbuseEval annotation scheme to reconcile these term definitions based on the explicitness and target of the abusive language.

The NLP community has risen to the challenge of investigating the methods that reduce the phenomenon of offensive language. Due to the diversity of internet communication, this problem is addressed for languages other than English, such as Arabic (Mubarak et al., 2017), German (Wiegand et al., 2018), Italian (Bosco et al., 2018), and other languages. The most prominent efforts in this regard have been the HatEval Task 5 of Semeval-2019: Multilingual detection of hate speech against immigrants and women in twitter (Basile et al., 2019) and the OffensEval Task 6 of SemEval-2019:Identifying and Categorizing Offensive Language in Social Media (Zampieri et al., 2019b), which is seeing its second edition this year, as Task 12 of SemEval-2020: Multilingual Offensive Language Identification in Social Media (Zampieri et al., 2020).

The OffensEval 2019 challenge (Zampieri et al., 2019b) introduced labeled data sets containing offensive tweets. In its second edition it is again divided into three sub-tasks for granular identification of abusive language. The goal of the first task (Task A) is to differentiate between offensive and non-offensive language. The second (Task B) and third (Task B) tasks are focusing on a more fine-grained identification such as the type and the target of the offensive text.

For our submission in sub-task A this year, we have opted for the BlazingText (Gupta and Khare, 2017) which can leverage multiple GPUs for training which leads to a 9x speedup compared to CPU implementation with minimal effect on the quality of embedding and classification. For task B we have chosen a BERT fine- tuned model for our submission since it produces better results in our experiments.

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 2098–2104
Barcelona, Spain (Online), December 12, 2020.

The rest of this paper is structured as follows. We explore some background of the problem of offensive language detection from the prospective of NLP and previous work in the field. Then we discuss the OffensEval 2020 (Zampieri et al., 2020) challenge, including the dataset and task description. Afterwords, we present our system description and the obtained results. Finally, we conclude the paper and give suggestions for future work.

2 Background

Identifying offensive language on the internet has become one of the most popular applications of NLP (Schmidt and Wiegand, 2017). In (Baldwin et al., 2013) authors conducted a systematic study on texts of social media and microblogs. They proved that text on social media texts is linguistically noisy and less grammatical than edited text. Researchers have proposed a set of feature engineering methods with different levels of effectiveness. Surface features such as bag of words and n-grams are the simplest representative features of texts. A more advanced technique for extracting numerical representations for texts is word generalization which include methods such as word clustering (Warner and Hirschberg, 2012), distributed word representations using neural networks (word embedding), and document embedding (Djuric et al., 2015). Another prominent feature engineering technique is multimodal features, which incorporates additional features from images, audio and video. Authors in (Yang et al., 2019) used feature fusion of text and photos for detecting hate speech in Facebook posts.

On the other side, the advancements of language modeling and machine learning techniques show promising results when tackling the problem of offensive language identification. In the first edition of OffensEval (Zampieri et al., 2019b), transfer learning methods such as BERT (Devlin et al., 2019) proved to be among the most effective and accurate (as shown by the best system at OffensEval 2019 (Liu et al., 2019)) especially when dealing with limited labeled datasets. Still, some other models proved their predictive power as well, e.g., the C-BiGRU model which combines a Convolutional Neural Network (CNN) with a bidirectional Recurrent Neural Network (RNN) (Mitrović et al., 2019) that scored in the 9th position of OffensEval 2019.

As this year's OffensEval dataset for English is significantly larger than last year, we have to think about the time needed for building a model and generating inferences. Conventionally large pre-trained model might pose a difficulty on the system performance when it is deployed to generate inferences in the long run.

3 Dataset and Tasks Description

OffensEval 2020 provides a massive dataset compared to the OffensEval 2019 round of the challenge which only contains 13420 records. There are many opportunities now for the state of the art deep learning models which tend to perform better on large amounts of data. At the same time this generated a new challenge on how to process the data as efficiently and as fast as possible. It becomes a necessity to rely on a GPU powered machine to train complex neural network models with billions of words in the given text corpus. In the following sections of this paper, we will illustrate our methodology to tackle that challenge and harness the usefulness of data proliferation.

The training and development data set for OffensEval 2020 is composed of over 9 million records, each record containing a short text representing a tweet. Similarly to OffensEval 2019 dataset(Zampieri et al., 2019a), OffensEval 2020 (Rosenthal et al., 2020) is annotated with a hierarchical three-level annotation schema. The goal of the first annotation level is to discriminate between offensive and non-offensive tweets. The second annotation level is based on the of offense seen as a targeted insult (TIN) or un-targeted insult (UNT). Finally, the third annotation level is focusing on three target types including individual (IND), group (GRP) and other (OTH). A summary showing the statistics of the dataset is shown in Figure 1. As the dataset labels where given in terms of probabilities, in the next section we will discuss how the label assignment was performed.

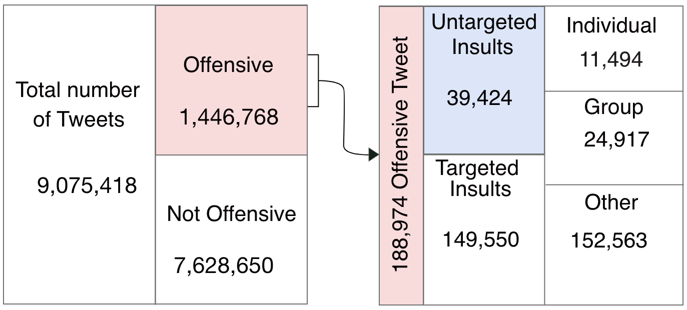

Figure 1: OLID Data and label organization based on an 0.5 threshold for label assignment

4 System Description

Our submissions target tasks A and B for the English language dataset (Rosenthal et al., 2020). We will start with discussing the pre-processing pipeline and then the classification methods we used.

4.1 Data pre-processing

Text extracted from social media contains a wide range of irregularities such as misspelling, duplication and incorrect punctuation, that is specifically true for texts with offensive language tunes. Therefore, we start with performing data cleansing and normalization by removing the special characters and applying lower case transformation to help reduce the dimensionality. We then replace emojis into their respective representative text to improve the context of our target text. Finally, we used the Wordsegment library to tokenize and extract individual words from hashtags (Djuric et al., 2015).

In contrast to the OffensEval 2019 dataset (Zampieri et al., 2019a), the labeled data are given as probability values. For sub-task A, a probability value closer to one indicates an offensive tweet, while a probability closer to zero indicates a non-offensive tweet. After investigating the dataset, we have decided to assign the label (OFF) for tweets that have a probability higher than 0.5 and (NOT) otherwise. Similarly, for the sub-task B, we have used the 0.5 threshold to assign whether the offensive tweet is untargeted (UNT) or targeted (TIN). Figure 1 shows a reasonable class distribution according to the aforementioned threshold.

Since the data set for task A is relatively large, we have used 99% of the development data as a training set and the remaining 1% are considered sufficient for validation and testing sets. While for task B, we have stick with the conventional data splitting ration of 60:20:20 for training, validation and test sets respectively.

4.2 Methodology

Baseline Model After cleaning the dataset as mentioned in the previous section, we have used the uni-gram, bi-gram and tri-gram TF-IDF as the input features and compared the performance of three classifiers, including Logistic Regression (LogR), Support Vector Machines (SVM) , and Decision Trees (DT). We have only considered the results of the classifiers trained on the bi-gram TF-IDF as it shows the best performance.

BlazingText Algorithm For sub-tasks A and B, we have used BlazingText for classification. BlazingText was introduced by researchers at Amazon (Gupta and Khare, 2017). They propose an efficient implementation of Word2Vec (Mikolov et al., 2013) and FastText classifier (Joulin et al., 2016) that can take advantage of multiple CPUs and GPUs in training the model and real-time inference. We have also turned on automatic hyperparameter tuning for the learning rate, vector dimension, minimum word count and word n-grams parameter of the BlazingText for classification model. The best performing hyper-parameters were as following; the selected mode is of type Supervised, a minimum word count of 5, learning rate set to 0.05, early stopping is activated with minimum number of 5 epochs, word ngrams is set to 2, and the dimension of the embedding layer is set to 10. The model training time for the whole dataset amounts to 40 seconds and generating inferences for the whole testing set took around 2 seconds.

BERT The Bidirectional Encoder Representations from Transformers, is a pretrained transformer model developed by the Google research team (Devlin et al., 2019) where it is originally used for understanding user searches. We have utilized the uncased-BERT-Base architecture which is composed of a network of 12 layers for Task B where we have a considerably smaller dataset. We have decided to limit the length of the input sentences to 128 characters where longer sentences are truncated while padding tokens are added for shorter sentences.

5 Results

In Table 1 we report our results for sub-task A. Starting with the results on the cross validation test, BlazingText appears to have a better F1 score of 94.1% compared to the 92.9 model of the baseline SVM model. Similarly, on the gold test set, the F1 score from the BlazingText model outperforms the base model.

Model	Accuracy		F1		Runtime
	CV	Gold	CV	Gold	
BlazingText	96.89%	92.1%	94.1%	90.88%	40 seconds
BaseMode	96.37%	92.3%	92.9%	90.2%	12 minutes
All NOT	83.87%	72.1%	45.9%	42%	NA
All OFF	16.1%	27.9%	13.8%	21.7%	NA

Table 1: Experimental results of sub-task A (CV = cross-validation; gold = gold test set) based on a 0.5 threshold for label assignment..

One important observation on the difference between the 2019 and 2020 OffensEval sub-task A results with the same methods is that, using an additional dataset for the considerably smaller dataset appeared to give a big performance benefit in terms of the accuracy as well as the F1 score, while that was not the case for the larger, 2020 dataset. Table 2 shows the test results of the gold test data for 2019 based on training data set of 2019 dataset only versus using additional data from the 2020 OLID data.

Model	Model results on 2019 Gold Testset	
	Training with OLID 2019	Training with OLID 2019 and 2020
BlazingText	76.5%	82.7%
BaseMode	70.9%	79.6%

Table 2: Test results comparison on 2019 data when training with additional data set versus only training with OLID 2019 dataset

For sub-task B, we have chosen the fine-tuned BERT model as it performs the best according to our experiment, however the training time was significantly longer than training a BlazingText model with a marginal performance improvements of around 0.3%. Our submission achieved 56.86% F1 score on the gold test set. After the competition ended we have experimented with various thresholds when assigning the labels for sub-task B. It was apparent that choosing a threshold of 0.4 rather than 0.5 significantly improved our results for both cross validation and gold test set by 9% where the BERT model was also best performing model with 64% F1 score as shown in table 3.

6 Conclusion and Future Work

In this paper we have presented our submission for the OffensEval 2020 competition for sub-tasks A and B. While being optimizing for accuracy, we have also considered the time required to train a model and generate inferences which is critical for taking proactive action to fight against hate speech on social media. To achieve that we have chosen a highly scalable and speedy implementation of Word2Vec and FastText called BlazingText. We have also compared its performance with the BERT model as one of the efficient transfer learning methods used for offensive language identification. BERT shows marginal accuracy

Model	Accuracy		F1		Runtime
	CV	Gold	CV	Gold	
BERT	75.6%	69.7%	67.4%	64%	2 hours
BaseModel	70.3%	69.5%	64.5%	63.2%	7 minutes
BlazingText	71%	68.34%	64.3%	62.9%	12 seconds

Table 3: Experimental results of sub-task B (CV = cross-validation; gold = gold test set) based on a 0.4 threshold for label assignment.

improvements which raises some questions regarding the cost and benefit of transfer learning models.One of our future goals is to study the trade-offs between accuracy of the system and its responsiveness. We also want to compare the more computationally efficient implementations of BERT such as TinyBERT (Jiao et al., 2019) and DistilBERT (Sanh et al., 2019) to find a balance between accuracy and speediness of an offensive language detection system. An important aspect of offensive language is its rhetorical impact. We want to study the effects of rhetorical features in offensive language as they are quite common and contribute to the implicit interpretation of such language. We will use the methods laid out in (Mitrovic et al., 2020) and (Mitrović et al., 2017), and combine them with the latest NN approaches to detecting offensive language.

6.1 Acknowledgement

SPONSORED BY THE

Federal Ministry
of Education
and Research

The project on which this report is based was funded by the German Federal Ministry of Education and Research (BMBF) under the funding code 01—S20049. The author is responsible for the content of this publication.

References

Timothy Baldwin, Paul Cook, Marco Lui, Andrew MacKinlay, and Li Wang. 2013. How noisy social media text, how diffrnt social media sources? In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 356–364.

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.

Cristina Bosco, Fabio Poletto Dell'Orletta, Felice, Manuela Sanuguinetti, and Maurizio Tesconi. 2018. Overview of the EVALITA Hate Speech Detection (HaSpeeDe) Task. In Tommaso Caselli, Nicole Novielli, Viviana Patti, and Paolo Rosso, editors, *Proceedings of the 6th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA'18)*, Turin, Italy. CEUR.org.

Tommaso Caselli, Valerio Basile, Jelena Mitrović, Inga Kartoziya, and Michael Granitzer. 2020. I Feel Offended, Don't Be Abusive! Implicit/Explicit Messages in Offensive and Abusive Language. In *Proceedings of the Twelfth International Conference on Language Resources and Evaluation (LREC 2020)*, Marseille, France. European Language Resources Association (ELRA).

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT 2019: Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186.

Nemanja Djuric, Jing Zhou, Robin Morris, Mihajlo Grbovic, Vladan Radosavljevic, and Narayan Bhamidipati. 2015. Hate speech detection with comment embeddings. In *Proceedings of the 24th international conference on world wide web*, pages 29–30.

Saurabh Gupta and Vineet Khare. 2017. Blazingtext: Scaling and accelerating word2vec using multiple gpus. In *Proceedings of the Machine Learning on HPC Environments*, page 6.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2019. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Ping Liu, Wen Li, and Liang Zou. 2019. NULI at SemEval-2019 task 6: Transfer learning for offensive language detection using bidirectional transformers. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 87–91, Minneapolis, Minnesota, USA, June. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Jelena Mitrović, Cliff O'Reilly, Miljana Mladenović, and Siegfried Handschuh. 2017. Ontological representations of rhetorical figures for argument mining. *Argument & Computation*, 8:267–287.

Jelena Mitrović, Bastian Birkeneder, and Michael Granitzer. 2019. nlpup at semeval-2019 task 6: a deep neural language model for offensive language detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 722–726.

Jelena Mitrovic, Cliff O'Reilly, Randy Allen Harris, and Michael Granitzer. 2020. Cognitive modeling in computational rhetoric: Litotes, containment and the unexcluded middle. In Ana Paula Rocha, Luc Steels, and H. Jaap van den Herik, editors, *Proceedings of the 12th International Conference on Agents and Artificial Intelligence, ICAART 2020, Volume 2, Valletta, Malta, February 22-24, 2020*, pages 806–813. SCITEPRESS.

Hamdy Mubarak, Kareem Darwish, and Walid Magdy. 2017. Abusive language detection on arabic social media. In *Proceedings of the First Workshop on Abusive Language Online*, pages 52–56.

Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Marcos Zampieri, and Preslav Nakov. 2020. A Large-Scale Semi-Supervised Dataset for Offensive Language Identification. In *arxiv*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

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.

LaShel Shaw. 2011. Hate speech in cyberspace: bitterness without boundaries. *Notre Dame JL Ethics & Pub. Pol'y*, 25:279.

William Warner and Julia Hirschberg. 2012. Detecting hate speech on the world wide web. In *Proceedings of the second workshop on language in social media*, pages 19–26. Association for Computational Linguistics.

Michael Wiegand, Melanie Siegel, and Josef Ruppenhofer. 2018. Overview of the GermEval 2018 Shared Task on the Identification of Offensive Language. In *Proceedings of GermEval*.

Fan Yang, Xiaochang Peng, Gargi Ghosh, Reshef Shilon, Hao Ma, Eider Moore, and Goran Predovic. 2019. Exploring deep multimodal fusion of text and photo for hate speech classification. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 11–18.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019a. Predicting the Type and Target of Offensive Posts in Social Media. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 1415–1420.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019b. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). *arXiv preprint arXiv:1903.08983*.

Marcos Zampieri, Preslav Nakov, Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Hamdy Mubarak, Leon Derczynski, Zeses Pitenis, and Çağrı Çöltekin. 2020. SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020). In *Proceedings of SemEval*.