HATEMINER at SemEval-2019 Task 5: Hate speech detection against Immigrants and Women in Twitter using a Multinomial Naive Bayes Classifier

Nikhil Chakravartula
Teradata, Hyderabad
nikhil.chakravartula@gmail.com

Abstract

This paper describes our participation in the SemEval 2019 Task 5 - Multilingual Detection of Hate. This task aims to identify hate speech against two specific targets, immigrants and women. We compare and contrast the performance of different word and sentence level embeddings on the state-of-the-art classification algorithms. Our final submission is a Multinomial binarized Naive Bayes model for both the subtasks in the English version.

1 Introduction

Twitter is a micro-blogging platform where people exchange ideas using short messages called tweets. Users can propagate their notions, including hatred against an individual or a group, to the entire global population with a latency of a few seconds. This poses a unique challenge of developing systems that can automatically identify and mitigate hate speech. Although twitter condemns hate speech through its hateful conduct policy[1], enforcing it is difficult. There are several reasons for this. Tweets often contain emoticons, emojis, language slangs, hashtags and other noisy data. Often, offensive and abusive language may be erroneously perceived as hate speech and hence it is important to distinguish offensive, abusive and hateful languages (Davidson et al., 2017; Waseem and Hovy, 2016). These problems are exacerbated by the fact that even humans find it difficult to delineate offensive and hateful language.

Many approaches have been put forward to detect hate speech. Bag of words and ngram features are effective in hate speech detection (Burnap and Williams, 2015; Warner and Hirschberg, 2012) as well as the detection of abusive and offensive content (Nobata et al., 2016). Gitari et al. (2015) used lexical resources to look up certain words that contribute significantly to hate speech but such features, when used in isolation may not be very effective. SVM (Burnap and Williams, 2015), Naive Bayes (Kwok and Wang, 2013) and Logistic Regression (Davidson et al., 2017) are some of the classifiers used in this domain.

Most of the above methods are targeted to detect general hate speech. Through this task, we aim to identify hate speech, specifically against *immigrants and women*. Frenda et al. (2018) used lexicon resources to identify misogynistic comments. Ahluwalia et al. (2018) used an ensemble of random forest, gradient boosting and logistic regression with bag of words, ngram and lexical features to discern hatred against women. We did not find significant work in detection of hate speech in English against immigrants.

2 Shared Task Description

The SemEval 2019 Task 5 is divided into two subtasks.

1. Subtask A, where systems must predict whether a tweet is hateful (HS=1) against immigrants and women.

2. Subtask B, where systems must first classify hateful tweets as aggressive (AG=1) or not, and secondly to identify the target harassed as an individual (TR=1) or generic.

We used the datasets provided by the organizers. Table 1 describes the composition of the dataset. Further details of the task are available in the task description paper (Basile et al., 2019).

3 System Description

3.1 Pre Processing

We perform the following pre-processing operations on the text before feature engineering.

[1] https://help.twitter.com/en/rules-and-policies/hateful-conduct-policy

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 404–408
Minneapolis, Minnesota, USA, June 6–7, 2019.

	HS=1	TR=1	AG=1
Train	57.9%	17.3%	14.9%
Dev	42.7%	20.4%	21.9%

Table 1: Dataset composition. HS: Hate Speech TR: Target AG:Aggressiveness

- All text is converted to lower case.
- All URLs, mentions, emojis and smileys are removed from the tweets. We used a python package tweet-preprocessor[2] to achieve this.
- All contractions are replaced with their full form. For example, *don't* will be replaced by *do not* and *can't* will be replaced by *can not*.
- All punctuation marks are removed.
- All numerical sequences are removed from the text.
- **Hashtag segmentation and spell correction:** Hashtags provide insights about a specific ideology by a group of people. These notions provide vital information for text classification, especially in the case of hate speech against immigrants and women. For example, hashtags like *#endimmigration*, often come from a group of people who are against immigrants. Segmentation (Segaran and Hammerbacher, 2009) of the hashtags is essential to allow the classifier to treat *#buildthatwall*, *#buildthewall*, *#buildthedamnwall*, *#buildwall*, etc with the same importance. After segmentation, *#buildthatwall* becomes *build that wall*, *#buildthedamnwall* becomes *build the damn wall* etc. Many tweets contain abusive words in elongated form, such as *f****kkkk*. We perform spell corrections (Jurafsky and Martin, 2018) on these words to reduce the vocabulary size and to account for better results. Text8[3] is utilized to generate unigram and bigram word statistics with ekphrasis (Baziotis et al., 2017) to perform both these operations.
- **Stemming**: Stemming is the process of reducing a word to its base root form. We used Porter Stemmer[4] from NLTK (Steven Bird and Loper, 2009) to stem. Stemming is used in combination with the Naive Bayes classifier. For other classifiers, pretrained word embeddings without stemming are used.

3.2 Feature Engineering

The following features are considered in our experiments.

- **Bag of words (BoW):** Bag of words is used to represent the presence of word n-grams.
- **Word Embeddings:** Glove840B - common crawl, GloveTwitter27B - twitter crawl (Pennington et al., 2014) and fasttext - common crawl (Mikolov et al., 2018) pre-trained word embeddings are used to analyze their impact on the classification.
- **Sentence Embeddings:** Infersent (Conneau et al., 2017) is used to produce sentence level embeddings. InferSent is a sentence embedding method that provides semantic representations for English sentences. It is trained on natural language inference.

4 Experiments

In this section, we describe the experimental settings used in our research. All our code is publicly available in a github repository.[5]

4.1 Evaluation Metrics

The evaluation metrics for subtask A are precision(HS), recall(HS) and F_1-score(HS). Macro averaged F_1-score(HS,TR,AG) and Exact Match Ratio (EMR) are the evaluation metrics for subtask-B. Submissions are ranked based on F_1-score(HS) and EMR for subtask-A and subtask-B, respectively.

4.2 Methodology

All the experiments are developed using the Scikit-Learn (Pedregosa et al., 2011) machine learning library. Five-fold cross validation score on the train set used to evaluate our models. We ran several experiments on various classification algorithms. The best performing classifiers were Naive Bayes, logistic regression, SVM and XGBoost. The following are the details of the classifier settings.

[2] https://github.com/s/preprocessor

[3] http://mattmahoney.net/dc/textdata.html

[4] https://tartarus.org/martin/PorterStemmer/

[5] https://git.io/fhFGR

WordVector	Logreg			SVM			XGB		
	F_{avg}(HS)	$F_{avg(HS,TR,AG)}$	EMR	F_{avg}(HS)	$F_{avg(HS,TR,AG)}$	EMR	F_{avg}(HS)	$F_{avg(HS,TR,AG)}$	EMR
glove	0.58	0.57	0.49	0.54	0.54	0.45	0.61	0.56	0.44
fasttext	0.58	0.56	**0.53**	0.55	0.52	0.45	0.61	0.56	0.43
glove twitter	0.69	0.67	0.48	**0.69**	0.61	0.45	0.69	0.65	0.44
glove + infersent	**0.73**	**0.70**	0.47	0.66	**0.64**	**0.46**	**0.72**	**0.68**	**0.46**

Table 2: Pretrained word and sentence embeddings results. For each classifier family, the best score is made bold.

Word ngrams	Stem	Binary	$F_{avg(HS)}$	F_{avg} (HS,TR,AG)	EMR
1,2	false	true	0.69	0.66	0.45
1,2	false	false	0.68	0.66	0.45
1,2	true	true	**0.69**	**0.67**	**0.47**
1,2	true	false	0.68	0.66	0.45
1,3	false	true	0.69	0.66	0.45
1,3	true	true	0.69	0.66	0.47
1,4	false	true	0.69	0.66	0.45
1,4	true	true	0.69	0.66	0.47

Table 3: Multinomial Naive Bayes Classifier results with word ngram range, stemming and binarization

Classifier	$F_{1(HS)}$	F_1 (HS,TR,AG)	emr
glove twitter+logreg	0.70	0.70	0.48
glove twitter+XGB	0.69	0.66	0.55
glove+infersent+ XGB	0.72	0.69	0.55
glove+infersent+ logreg	0.72	0.72	0.53
stem+ NB binarized+ word-ngrams(1.2)	**0.74**	**0.73**	**0.57**

Table 4: Results on the dev set

- **Logistic Regression, SVM and XGBoost** Word or sentence level embeddings are fed as inputs to these classifiers. In the absence of a sentence embedder, we averaged all the word vectors to get a vector representation of the tweet. For logistic regression, the solver is liblinear (Fan et al., 2008) and L2 norm is used for penalization. For SVM, inputs are normalized using a soft scaling scheme and the kernel used is a Radial Basis Function (Buhmann and Buhmann, 2003). The default parameters are kept as is for XGBoost[6]. Table 2 shows the results.

- **Naive Bayes classifier:** Multinomial Naive Bayes classifier, along with the bag of words generated with CountVectorizer[7] gave better results than other Naive Bayes variations. Different runs are carried out to tune the parameters as shown in Table 3.

[6] https://xgboost.readthedocs.io/en/latest/python/python_api.html

[7] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html

5 Results and Analysis

We wanted to submit a single system for both the subtasks. Hence, our goal was to maximize all three metrics: F_1(HS), F_1(HS,TR,AG) and EMR. The results show that there is no single variation that defeats the others in all the metrics combined. Logistic regression with glove and infersent performed the best in F_1(HS) and F_1(HS,TR,AG), but only with an acceptable EMR. Regarding the XGBoost family, glove with inferesent version outperforms the rest in all the metrics. Stemmed-binarized Naive Bayes classifier with ngram range (1,2) performed better in F_1(HS) and EMR in the Naive Bayes family. The Glove-Twitter version of logreg and XGboost aren't too far behind as well. We applied all these high performing models on the dev set to analyse their performance further. The results are shown in Table 4. Naive Bayes comfortably achieved the highest score on the dev set on all the three metrics as shown in Table 4. Hence, we finalized the Naive Bayes model as our official submission. This submission scored an F_1(HS) of 0.405 in subtask-A, F_1(HS,TR,AG) of 0.54 and EMR of 0.296 in subtask-B.

6 Conclusion and Future Work

The aim of this research was to detect hate speech against two specific targets, immigrants and women. We described a naive bayes classifier system and also elucidated our trials of using different pre-trained word and sentence level embeddings on the state-of-the-art classification algorithms. In the future, we would like to include lexicon-based, Parts Of Speech features to further investigate the performance of these classifiers. We would also like to evaluate how deep learning approaches respond to this task.

References

Resham Ahluwalia, Himani Soni, Edward Callow, A. Nascimento, and Martine De Cock. 2018. Detecting hate speech against women in english tweets. In *EVALITA@CLiC-it*.

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Rangel, 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 (SemEval-2019)*. Association for Computational Linguistics", location = "Minneapolis, Minnesota.

Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. 2017. Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 747–754, Vancouver, Canada. Association for Computational Linguistics.

Martin D. Buhmann and M. D. Buhmann. 2003. *Radial Basis Functions*. Cambridge University Press, New York, NY, USA.

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.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Thomas Davidson, Dana Warmsley, Michael W. Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. *CoRR*, abs/1703.04009.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Simona Frenda, Bilal Ghanem, Estefanía Guzmán-Falcón, Manuel Montes-y-Gómez, and Luis Villaseñor Pineda. 2018. Automatic expansion of lexicons for multilingual misogyny detection. In *Proceedings of the Sixth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2018) co-located with the Fifth Italian Conference on Computational Linguistics (CLiC-it 2018), Turin, Italy, December 12-13, 2018.*

Njagi Dennis Gitari, Zhang Zuping, Hanyurwimfura Damien, and Jun Long. 2015. A lexicon-based approach for hate speech detection.

Daniel Jurafsky and James H. Martin. 2018. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall.

Irene Kwok and Yuzhou Wang. 2013. Locate the hate: detecting tweets against blacks. In *AAAI 2013*.

Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. 2016. Abusive language detection in online user content. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, pages 145–153, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

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

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Toby Segaran and Jeff Hammerbacher. 2009. *Beautiful Data: The Stories Behind Elegant Data Solutions*. O'Reilly Media, Inc.

Ewan Klein Steven Bird and Edward Loper. 2009. *Natural Language Processing with Python– Analyzing Text with the Natural Language Toolkit*. O'Reilly Media, Inc.

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*, LSM '12, pages 19–26, Stroudsburg, PA, USA. Association for Computational Linguistics.

Zeerak Waseem and Dirk Hovy. 2016. Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *Proceedings of the NAACL Student Research Workshop*, pages 88–93. Association for Computational Linguistics.