Addressing Class Imbalance in Hate Speech Detection

Jinming Hu[1[0000-0001-6433-1157]], Yincheng Ren[1[1111-2222-3333-4444]] and Yuan Tian[1[1111-2222-3333-4444]]

[1] University of Virginia, Charlottesville VA 22904, USA

Abstract. This paper examines hate speech on Twitter and ways to detect them through machine learning algorithms. Our research explores the effects of 2 oversampling techniques and cost-sensitive learning on an imbalanced dataset using a single recurrent neural network. Our models classify hate speech by using word vectorization and feeding the data to a single layer RNN network. With a public English Twitter corpus of 20 thousand tweets with 19.4% sexist tweets and 11.9% racist tweets, we achieved F1 Scores of 0.883 for non-hate speech, 0.700 for sexist and 0.744 for racist tweets. This is higher than the results of state-of-art single neural network models.

Keywords: Hate speech detection, class imbalance, natural language processing

1 Introduction

Combating hate speech or abusive language use becomes significant in today's world as social media is more and more popular in our society. Social media companies like Twitter and Facebook have been actively combatting hate speech [1]. Current detection of hate speech involves a manual review of questionable documents, which not only is labor intensive but also introduces subjective notions of what constitutes hate speech. Therefore, major companies invested a lot to automatically detect hate speech by using machine learning algorithms due to the limitation of human resources. Numerous studies have also been conducted to solve this problem.

None of the recent papers on hate speech detection have addressed the issue of class imbalance in hate speech detection in general. Hate speech, by its nature, only accounts for a small portion of the total tweets. This would lead to lower prediction accuracies and recalls for the minority classes as the training set is inherently imbalanced [2]. We propose 2 classes of methods to solve the imbalance problem, namely oversampling and cost-sensitive learning. Both methods are shown to improve classifier performance on imbalanced datasets [3].

Oversampling increases the number of minority classes samples to make a more balanced dataset, allowing our neural networks to classify minority classes better. We used 2 oversampling techniques separately. The first sampling technique is random oversampling, which just adds instances of random data points of minority classes into the

dataset. It is the simplest and fastest oversampling method, which serves as the baseline oversampling method. The second is the SMOTE oversample technique which creates synthetic data points by using KNN. SMOTE has been shown to improve model performance by bringing in new data which fits the general distribution [4].

Cost-sensitive learning solves class imbalance by assigning higher costs to misclassifications of minority classes. This would make our neural network classify minority classes with higher accuracy.

2 Methods

2.1 Word Vectorization

The first step is to transform each tweet into an array of vectors that can be embedded into our neural networks. We utilized a pre-trained GloVe file provided by Stanford NLP researchers. The file is a dictionary which can translate each word to a vector of dimension 100. As the file is trained on general tweets, it is suitable for our analysis. We choose to vectorize by words instead of characters as words can better capture the semantic sentiments of the tweet. To account for the difference in the length of words in each tweet, we padded the tweets with dummy vectors to achieve a uniform size.

2.2 Oversampling

We used the Imblearn Python package for both random and SMOTE oversampling. In random oversampling, random data from the training set are repeated to create a balanced dataset. In SMOTE, synthetic data points are created for the minority classes to balance the classes. Borderline SMOTE is used to create additional samples which lie on the borders between classes. 5 nearest neighbors are considered when creating each data point. After the oversampling, undersampling is done though Tomek links to clean up some outliers. About 20 data points in each minority class are deleted.

2.3 Model Construction

Our models use a single LSTM layer to train. We constructed a Long Short-Term Memory (LSTM) layer with 200 hidden nodes. A dropout layer is added to reduce overfitting and the final dense layer is activated by a softmax layer. We used the ADAM optimizer to compile our model based on categorical cross-entropy. Due to the relatively small data size, we only trained the model for 4 epochs to prevent overfitting on the training data. The LSTM network directly classifies the 3 classes present in our data, namely, “racist”, “sexist” and “none”.

For Cost-Sensitive Learning, weights are assigned to each data sample when the model is trained. None hate-speech samples are assigned the base weight of 1, while racist and sexist samples are assigned a weight of 8 and 5 respectively.

3 Experiments and Results

3.1 Set Up

The base model only uses a LSTM network. The next 2 models oversample the data through random oversampling and SMOTE respectively before training. The final model does not alter the data but assigns higher weights to minority classes. The results are evaluated by the f1 scores for the 3 classes ("none", "racist", "sexist").

3.2 Results

Table 1. Table captions should be placed above the tables.

Set Up	Class	F1 Score
Base LSTM	Non-hate	0.874
Base LSTM	Sexist	0.675
Base LSTM	Racist	0.730
Random Oversampling	Non-hate	0.854
Random Oversampling	Sexist	0.655
Random Oversampling	Racist	0.750
SMOTE	Non-hate	0.873
SMOTE	Sexist	0.663
SMOTE	Racist	0.750
Cost-Sensitive Learning	Non-hate	0.883
Cost-Sensitive Learning	Sexist	0.700
Cost-Sensitive Learning	Racist	0.744

3.3 Evaluation

Cost-sensitive learning method produces the highest overall F1 scores as shown in table 1 and Fig 1. The results are better than the state-of-art models in Park and Fung (2017) as well as all of the non-ensemble methods in Batjatiya (2017).

Random oversampling raised the overall F1 score of racist class increased by 2.5%. However, this comes at the expense of lower F1 scores for non-hate speech class. Random oversampling only increases the proportion of minority class samples. As the samples added are from the existing samples, they provide no additional information that cannot be learned from the original data. This would increase the probability of overfitting the model [3]. Although minority classes classification is more accurate, the majority classification and overall accuracy dropped.

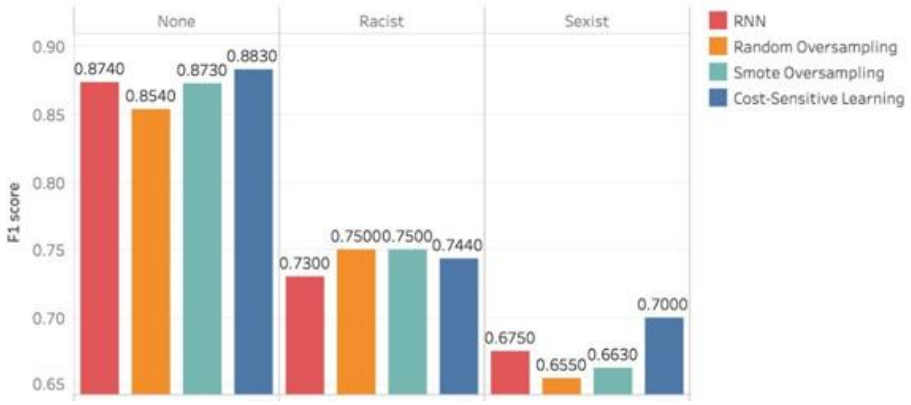

Fig. 1. F1 Scores Across Set-ups

SMOTE oversampling has slightly better results than random oversampling. The method creates new data points near the class borders, which provides more information than just random oversampling. It improved racist F1 scores by 0.02, but it was not effective for sexist tweets. This may be because that sexist tweets in our dataset are very scattered. It can be shown though ENN, a more aggressive cleanup technique, which deletes sexist data points which do not lie around other sexist data. Applied on the sexist data, ENN cleared more than 80% of all the sexist data. This shows the most of sexist data in our dataset are isolated. As SMOTE creates data points using KNN which uses the vector distance of data points, it would not work well on sexist tweets in our dataset. With a better dataset or better vectorization models, SMOTE has the potential to obtain better results.

Cost-sensitive learning produces the best method overall. It increased both racism and sexism F1 scores by 1.5% and 2.5% respectively. Weiss, McCarthy, Zabar (2007) showed that with a relatively large dataset, cost-sensitive learning consistently outperforms sampling methods for data with over 10000 samples. When the sample size is big enough, in our case 15000 samples, the classifier can estimate the class-membership probabilities more accurately. This allows accurate classification based on cost information. Another possible reason is that cost-sensitive learning handles the imbalance internally within the classifier. It does not affect the proportions in the training set like oversampling, which may increase the recall of minority classes at the expense of others. The data are all from real data points, which may be more relevant and consistent than synthetically generated data points.

3.4 Conclusions

Our oversampling and cost-sensitive learning methods have successfully improved the classification of minority classes. Cost-sensitive learning is the best method and has raised F1 scores for all 3 categories and produced better results than the current state-of-art. These methods can be applied to other classification tasks with class imbalance.

4 Discussions

4.1 Limitations

Due to the nature of hate speech, the amount data for hate speech would always be much lower than that of non-hate speech. With the imbalanced and limited data, it is infeasible to classify hate speech to the accuracy of non-hate speech. However, with our sampling and cost-sensitive learning methods, we mitigated the imbalance problem.

4.2 Future Work

Degree of Hate. Natural occurring hate speech have different degrees of hate. It is important to not only identify hate speech, but also determine the level of hate [6]. With the appropriate data, we can train a second layer of neural networks which further classifies hate speech into different degrees.

Semantic Features. Our current word vectorization only looks at the meaning of the words. However, certain semantic features are also found to be good indicators of sentiment [6] [7]. For example, all capital letters in tweets would likely mean extreme emotions like anger or excitement. We can incorporate these features in our word embeddings to better capture the sentiment of tweets.

5 Related Works

5.1 Hate Speech Detection

As online hate speech becomes an increasing problem, there has been many papers in recent years to address the problem. Waseem and Hovy (2016) generated a well-labelled dataset of 16 thousand tweets. The tweets are divided into 3 categories of non-hate speech, sexist tweets and racist tweets. The corpus is consistently labelled and relatively large, which makes it a good dataset for our experiments. The dataset is also used by many other hate speech detection papers including Badjatiya et al (2017), which makes it easy for us to compare performance.

Previous attempts to vectorize the words in tweets included using TF-IDF [10], Bag of Words and GloVe [9] [11] GloVe and Word2Vec are the better methods as they allow word generalizations for similar words [12]. Research has also showed that the best word embeddings are trained on the in-domain data as they can better capture the characteristics [13]. Hence, we used the Stanford GloVe files pretrained on Twitter for our analysis on tweets.

Warner and Hirschberg (2012) is one the earliest papers on hate speech detection. For a given word in a sentence, the authors considered the occurrences of other words and their distances to either side of the given word to generate a features vector for sentences. Although we vectorized tweets differently, we used a LSTM network that would also account for the relative sequence of words in each tweet for classification.

Many works compared different machine learning algorithms to best classify hate speech. It has been shown extensively that CNN and RNN models consistently perform better than other models [9] [15] [16]. Zhang and Luo (2018) showed that adding a GRU layer, a simplified LSTM layer, can improve the result of neural networks. Recent works also confirmed that LSTM models, which is an RNN model, produced much better results in speech classification compared to CNN and other networks [18]. Hence, we decided to use the LSTM model for our classifications. It is also a logical choice as the semantic meaning of a sentence are constructed by the sequence of words, which fits the LSTM model that considers the sequence of vectors.

5.2 Imbalanced Datasets

Due to the nature of hate speech, it makes up only a small portion of online speech. In our case, 70% of our data set is made up of non-hate speech. Natural proportion of classes in natural datasets are often not the best for classifiers to learn [19]. This is evident in our work as the 2 hate speech classes has significantly lower F1 scores than the non-hate speech class. However, none of the recent hate speech detection papers addressed this issue.

The first class of methods we use to address class imbalance is oversampling. Random oversampling is a quick method to address class imbalance, but it may lead to overfitting [19]. We used random oversampling as a baseline to compare with for other methods to address class imbalance. We chose Synthetic Minority Oversampling Technique (SMOTE) as our second method as it is shown to be a great oversampling method [3] [4]. SMOTE creates new data points based on the distribution of existing points, allowing to classifier to learn more information while preserving the general distribution of classes.

The second class of methods we chose to address class imbalance is cost-sensitive learning. Instead of changing the data set, cost-sensitive learning changes the misclassification costs for different classes instead. They are shown to improve classifier performance [20] [21]. For large data sets of over 15000 samples, cost-sensitive learning is shown to perform better than sampling methods [3]. As we have a relatively large dataset of about 16000 samples, cost-sensitive is suitable for our task.

References

1. Facebook, Google, Twitter Commit To Hate Speech Action In Germany, https://techcrunch.com/2015/12/16/germany-fights-hate-speech-on-social-media/, last accessed 2019/04/14
2. Haibo He, Edwardo A. Garcia: Learning from Imbalanced Data. IEEE Transactions on Knowledge and Data Engineering. (2009)
3. Weiss, McCarthy and Zabar: Cost-Sensitive Learning vs. Sampling: Which is Best for Handling Unbalanced Classes with Unequal Error Costs?. In: International Conference on Data Mining. pp.35-41. (2007).
4. N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer: SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, Volume 16, 321-357 (2002).
5. Sanjana Sharma, Saksham Agrawal, Manish Shrivastava: Degree based Classification of Harmful Speech using Twitter Data. eprint arXiv:1806.04197 (2018).
6. Joan Serra, Ilias Leontiadis, Dimitris Spathis, Gianluca Stringhini, Jeremy Blackburn, Athena Vakali.: Class-based Prediction Errors to Detect Hate Speech with Out-of-vocabulary Words. In: First Workshop on Abusive Language Online. (2017)
7. H. Watanabe, M. Bouazizi, and T. Ohtsuki: Hate Speech on Twitter: A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection. IEEEAccess. (2018).
8. Z. Waseem and D. Hovy: Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter. In: NAACL Student Research Workshop (NAACL SRW 2018), pp 88–93. (2016).
9. Badjatiya, P., Gupta, S., Gupta, M., & Varma, V: Deep learning for hate speech detection in tweets. In: 26th International Conference on World Wide Web Companion, pp 759-760. (2017).
10. T. Davidson, D. Warmsley, M. Macy, and I. Weber: Automated Hate Speech Detection and the Problem of Offensive Language. In: Eleventh International AAAI Conference on Web and Social Media, pp 512–515. (2017).
11. Gupta, Waseem. A Comparative Study of Embeddings Methods for Hate Speech Detection from Tweets (2018)
12. Anna Schmidt, Michael Wiegand: A Survey on Hate Speech Detection using Natural Language Processing. In: Fifth International Workshop on Natural Language Processing for Social Media, pp 1-10. (2017)
13. Rohan Kshirsagar, Tyus Cukuvac, Kathleen McKeown, Susan McGregor: Predictive Embeddings for Hate Speech Detection on Twitter. In: Abusive Language Online Workshop. (2018).
14. Warner, Hirschberg: Detecting hate speech on the world wide web. In: Second Workshop on Language in Social Media, pp 19-26. (2012).
15. Ji Ho P, Pascale F, V: One-step and Two-step Classification for Abusive Language Detection on Twitter. In: First Workshop on Abusive Language Online, pp 41–45. (2017)
16. Björn Gambäck, Utpal Kumar Sikdar: Using Convolutional Neural Networks to Classify Hate-Speech. In: First Workshop on Abusive Language Online, pp 85-90. (2017)
17. Ziqi Zhang, Lei Luo: Hate Speech Detection: A Solved Problem? The Challenging Case of Long Tail on Twitter. The Semantic Web Journal (2018)
18. Ona de Gibert, Naiara Perez, Aitor García-Pablos, Montse Cuadros: Hate Speech Dataset from a White Supremacy Forum. In: 2nd Workshop on Abusive Language Online. (2018).

19. Chawla: Data Mining for Imbalanced Datasets: An Overview. Data Mining and Knowledge Discovery Handbook. Springer (2005).
20. Nguyen Thai-Nghe, Zeno Gantner, Lars Schmidt-Thieme: Cost-sensitive learning methods for imbalanced data. In: The 2010 International Joint Conference on Neural Networks (IJCNN). (2010)
21. Charles X. Ling, Victor S. Sheng, C. Sammut (Ed.): Cost-Sensitive Learning and the Class Imbalance Problem. Encyclopedia of Machine Learning. Springer (2008)