AlexU-BackTranslation-TL at SemEval-2020 Task 12: Improving Offensive Language Detection using Data Augmentation and Transfer Learning

Mai Ibrahim, Marwan Torki and Nagwa El-Makky
Computer and Systems Engineering Department
Alexandria University
Alexandria, Egypt
eng-mai.ibrahim, mtorki, nagwa.elmakky@alexu.edu.eg

Abstract

Social media platforms, online news commenting spaces, and many other public forums have become widely known for issues of abusive behavior such as cyber-bullying and personal attacks. In this paper, we use the annotated tweets of Offensive Language Identification Dataset (OLID) to train three levels of deep learning classifiers to solve the three sub-tasks associated with the dataset. Sub-task A is to determine if the tweet is toxic or not. Then, for offensive tweets, sub-task B requires determining whether the toxicity is targeted. Finally, for sub-task C, we predict the target of the offense; i.e. a group, individual or other entity. In our solution, we tackle the problem of class imbalance in the dataset by using back translation for data augmentation and utilizing fine-tuned BERT model in an ensemble of deep learning classifiers. We used this solution to participate in the three English sub-tasks of SemEval-2020 task 12. The proposed solution achieved 0.91393, 0.6300 and 0.57607 macro F1-average in sub-tasks A, B and C respectively. We achieved the 8th, 14th and 21st places for sub-tasks A, B and C respectively.

1 Introduction

Nowadays, public online communities such as blogs, forums and social networks have become an integral part of many people's lives. These platforms enable their users to express their opinions and discuss things they care about as well as immediately react to and comment on other users posts and stories. Knowing the importance of online commenting, many online news providers nowadays have also established commenting services that enable their users to exchange their thoughts and opinions regarding the published news (Cho and Acquisti, 2013). Despite all its benefits, unfortunately without proper moderation, these platforms can easily be used for online abuse and harassment that can have serious effects on its victims. Therefore offensive content detection has become a great concern for online communities and social media platforms.

The online offensive content can vary in different aspects, such as the toxicity type, the target and whether the abuse is implicit or explicit. SemEval-2020 task 12 (Zampieri et al., 2020) addresses the problem of offensive language detection in social media at three levels that map to three sub-tasks. The task dataset consists of tweets labeled for offensive content using three-level hierarchical annotation scheme. Each tweet in the dataset has three labels. The first label shows if the tweet is offensive or not (sub-task A). For offensive tweets, the second label divides them to targeted and untargeted (just profane) ones (sub-task B). Finally, the third label categorizes the targeted offensive tweets based on their target that can be an individual, a group of people or other entity or organization (sub-task C).

The main challenge in this task is that the published task training dataset (Rosenthal et al., 2020) is not manually labeled by human annotators. Instead, the dataset is semi-supervised where the samples are labeled using a number of models trained on Offensive Language Identification Dataset (OLID) (Zampieri et al., 2019a). For each record of the training data, the average score (from different models) and standard deviation of these scores are provided for the classes in each sub-task instead of the hard label. Therefore, we preferred to train our models on the original OLID data which was the official dataset

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1881–1890
Barcelona, Spain (Online), December 12, 2020.

for SemEval-2019 task 6 (OffensEval 2019) (Basile et al., 2019). We also evaluated our models on the new training dataset and the results showed that the existing models achieve significantly high macro average F1-scores on the new training dataset.

Moreover, OLID dataset suffers from severe class distribution imbalance problem in its records. For example, 67% of the data is not offensive at all while some classes such as untargeted offensive tweets are represented with less than 4% of the records. Data augmentation is a powerful tool to increase the size of the dataset and solve the class imbalance problem by generating new samples of the minority classes. In this paper, we propose using back translation augmentation method to create multiple new versions of existing records by simply translating them from English to some other language and then back to English. The back translated tweet will not be exactly the same as the original one, however, it will still have the same meaning and therefore we can assign the same class label to it. We use Google Translate API for this purpose and the evaluation results show the effect of data augmentation in significantly improving the models performance.

Another challenge we faced when working on OLID dataset is that it is a comparatively small dataset ($14,100$ records) which makes it hard to train complex models. Transfer learning is usually used in these cases where a model trained for some task is reused as a starting point for a model in a second task. In natual language processing, Word embeddings such as word2vec (Mikolov et al., 2013), FastText (Joulin et al., 2016) and Glove (Pennington et al., 2014) are good realizations of transfer learning where they are used to convert the input text (of some target task) to low-dimensional vectors learnt from large-scale dataset. Recently, Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018) became the most popular natural language processing (NLP) approach to transfer learning. BERT which is pretrained by Google AI Language team was successfully fine-tuned for a wide range of tasks, such as question answering and language inference where it achieved state-of-the-art performance. Therefore, we utilize fine-tuned BERT as an integral part of our solution.

This paper is organized as follows. In Section 2, we discuss relevant related works in offensive language detection. We describe our proposed system in Section 3. The data preprocessing and augmentation techniques in addition to the models implementation details are explained in Section 4. Then we report and analyze the evaluation results in Section 5. Finally, we provide our conclusions and future work in Section 6.

2 Related Work

After proving their effectiveness in different fields, deep learning techniques became very common in solving NLP problems. One of these problems is text classification that span a wide range of application domains including business, medicine, law and society (MALI and Atique, 2014). In the last few years, offensive content prediction in social media gained a great attention as a text classification task that can help reduce online harassment and abuse.

Recently, many different offensive language datasets were published that allowed several studies to work on online toxic content recognition problem. These studied worked on data collected from famous social media spaces such as Twitter (Xu et al., 2012; Burnap and Williams, 2015; Davidson et al., 2017; Badjatiya et al., 2017), Facebook (Kumar et al., 2018) and Wikipedia comments available on Kaggle competition [1]. These researches addressed different aspects of offensive language such as detecting hate speech (Malmasi and Zampieri, 2017) and recognizing the multiple types of toxicity in a comment (Ibrahim et al., 2018). However, OffensEval 2020 adopts a new hierarchical labeling scheme to identify offensive tweets and their targets. This scheme is the same as that used in OLID dataset to determine the target of the offensive tweets (Zampieri et al., 2019a).

One of the main challenges of OLID dataset is that it is considered relatively small compared to other offensive language datasets with only $14,100$ records. The problem of small datasets is quite common and transfer learning was able to overcome it in many fields. The main idea of transfer learning is to use a pretrained model, trained on larger dataset of some task, as a starting point for training another model for a similar task to speed up the training and improve the performance. This approach is commonly used in

[1] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge

computer vision where pretrained models such as VGG (Simonyan and Zisserman, 2014) and Inception (Szegedy et al., 2016) are reused to initialize new training processes. In NLP, great advances have been achieved in word embeddings to provide accurate representations of the language words that can improve the performance of the models trained on them. These embeddings are usually obtained from training large models on large corpus and then reuse them in other tasks with smaller datasets which makes it a valid form of transfer learning. These representations include word2vec (Mikolov et al., 2013), FastText (Joulin et al., 2016) and Glove (Pennington et al., 2014) which significantly boosted the performance of the models on various NLP tasks. Recently, Google AI Language team proposed the BERT model that exploits transformer architecture to provide word representations that are dynamically informed by the words around them (Devlin et al., 2018). Pretrained BERT models achieved state-of-the-art results in multiple NLP tasks with minimal fine-tuning (up to 4 epochs) on the task data. Moreover, Bert evaluation results show its significant improvement specially on small datasets (1000s of records) which encouraged us to attempt to tune a Bert model for each of the three subtasks we have.

Another common problem with offensive language datasets is the imbalanced class distribution since it is usually hard to collect equal number of records for each class in the data. Again this problem was addressed in other fields such as computer vision where the images of the minority classes are augmented by cropping, rotating, or flipping to create new samples of these classes (Shorten and Khoshgoftaar, 2019). For toxicity types classification, data augmentation significantly improved the classifier performance on minority classes as shown in (Ibrahim et al., 2018) that worked on Wikipedia toxic comments dataset. The authors used different augmentation methods such as randomly removing or replacing words from the original comment with its synonyms. In this paper, we use back translation based augmentation, proposed in (Sennrich et al., 2015), since it is more guaranteed to generate records of the same class as the original one. That is because the sources of offensiveness in the text can be lost by removing random words or replacing them with their synonyms.

OLID is the official dataset of SemEval 2019 Task 6 for identifying and categorizing offensive language in social media (Zampieri et al., 2019b). The participating teams used different approaches in their solutions. However, many teams used ensembles of deep learning models (Mahata et al., 2019) to benefit from its minimal need for features engineering and ability to boost the classifier performance. Moreover, to address the small dataset problem some teams used Bert model (Liu et al., 2019) and others utilized external datasets to further increase the training data (Seganti et al., 2019). In this paper, we show that using back translation data augmentation and transfer learning significantly improves the offensive language prediction performance.

3 Proposed System

SemEval 2020 task 12 focuses on detecting offensive tweets and recognizing the target of those tweets. The task supports different languages and we tackle the problem on English language. Similar to OffensEval 2019, this task has three sub-tasks each one corresponds to a level in the three-level hierarchical labeling scheme of OLID dataset.

- **Sub-task A** categorizes the tweet for being offensive (OFF) or not (NOT).
- **Sub-task B** considers only offensive tweets and labels them based on whether they contain target offense (TIN) or untargeted (UNT) profanity.
- **Sub-task C** classifies targeted offensive tweets into three classes based on their target being an individual (IND), a group of people (GRP) or another entity or organization (OTH).

Figure 1 illustrates the main components of the proposed classification system. The input tweet first goes through preprocessing and cleaning step. After preprocessing, the tweet is converted to a fixed length sequence of words by padding shorter tweets and truncating longer ones. Then each word is replaced by its representation vector obtained from the pretrained word embeddings model.

For each of the three sub-tasks, three different deep learning networks were built: convolutional neural network (CNN), bidirectional long-short term memory (Bi-LSTM) and bidirectional gated recurrent unit

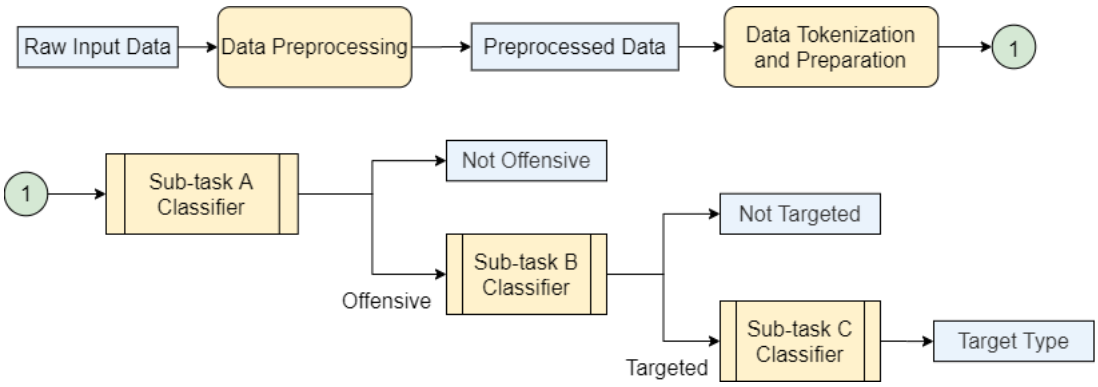


Figure 1: System Architecture and Main Components. The classifiers blocks represent an ensemble of deep learning models.

(Bi-GRU). Additionally, a separate BERT-base model is fine-tuned for each one of them. We perform hyper parameters tuning to find the best parameters configuration for each of these models. After that, to further improve the performance, the best models are combined in ensembles. The classifiers in figure 1 represent these models ensembles.

4 Implementation

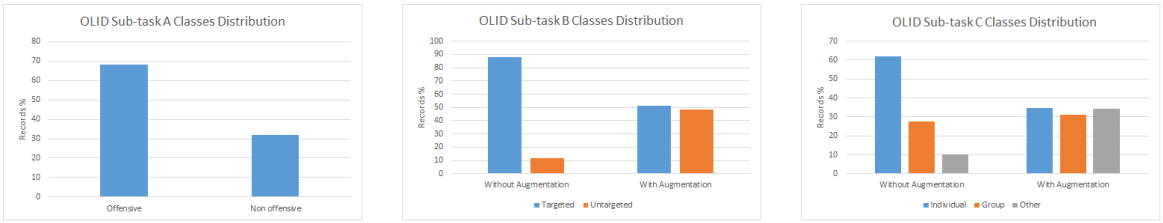
We trained the proposed solution models using OLID dataset. We divided its training data into 20% for validation and 80% for training. In this section, we explain the preprocessing methods and the back translation data augmentation technique. Additionally, we discuss the details of the different deep learning models we built or fine-tuned.

4.1 Data Preprocessing and Preparation

In online user generated content, spelling and grammar mistakes are quite common and some of them are even intentional. Leet speak is becoming more and more common amongst online users in which they replace standard letters by numerals or special characters that resemble the letters in appearance. Abusive users usually replace some letters in offensive words with special characters to fool the blacklists based detectors that may stop their posts from being published or get their accounts closed. Moreover, emojis, hashtags and links are usually found in users tweets. We apply a series of preprocessing steps to make the input tweet ready to be passed to the different deep learning models.

- **Normalizing Words and Letters** A dictionary was built to map common spelling variants of a large number of offensive words to their canonical form. A list of offensive words usually used in social media platforms [2] is used to build this map. For each of these words, we prepared a set of its common spellings.
- **Emojis Substitution** In some cases, the emojis (specially the facial expressions) can greatly change the meaning of the sentence and show the real intention of the writer. For this reason, we replace the emojis with their description phrase.
- **HashTag Segmentation** to split the hashtag to a phrase of separate words to further enrich the input sentence with more words.
- **Users Mentions** in the OLID dataset are replaced with a placeholder "@USER". In order to benefit from these mentions while reducing any possible redundancy, we limit the number of consecutive mentions to only one.
- **Links** in OLID dataset are also represented by a placeholder "URL". We replace "URL" with "HTTP" because it has a word embedding in pretrained models.

[2] https://github.com/RobertJGabriel/Google-profanity-words



(a) Sub-task A (b) Sub-task B (c) Sub-task C

Figure 2: Classes Distribution in the Three Levels (Sub-tasks) of OLID Dataset.

After preprocessing, we fix the tweets length to 64 words by truncating longer tweets and padding shorter ones. This length is selected as the mean value of the OLID training tweets lengths. After that each word in the input tweet is replaced with its vector representation obtained from a pretrained word embeddings model. We are using pretrained FastText to initialize the words embeddings but these representations are then updated during the training of the deep learning models. For unknown words that do not have a representation in the pretrained FastText model, we sample their representation vectors from a normal distribution with its mean and standard deviation set to the mean and standard deviation of the existing FastText embeddings.

4.2 Back Translation Data Augmentation

Like most of the offensive language datasets, OLID dataset suffers from imbalanced class distribution in its records. This problem, if not tackled properly, can make it very hard for the deep leaning models to learn the discriminating features of the minority classes which are represented with only a small number of samples in the training set. Figure 2 demonstrates the classes distributions in each of the training data available in each of the three levels (sub-tasks) of OLID.

In order to tackle this problem, data augmentation is usually used to create new samples and increase the diversity of data available for training models, without actually collecting new data. It is conducted by applying transformations on the existing samples in order to generate new ones under the condition that these transformations do not change the sample class. Based on the classes distribution, we only augmented the samples of ”untargeted” class for sub-task B and the ”group” and ”other” classes for sub-task C.

We applied back translation augmentation method to balance the classes distribution. In this method, the existing English tweet is translated to another language and then translated back to English. This technique was applied using a number of languages to get multiple copies of the same tweet. This method is very effective in creating new tweets that have the same meaning of the original ones but probably rewritten with different words or even a different structure. The quality of the translated text and hence the advantage of this augmentation method highly depends on the performance of the used translator. For this reason, we used Google translate API due to its impressive ability to dynamically translate text between numerous language pairs. However, this performance is not consistent over all the languages and the API might face some difficulties while translating to certain languages. Therefore, to choose the best languages to use, we examined the back translated text from a number of languages. This step is very important because mistranslated records can mislead the models during training and negatively affect their ability to learn.

Table 1 shows an example tweet and how its back translated versions look like using a number of different languages. We can see that the words are usually replaced with their synonyms when they are translated back to English. But in some cases, the new word does not contain the profanity in the original one and hence the class of the new sentence becomes different from the original class. This can be seen in the Arabic back translation where the word ”fucking” in the original sentence was replaced by ”dreaded” and the sentence became no longer offensive.

Furthermore, we studied the effect of adding back translated data on the models prediction performance.

Original Text	this gave me fucking heart palpitations I'm shaking as I type this
From Spanish	that gave me fucking heart palpitations that I'm shaking as I write this
From German	that gave me fucking palpitations I tremble as I type this
From Polish	it gave me a fucking palpitations. I shiver writing this
From Portuguese	it gave me fucked up palpitations, I'm shaking as I type this
From Arabic	That gave me the dreaded heartbeat that I was shaking with this writing
From Italian	this gave me fucking palpitations i'm shaking as i write this
From French	that gave me fucking heart palpitations that i'm shaking by typing this

Table 1: Sample tweet and its copies after back translation from different languages.

This experiment is conducted on sub-task B data where only the untargeted tweets are augmented. We used seven different languages for back translation augmentation. The augmented data is used to fine-tune a BERT-base model using the same hyper parameters in each experiment. Table 2 summarizes the new classes distribution and the macro average F1-score of the different models evaluated on the validation set.

The results show that augmenting the data using back translation caused a significant improvement in the models performance compared to using the data without augmentation. However the amount of improvement clearly depends on the language used. For example, back translation from Arabic downgraded the performance of the classifier. The reason behind this can be that the quality of translation between Arabic and English is not good enough in many machine translators including Google Translate API. Portuguese introduced the highest improvement in the model F1-score. Furthermore, using the best six languages resulted in a more balanced classes distribution and achieved the highest improvement in the results. In sub-task B, for "untargeted" class samples, we used the best six languages to generate new six copies of each record. While for sub-task C, we used the best six languages to augment the "other" class and only the best three languages to augment the "group" class. The new classes distribution for the two sub-tasks is shown in figure 2.

4.3 Models Implementation

We built an ensemble of deep learning classifiers for each one of the three sub-tasks. A crucial part of this ensemble is a fine tuned BERT base model. We tuned a Bert base model for each one of the sub-tasks and combined it with other deep learning models that we built from scratch in an ensemble. These different models predictions are combined using weighted soft voting. Each model prediction is weighted by its individual score on the validation set compared to the other models in the ensemble.

For the deep learning classifiers, we built two variations of CNNs. The first uses a single kernel that goes through the input words representations matrix. While in the second variant, we used two kernels with different sizes to scan the input matrix simultaneously. The features extracted by these kernels are then concatenated in a single vector and passed to the fully connected layers. Furthermore, we made use of recurrent neural network models in our solution by building bidirectional LSTM and bidirectional GRU models for each one of the three sub-tasks. We applied hyper-parameters tuning and selected the best network architectures and parameters configuration for each sub-task based on the models performance on the validation set.

5 Evaluation Results

In this section we report and discuss the results of the proposed solution when evaluated on OLID testing data. Moreover, we show the results of our submission to SemEval 2020 task 12 on English data.

For each sub-task, after selecting the best hyper parameters configuration for each model, we combine these models predictions in an ensemble using weighted soft voting. When evaluated on the testing set of the OLID dataset, our ensembles outperformed the winning models in the three sub-tasks of SemEval 2019 task 6 (OffensEval 2019) which used OLID as its official dataset. The results of the individual models and their ensembles on each sub-task are summarized in table 3.

	Class Distribution [Targeted Untargeted]	Macro average F1-score
Without Augmentation	[88% 12%]	0.5747
Portuguese (PT)	[79% 21%]	0.6070
German (DE)	[79% 21%]	0.6003
Spanish (ES)	[79% 21%]	0.5894
Italian (IT)	[79% 21%]	0.5886
French (FR)	[79% 21%]	0.5833
Polish (PL)	[79% 21%]	0.5727
Arabic (AR)	[79% 21%]	0.5391
PT + DE + ES	[65% 35%]	0.6187
Top 6 Languages (all except AR)	[52% 48%]	**0.6259**
All 7 Languages	[48% 52%]	0.5981

Table 2: Results of Back Translation Augmentation on OLID Sub-task B Validation Set.

The evaluation results on the testing set show that the fine-tuned BERT-Base model outperforms all the other deep learning models. However, combining its predictions with those obtained from the other models clearly improved the overall performance. This also shows that using ensemble of different models was beneficial and achieved its objective to boost the results and provide more accurate predictions than its models when used individually.

When used separately, CNN models give higher (or very close) F1-score compared to Bi-LSTMs and Bi-GRUs on testing set. This is quite expected since CNNs are usually preferred in text classification tasks where recurrent networks like LSTM and GRU are commonly used in sequence-to-sequence problems.

Additionally the different CNN variations applied affect the performance of the classifiers on all sub-tasks. In all sub-tasks, using two kernels simultaneously achieves higher macro average F1-score compared to using a single kernel. This is because using multiple kernels help capturing different information from different windows of words which increases the model ability to extract useful knowledge from the input text to improve its classification performance.

Finally for the recurrent neural networks, the evaluation results show that Bi-LSTMs always provide better performance compared to Bi-GRUs. This shows that LSTMs can learn better even with small dataset size like that of OLID. Generally in many tasks, both architectures yield comparable performance and tuning hyper parameters is more important than picking the ideal architecture.

5.1 Results on SemEval 2020 Task 12

We participated in SemEVal 2020 task 12 for multilingual offensive language identification in social media. We submitted solutions for the three sub-tasks on English tweets. A similar task was published in SemEval 2019 but it was only focused on English tweets and used OLID as its official dataset. For the new task a new dataset is provided for the participants. The dataset is much larger than OLID with about 9 million records for sub-task A, 188,000 records for sub-task B and the same number for sub-task C.

The main difference between this new dataset and the original OLID data is that the new data is not labeled by human annotators. Instead, the training data samples are labeled using a number of models trained on the old OLID dataset. For each record of the training data, the average score (from different models) and standard deviation of these scores are provided for the positive class instead of the hard label. This applies in sub-tasks A and B whereas in sub-task C, the average confidence and its standard deviation was given for each of the three classes.

After training our models on OLID data, we evaluated the ensembles performance on the new training dataset records. The purpose of this evaluation was to check whether we need to train our models on the new data or they perform well on it and there is no need for retraining. For this evaluation, we selected a set of the most reliable records for each sub-task. The records are selected based on their class average score and standard deviation (std) so that if std is added or removed from their scores the predicted class will

	Sub-task A	Sub-task B	Sub-task C
1. CNN (1 Kernel)	0.804	0.723	0.626
2. CNN (2 Kernels)	0.805	0.734	0.671
3. Bidirectional LSTM	0.816	0.730	0.661
4. Bidirectional GRU	0.814	0.716	0.618
5. BERT-base	0.829	0.784	0.670
Ensemble Models	2 + 3 + 5	3 + 5	2 + 3 + 5
Soft Voting	0.831	0.778	0.679
Weighted Soft Voting	**0.834**	**0.799**	**0.696**
Best Score in OffensEval 2019	0.829	0.755	0.660

Table 3: Comparison of Models Macro average F1-score on OLID Testing Data.

	Training Set	Testing Set
Sub-task A	0.9994	0.9139
Sub-task B	0.9289	0.6300
Sub-task C	0.8372	0.5760

Table 4: The Macro average F1-score of the proposed models ensemble on the data of SemEval 2020 Task 12.

not change. After that, our models ensembles are evaluated on these subsets and the results are reported in table 4. The results show that the existing models achieve significantly high macro average F1-scores on the new training dataset. This is quite expected since the new data is labeled in an unsupervised manner using models trained on OLID dataset. And since the system is also trained on OLID, it is likely that it gives similar predictions to the models used to score the new dataset.

Therefore, we used the best ensembles obtained earlier from training on OLID data in our submissions to SemEval 2020 task 12. The results of these models on the competition testing set are summarized in table 4. Our proposed solution was ranked 8th, 14th and 21st in sub-tasks A, B and C respectively.

6 Conclusion

In this paper, we propose a scheme for improving offensive language identification in social media. The proposed solution depends on using back translation for data augmentation to solve the problem of class imbalance usually found in the offensive language datasets. We built ensemble of deep learning models CNN, bidirectional LSTM and bidirectional GRU in addition to fine-tuned Bert base model. These ensembles were trained and evaluated on OLID dataset. The evaluation results showed that we outperformed the best results on OLID dataset reported in OffensEval 2019. Additionally, we used these models in our submissions to SemEval 2020 task 12. We participated in the three sub-tasks for offensive language detection in English tweets. Our proposed approach to be ranked 8th, 14th and 21st in sub-tasks A, B and C respectively.

For future work, we consider applying the proposed solution on datasets from other languages. Additionally, we can enhance the deep learning models by adding attention layers to the LSTM models or exploring the impact of building deeper neural networks with multiple convolutions or recurrent layers applied sequentially on the input text.

References

Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web*

Companion, pages 759–760.

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.

Pete Burnap and Matthew L Williams. 2015. Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. *Policy & Internet*, 7(2):223–242.

Daegon Cho and Alessandro Acquisti. 2013. The more social cues, the less trolling? an empirical study of online commenting behavior. In *Proc. WEIS*.

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 Ibrahim, Marwan Torki, and Nagwa El-Makky. 2018. Imbalanced toxic comments classification using data augmentation and deep learning. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 875–878. IEEE.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Ritesh Kumar, Atul Kr Ojha, Shervin Malmasi, and Marcos Zampieri. 2018. Benchmarking aggression identification in social media. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, pages 1–11.

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.

Debanjan Mahata, Haimin Zhang, Karan Uppal, Yaman Kumar, Rajiv Shah, Simra Shahid, Laiba Mehnaz, and Sarthak Anand. 2019. Midas at semeval-2019 task 6: Identifying offensive posts and targeted offense from twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 683–690.

Manisha MALI and Mohammad Atique. 2014. Applications of text classification using text mining. *International Journal of Engineering Trends and Technology*, 13(5):209.

Shervin Malmasi and Marcos Zampieri. 2017. Detecting hate speech in social media. *arXiv preprint arXiv:1712.06427*.

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.

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

Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Marcos Zampieri, and Preslav Nakov. 2020. A large-scale semi-supervised dataset for offensive language identification. *arXiv preprint arXiv:2004.14454*.

Alessandro Seganti, Helena Sobol, Iryna Orlova, Hannam Kim, Jakub Staniszewski, Tymoteusz Krumholc, and Krystian Koziel. 2019. Nlpr@ srpol at semeval-2019 task 6 and task 5: Linguistically enhanced deep learning offensive sentence classifier. *arXiv preprint arXiv:1904.05152*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Connor Shorten and Taghi M Khoshgoftaar. 2019. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60.

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Jun-Ming Xu, Kwang-Sung Jun, Xiaojin Zhu, and Amy Bellmore. 2012. Learning from bullying traces in social media. In *Proceedings of the 2012 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pages 656–666. Association for Computational Linguistics.

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. *arXiv preprint arXiv:1902.09666*.

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*.