2018 17th IEEE International Conference on Machine Learning and Applications

Imbalanced Toxic Comments Classification using Data Augmentation and Deep Learning

Mai Ibrahim, Marwan Torki and Nagwa El-Makky
Computer and Systems Engineering Department
Alexandria University
Alexandria, Egypt
Email: {eng-mai.ibrahim, mtorki, nagwamakky}@alexu.edu.eg

***Abstract*—Recently cyber-bullying and online harassment have become two of the most serious issues in many public online communities. In this paper, we use data from Wikipedia talk page edits to train multi-label classifier that detects different types of toxicity in online user-generated content. We present different data augmentation techniques to overcome the data imbalance problem in the Wikipedia dataset. The proposed solution is an ensemble of three models: convolutional neural network (CNN), bidirectional long short-term memory (LSTM) and bidirectional gated recurrent units (GRU). We divide the classification problem into two steps, first we determine whether or not the input is toxic then we find the types of toxicity present in the toxic content. The evaluation results show that the proposed ensemble approach provides the highest accuracy among all considered algorithms. It achieves 0.828 F_1-score for toxic/non-toxic classification and 0.872 for toxicity types prediction.**

***Index Terms*—natural language processing, sentence classification, multi-label classification, deep learning, convolutional neural network, long short-term memory, gated recurrent units**

I. Introduction

In recent years, social media platforms and online communities have become very pervasive and important for content sharing and social interaction. Such platforms provide a great environment for their users to express their thoughts and ideas. Unfortunately, cyber-bulling and harassment have become serious issues affecting a wide range of users causing different psychological problems like depression and even suicidality [1]. The abusive online content can fall in multiple categories of toxicity such as identity-based hate, threat, insult or obscene. Furthermore, a single comment can simultaneously contain different types of toxicity. Therefore, we use the multi-label Wikipedia's talk page edits dataset provided by Kaggle competition [1] to build a multi-label model that can detect the different types of toxicity existing in a comment.

Although it has a large number of comments, the Wikipedia dataset suffers from class imbalance. More than 85% of the records are not toxic at all and the toxicity classes have highly skewed distribution where three of the six toxicity classes are represented by less than 7% of the samples. In this paper, we tackle this problem using data augmentation techniques to generate new comments for the minority classes. This is a main contribution of this work over previous work on the same dataset which ignored this problem by using a balanced subset of the data and neglecting the rest of it [2].

The proposed approach has two steps of prediction. In the first step, a classifier determines if the comment is normal or contains toxicity of any type. Then, for comments marked as toxic, another classifier detects the types of toxicity present in the comment. We trained different deep learning models: convolutional neural network (CNN), bidirectional long-short term memory (LSTM) and bidirectional gated recurrent unit (GRU) for both classifiers. Then, we compared the performance of these models against an ensemble model that utilizes all of them.

This paper is organized as follows. In Section II, we discuss relevant related works in sentence classification as well as toxic comments classification. We explain the data augmentation techniques and models implementation details in Section III. Then we describe the performance metrics and discuss the evaluation results in Section IV. Finally, we provide our conclusions in Section V.

II. Related Work

Text classification is one of the most important problems in natural language processing. It has been widely studied in multiple research projects that achieved good classification accuracy with the aid of different machine learning methods. Recently, deep learning models have been increasingly used in text classification due to their high performance and minimal need for features engineering.

Convolutional neural networks (CNNs) have been widely applied for classification problems in different fields including text classification. In [3], Kim used CNNs to address a series of sentence-level classification tasks. The paper shows that CNNs with little hyperparameter tuning can achieve high classification accuracy on different benchmarks. However, all the tasks addressed were multi-class tasks where the input sentence can belong to only one of the available categories. In our work, we use CNNs in a multi-label problem since the input comment can belong to multiple toxicity types at the same time.

Although LSTMs are commonly used in sequence-to-sequence applications, Wang et al. managed to introduce LSTM-based solution for tweets sentiment prediction in [4]. The experiments showed that the proposed model outperforms

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

978-1-5386-6805-4/18/$31.00 ©2018 IEEE
DOI 10.1109/ICMLA.2018.00141

several feature-engineering approaches. Although this is also a multi-class problem, it encouraged us to experiment RNN models on our classification problem.

Recently Georgakopoulos et al. tackled the problem of toxic comments classification using the same Wikipedia dataset we use [2]. However, they proposed a CNN-based model that only predicts whether or not a comment is toxic. They do not address the problem of finding the types of toxicity present in the comment. Their solution also avoids the problem of data classes imbalance by using a balanced subset of the data for building the model. On the other hand, our proposed solution predicts the different types of toxicity in the input comment and we use data augmentation to overcome the imbalanced classes distribution in the training data.

III. Implementation

In the proposed classification system, we have two classifiers. A binary toxic/nontoxic classifier that predicts if the input comment is toxic or not and a multi-label classifier that detects the different types of toxicity in toxic comments. For each of the two classifiers, we built three different deep learning models: CNN, bidirectional LSTM and bidirectional GRU. For CNN-based models, the input comment is represented as a matrix where the rows are the dense vector representations of the comment words. While in RNNs, i.e. LSTM and GRU models, the comment is treated as a sequence of inputs which are the comment words and each input is a word vector.

In this section, we begin by describing the dataset and the preprocessing we conducted. Then, we explain the details of the data augmentation techniques we used and the models we implemented.

A. Data Preprocessing

For models training and evaluation, we used Wikipedia's talk page edits dataset provided by Toxic Comment Classification Kaggle competition. The dataset contains 159571 records of Wikipedia comments which have been labeled by human raters for toxic behavior. The data has six types of toxicity (six classes) which are: toxic, severe toxic, obscene, threat, insult and identity hate. We used 80% of the records for training, 10% for validation during models parameters tuning and 10% for testing.

In online user-generated content, spelling and grammar mistakes are quite common and some of them are even intentional. That is why we start our preprocessing by replacing some of the words with their formal equivalents besides removing stop words and punctuation marks. After that we build a vocabulary from the most frequent V words in the training set. Words that are not present in the vocabulary are given unknown-word index. We tried different values for V and we chose $V = 50,000$ words as it resulted in better classification performance on validation set.

Furthermore, we fix the input comment length to be exactly N tokens by padding shorter comments with a dummy word and truncating longer ones. By analyzing the comments lengths in the training set, we chose $N = 150$ words since most of the comments has $<= 150$ words.

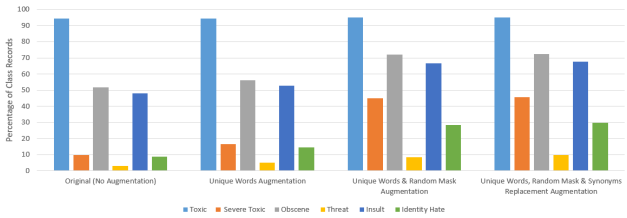

Fig. 1: Toxicity classes distribution in training set with and without performing different data augmentation methods.

Finally, each word in the input comment is replaced with its corresponding dense vector representation of D elements. This mapping is done using a look-up table of size $V \times D$ which is initialized by vector representation of the vocabulary words obtained from pre-trained FastText model [5] ($D = 300$ for FastText representations). These values are then updated during model training. For words that are not in the pre-trained 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.

B. Data Augmentation

Although the dataset has six classes of toxicity, it is highly imbalanced where about 89% of the samples are completely non-toxic (negative). That is the main reason we divided the solution into two levels by first detecting toxicity then determining its types to improve the performance. Additionally, the toxicity classes have very skewed distribution as shown in figure 1. This problem limited the model ability to learn how to distinguish and predict the rare categories which affected its overall evaluation results.

To overcome the classes imbalance problem, we applied data augmentation on training set by generating extra new samples from the comments that belong to the rare categories (severe toxic, threat and identity hate). We used the following methods to create new comments from existing ones.

- **Unique Words Augmentation:** For each comment of the minority classes, we remove duplicate words from it and create a new comment with only unique words.
- **Random Mask:** From a single comment, we create different new comments by randomly removing up to 20% of the original comment words.
- **Synonyms Replacement:** We create a new comment from an existing one by replacing the original comment words with their synonyms.

Figure 1 illustrates the changes in classes distribution in training set after applying the different augmentation methods. We can see that the rare classes: severe toxic, threat and identity hate have gained more records after applying data augmentation.

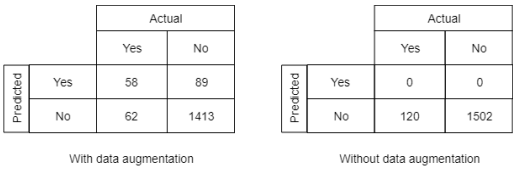

Fig. 2: Confusion matrices for identity hate class predicted using the same model trained with and without data augmentation.

TABLE I: Comparison of toxicity types classifier performance when trained using the different augmentation methods.

Augmentation Method	F_1**-score on Validation Set**
No Augmentation	0.8465
Unique Words Augmentation	0.8607
Unique Words and Random Mask	0.8752
Unique Words, Random Mask and Synonyms Replacement	0.8825

To prove the effectiveness of data augmentation, we trained the same CNN-based model using the original data without augmentation then we repeated the model training using data augmented with the different methods we discussed. The models evaluation results summarized in table I show improvements in the model overall performance after using each of the augmentation methods that achieves its best when the three methods are applied together. Moreover, figure 2 illustrates the improvement in the model performance at the class level. For identity hate class, a model trained using augmented data can distinguish comments with this type of toxicity while the same model trained without augmentation does not predict identity hate at all. Based on these results, we used data augmented using the three augmentation methods to build our models.

C. Models Implementation

We trained different models for toxic/nontoxic and toxicity types classifiers. We chose our baseline model to be logistic regression classifier based on NB-SVM features since it is a widely used strong baseline [6]. Using this method, we built a model for toxic/non-toxic classifier and another six models one for each toxicity type. The features are based on N-grams of 1 to 3 words.

Moreover, for each level we experimented five deep learning models. Two of them are variations of CNNs, two for bidirectional LSTMs and bidirectional GRUs and a model that ensembles CNNs, LSTMs and GRUs together. For CNN models, we started by using a single convolutional layer having a single kernel that goes through the input image constructed from the comment words embeddings. After that we built another CNN model using multiple convolutional layers (with different kernel sizes) that were applied to the same input then their output is concatenated before going through the fully connected hidden layers.

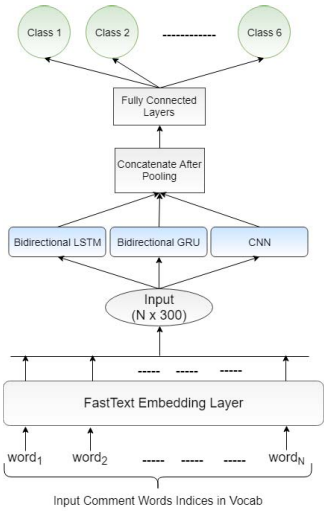

Fig. 3: The architecture of the ensemble model applied on toxicity type classification.

In addition to these methods, we provide an ensemble model that utilizes: bidirectional LSTM, bidirectional GRU and the best variation of CNNs together. Figure 3 illustrates the model architecture applied on the toxicity types classification problem, the same structure was used for toxicity detection as well.

For parameters tuning, we used grid search by training different models with different parameters values then choosing the values that yield the best results on the validation set. The best parameters values for toxic/nontoxic classifiers and toxicity types classifiers are summarized in table II.

As we can see from the models parameters values, the best kernel size of convolutional layer for toxicity types classifier (4×300) is less than that needed for toxic/non-toxic classifier (32×300). This indicates that to detect whether or not a comment is toxic requires examining larger window of words together by using a large kernel to make sure whether the comment is completely clean or has some sort of toxicity. But finding the specific types of toxicity in the comment only requires examining a small span of words since they can be easily defined based on words in small context.

IV. Models Evaluation

For models evaluation, simply using accuracy as the evaluation metric was not preferable because the data is imbalanced and the accuracy can be quite biased in such cases. Furthermore, area under the ROC curve (ROC AUC) is not a much better choice than accuracy because it may mask poor performance in cases like ours where most of the samples are negative (non-toxic at all). Therefore, we use F_1-score to measure our models performance. F_1-score can be interpreted as the harmonic mean of precision and recall values and it shows the real model performance even with skewed data [7]. For the sake of completeness, we also report the ROC

TABLE II: Models Parameters Values.

Model	Toxic/non-toxic Classifier		Toxicity Types Classifier	
	Special Layers	**Dense Layers**	**Special Layers**	**Dense Layers**
1. CNN	# filters = 300, kernel size = 32 × 300	50	# filters = 300, kernel size = 4 × 300	50
2. CNN Ensemble	# filters = 300, kernel sizes = 28, 32, 36 × 300	50	# filters = 300, kernel sizes = 2, 4, 6, 8 × 300	50
3. Bidirectional LSTM	# LSTM units = 30	80, 80, 80	# LSTM units = 30	60, 60
4. Bidirectional GRU	# GRU units = 50	100, 100	# GRU units = 40	80, 80
Ensemble of 1, 3 and 4	N/A	110, 110, 110	N/A	N/A
Ensemble of 2, 3 and 4	N/A	N/A	N/A	200, 200

TABLE III: Evaluation Results on Testing Set

(a) Toxic/non-toxic Classifier

Model	F_1-score	ROC AUC
NB-SVM Baseline	0.8028	0.9787
1. CNN	0.8251	0.9792
2. CNN Ensemble	0.8190	**0.9797**
3. Bidirectional LSTM	0.8146	0.9772
4. Bidirectional GRU	0.8174	0.9789
Ensemble model of 1, 3 and 4	**0.8282**	0.9793

(b) Toxicity Types Classifier

Model	F-measure	Micro Avg. F_1-score
NB-SVM Baseline	0.8560	0.8134
1. CNN	0.8675	0.8307
2. CNN Ensemble	0.8711	0.8338
3. Bidirectional LSTM	0.8598	0.8178
4. Bidirectional GRU	0.8643	0.8279
Ensemble model of 2, 3 and 4	**0.8724**	**0.8356**

AUC values but we use F_1-score for comparing the methods performance.

For toxicity types multi-label classifiers, we used two different evaluation methods. The first evaluates each label separately then averages the measure over all labels using the micro-average of F_1-score. Micro-average of F_1-score is the harmonic mean of precision and recall micro-averages [8].

The second metric uses the whole test set and calculates the average number of correctly classified samples. The following F-measure is presented in [8], for D a multi-label dataset, Y is the set of the actual labels, h is the multi-label classifier and Z is the set of predicted labels:

$$F(h, D) = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{2|Y_i \cap Z_i|}{|Y_i| + |Z_i|} \quad (1)$$

Table III summarizes the evaluation results of the different models for both classifiers. From the evaluation results, it is clear that deep learning models outperform the baseline and the proposed ensemble method effectively improves the solution performance. It gives 0.872 F_1-score for toxicity type classification and 0.828 for toxic/non-toxic classification which is better than any of the methods when applied separately. This is because training the three layers (CNN, LSTM and GRU) simultaneously in a single model allows them to cooperate to make more accurate classification decisions.

Additionally the CNN variations we applied affected the performance of the two classifiers. For toxicity types prediction, the CNN ensemble model improves the accuracy since using multiple kernels helped capturing different information from different windows of words. However, for toxic/non-toxic classification, the CNN network with single 32 × 300 kernel is better than the CNN ensemble network. This is because the kernels used include relatively large number of words (28, 32 and 36 words) with a small step between them (only four words) which does not add more information but rather causes distraction and badly affects the model accuracy. That is why we did not use it as part of the ensemble model but instead we used the single kernel CNN.

V. Conclusion

In this paper, we propose a multi-label classification scheme that can predict the different types of toxicity in an online comment. Additionally, we present data augmentation methods to overcome the class imbalance problem in the Wikipedia talk edits dataset. The proposed model is an ensemble of CNNs, bidirectional LSTMs and bidirectional GRUs trained together in a single model. Evaluation results show that ensemble model outperforms other methods with 0.828 and 0.872 F_1-score for toxic/non-toxic classification and toxicity types prediction respectively.

References

[1] Aboujaoude, Elias, et al. "Cyberbullying: Review of an old problem gone viral." Journal of Adolescent Health 57.1 (2015): 10-18.

[2] Georgakopoulos, Spiros V., et al. "Convolutional Neural Networks for Toxic Comment Classification." arXiv preprint arXiv:1802.09957 (2018).

[3] Kim, Yoon. "Convolutional neural networks for sentence classification." arXiv preprint arXiv:1408.5882 (2014).

[4] Wang, Xin, et al. "Predicting polarities of tweets by composing word embeddings with long short-term memory." Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Vol. 1. 2015.

[5] Joulin, Armand, et al. "Fasttext. zip: Compressing text classification models." arXiv preprint arXiv:1612.03651 (2016).

[6] Wang, Sida, and Christopher D. Manning. "Baselines and bigrams: Simple, good sentiment and topic classification." Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2. Association for Computational Linguistics, 2012.

[7] Jeni, Lszl A., Jeffrey F. Cohn, and Fernando De La Torre. "Facing imbalanced data–recommendations for the use of performance metrics." Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on. IEEE, 2013.

[8] Tsoumakas, Grigorios, and Ioannis Vlahavas. "Random k-labelsets: An ensemble method for multilabel classification." European conference on machine learning. Springer, Berlin, 2007.