ltl.uni-due at SemEval-2019 Task 5: Simple but Effective Lexico-Semantic Features for Detecting Hate Speech in Twitter

Huangpan Zhang, Michael Wojatzki, Tobias Horsmann and Torsten Zesch
Language Technology Lab, University of Duisburg-Essen
{huangpan.zhang, michael.wojatzki}@uni-due.de
{tobias.horsmann, torsten.zesch}@uni-due.de

Abstract

In this paper, we present our contribution to SemEval 2019 Task 5 *Multilingual Detection of Hate*, specifically in the Subtask A (English and Spanish). We compare different configurations of shallow and deep learning approaches on the English data and use the system that performs best in both sub-tasks. The resulting SVM-based system with lexico-semantic features (n-grams and embeddings) is ranked 23rd out of 69 on the English data and beats the baseline system. On the Spanish data our system is ranked 25th out of 39.

1 Introduction

Hateful, abusive, or offending statements which target individuals or groups on the basis of characteristics such as gender, nationality, or sexual orientation are called *hate speech* (Basile et al., 2019). Social media is particularly affected by hate speech, as it is known to poison the communication climate, build up negative sentiment towards groups of people, or even lead to real-life consequences (Warner and Hirschberg, 2012; Waseem and Hovy, 2016; Schmidt and Wiegand, 2017; Wojatzki et al., 2018; Benikova et al., 2017; Ross et al., 2017).

In this work, we present our submission to the *SemEval 2019 Task 5: Multilingual Detection of Hate* (Subtask A) for English and Spanish. The objective in Subtask A was to build a system which is able to predict whether given tweets in English or in Spanish are hateful or not hateful towards women or immigrants.

We develop a hate speech detection system by experimenting with a range of classifiers which are either based on engineered features or on neural network architectures. We systematically compare the performance of these different detection systems and for our final submission (for both English and Spanish) we use the model that performs best on the English training data. Our best system is a SVM equipped with n-gram features and fastText (Mikolov et al., 2018) embeddings. Our system obtains the 23rd rank (out of 69) on the English dataset and the 25th rank (out of 39) on the Spanish dataset.

2 System Description

For our submission, we compare a wide range of different neural and non-neural systems in terms of their performance. Our actual submission system is the system that performed best in this evaluation. We will now describe both our neural network approaches and the feature-engineering approaches for detecting whether tweets are hateful towards women or immigrants (Subtask A). We developed and evaluated the approaches for the English dataset and applied the best performing system as-is to the Spanish data. We now first briefly describe the provided data and then discuss the prediction approaches in more detail.

Dataset In Subtask A, the English training set consists of 9,000 tweets and the development set consists of 1,000 tweets. The Spanish training set consists of 5,000 tweets, the development set consists of 500 tweets. In the test data, there are 2,971 English and 1,600 Spanish tweets. For each tweet, the task organizers provided a binary annotation indicating whether a tweet is hateful or not hateful towards a given target (i.e. women or immigrants). An example for the label hateful (towards immigrants) is the tweet:

> *This immigrant should be hung or shot! Period! Animal. https://t.co/wFcGoLCqJ5*

An example for the label not hateful is the following tweet:

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 441–446
Minneapolis, Minnesota, USA, June 6–7, 2019.

Don't mess with these migrant dads #SkimmLife https://t.co/swVmkTlFRz via @theSkimm.

For more details on the dataset and its creation, we refer to the overview paper of the shared task (Basile et al., 2019).

Preprocessing In almost all of our classification approaches, we vectorize the tweets based on word occurrences. Hence, we tokenize the tweets with the twitter specific tokenizer provided by Owoputi et al. (2013). We decided not to remove or normalize social media specific phenomena such as @-mentions, #-hashtags, URLs, and emojis as we hypothesize that these phenomena may provide useful signals for classification. For example, it is conceivable that a reference to the twitter-handle of Donald Trump (*@realDonaldTrump*) may indicate hatred towards immigrants.

2.1 Feature Engineering Approaches

We now report on those approaches that are based on traditional machine learning algorithms and that represent the train and test instances using manually crafted and engineered features. The explored machine learning algorithms are: SVM (LibSVM by Chang and Lin (2011), XGBoost (Chen and Guestrin, 2016), RandomForest (Witten et al., 2016) and Vowpal Wabbit.[1]

We implement the classifiers using the text classification framework DKPro TC (Daxenberger et al., 2014) which includes all of the above-mentioned classifiers. We use the following features to represent the tweets:

N-grams As a baseline feature, we represent the tweets using word and character n-grams. We experiment with n-gram sizes in the range from 1-3 for word n-grams and 2-5 for character n-grams. To reduce the feature space, we only use the n-grams that are most common in the (English and Spanish) training data. We experiment with the frequency cut-off values of 200, 500 and 1,000.

Hateword lists We hypothesize that the presence of specific hate or insult words gives an indication of whether a tweet constitutes hate speech. Hence, we check if the words in the tweets occur in lists of hate or insult words. We use the word lists provided by Wiegand et al. (2018), which contain a basic word list and a extended word list. There are 1,650 words in basic list with binary labels (abusive or not), and 8,478 words in extended list with a numeric weight. We extract abusive words to use in the following features: a) a **boolean hateful** feature if a posting contains any word contained in the basic list, b) a **hatefulness ratio** of total words to hateful words, and c) the sum of the **hatefulness weights** based on the extended list.

Sentiment We also suspect that the tone in which a tweet is composed can be an indication for hate speech. For instance, we assume that tweets that have a strong positive sentiment are rarely hate speech. To measure the overall sentiment of tweets, we use the tool by Socher et al. (2013) to compute a sentiment score for each tweet. The computed sentiment score uses a five-degree scale from very positive to very negative.

Word embeddings We use pre-trained word embeddings to enhance our tweet representation with a semantic component. For computing semantic features, we first average the 300-dimensional (Spanish or English) word embeddings provided by Mikolov et al. (2018) of all words of a tweet. Next, we use every dimension of the averaged vector as a feature.

2.2 Neural Network Approaches

Besides traditional machine learning approach, we also experiment with neural network architectures: multilayer perceptrons (MLP), convolutional neural networks (CNN), bi-directional LSTMs and a combination of LSTMs and CNNs (LSTM + CNN). We initialize all setups with the 300-dimensional word embeddings provided by Mikolov et al. (2018), which were trained on the common crawl corpus. Furthermore, in all setups, we use a dropout of 0.25 after the embedding layer and update network weights using the *Adam* optimizer (Kingma and Ba, 2014). For all architectures, we have optimized the hyperparameters (e.g. number and size of layers) on an held-out development set. We here report only the best-found parameterization.

MLP Besides the final softmax layer, our MLP has a total of 6 densely connected layers. Starting from the input, the layers have 256, 128, 64, 32, 16 and 8 nodes. We use *relu* as activation function in all layers.

[1] https://github.com/VowpalWabbit/vowpal_wabbit

CNN Our CNN uses three stacked convolutional layers that use a filter size of two. The first layer has 128 nodes, the second 64 and the third 32. Subsequently, we apply *max pooling*, a dense layer with ten nodes and the final softmax classification layer.

LSTM At the core of our LSTM is a bidirectional LSTM layer with 128 nodes. This layer is followed by two dense layers (40 and 10 nodes) and the softmax layer.

LSTM + CNN For the combination of LSTM and CNN, we put our CNN model on top of LSTM model.

All of the above-described architectures are implemented using deepTC (Horsmann and Zesch, 2018) with the Keras (Chollet et al., 2015) and Tensorflow (Abadi et al., 2015) backend.

BERT We also experiment with Bidirectional Encoder Representations from Transformers (BERT), which recently excelled in a number of NLP tasks (Devlin et al., 2018). For our experiments, we use the provided pre-trained multilingual-cased BERT-Base model,[2] a maximum sequence-length of 128 and batches of 32 instances. In the described configuration, BERT yields an accuracy of 0.66 after fine-tuning for the second time. As we observe that the performance of BERT begins to decrease from the third fine-tuning, we do not fine-tune the model furthermore.

3 Model Selection and Results

We evaluate each of the proposed prediction approaches in a 10-fold cross-validation on the English training dataset to determine the best performing one. As baseline, we use an SVM equipped with word unigram feature.

For all our approaches, we optimize the hyperparameters (e.g. SVM's slack variable or number of layers in neural networks) and feature configurations (e.g. frequency cut-offs for n-gram features) on the training data and report the best performance for each approach. We start with fine-tuning the n-gram features. We test a wide range of different combinations of n-gram sizes and frequency cut-offs with different classifiers. We report the results in Table 1. We use **wn** and **cn** as the abbreviations of word n-grams and character n-grams.

	macro-F_1	Best n-gram Combination
LibSVM	0.780	wn=1 / topk=1000 cn=2-4 / topk=200
Random Forest	0.771	wn=1-2 / topk=500 cn=2-4 / topk=1000
XGBoost	0.764	wn=1-2 / topk=1000 cn=2-5 / topk=1000
Vowpal Wabbit	0.742	wn=1-3 / topk=1000 cn=2 / topk=200

Table 1: Results for Fine-tuned n-gram Features.

	macro-F_1	accuracy
Baseline	0.693	0.692
LibSVM	0.787	0.794
LSTM + CNN	0.744	0.768
MLP	0.741	0.750
CNN	0.740	0.746
LSTM	0.674	0.688
BERT	0.660	0.660

Table 2: Results for 10-folds Cross-validation on the Training Dataset for English.

We find that SVM has the overall best performance based on cross-validation, and we continue our experiment (hateword lists, sentiment, word embeddings) using LibSVM with the best n-gram setup. We compare this best feature-engineered system in Table 2 with the neural approaches.

Overall, we observe that the approaches based on feature engineering tend to outperform the neural approaches. As our SVM classifier performs best, we select it as our official submission and also apply it to the Spanish data. Interestingly, in our experiments, BERT and and LSTM perform worst by a considerable margin. However, the combination of LSTM and CNN shows to be competitive with feature engineering approaches.

In Table 3, we show how our system performs on the official test data. We observe a dramatic drop of 30.5 percentage points between performance on the English training and test set. We attribute this loss to the over-fitting to the training data. Nevertheless, our system is able to outperform the most frequent class baseline substantially and especially on the Spanish data the absolute difference to the top-scoring system is low (about 3 percentage points). This means that our system is indeed effective in the task at hand, but also that hate speech detection is a very challeng-

[2] https://storage.googleapis.com/bert_models/2018_11_23/multi_cased_L-12_H-768_A-12.zip

	English	Spanish
Most Frequent Class	0.367	0.370
SVM (baseline)	0.451	0.701
SVM (ours)	0.475	0.696
Top-scoring Team	0.651	0.730

Table 3: Results in Terms of macro-F_1 on the English and Spanish Testset.

Feature Set	macro-F_1
All Features	0.785
-N-grams	0.724
-Word Embeddings	0.778
-Hatefulness Ratio	0.785
-Boolean Hateful	0.785
-Sentiment	0.787
-Hatefulness Weights	0.787

Table 4: Feature Ablation for Our SVM Classifiers.

ing task.

Feature ablation To understand how important the individual features are for our system's performance, we conduct an ablation test for our feature set. We show the results of this ablation in Table 4. The results show that the absence of all features except n-grams and word embeddings leads to an improvement in performance. Consequently, we only use n-grams and word embeddings for our final model. The results also show that n-grams are the most important feature for our model.

4 Distribution of Hate Indicators

When comparing the performance of our system between the training data and test data, we notice a dramatic drop of 30.5 percentage points on macro-F_1. To better understand this drop, we examine the distribution of words for which we suspect that they are good indicators for hate speech – i.e. words which both occur frequently in the data and are commonly seen as a highly offensive words. We examine a frequency distribution of all words and find that the word *'bitch'* meet these criteria. However, the distribution of this word is significantly different in train data and test data. To see whether this is a special case, we examine another high frequency word *'fuck'*. The result is shown in Table 5.

Furthermore, we inspect how these words are distributed across the classes hate speech and not hate speech in both the train and the test set. We visualize this analysis in Table 6.

	Train	Test
Bitch	1,115 in 9,000	1,134 in 2,971
Fuck	675 in 9,000	260 in 2,971

Table 5: Distribution of postings contain *Bitch* and *Fuck*.

	Train		Test	
	Hate Speech		Hate Speech	
Word	Yes	No	Yes	No
Bitch	0.78	0.22	0.44	0.57
Fuck	0.59	0.41	0.57	0.44

Table 6: Hate/not-hate Class Distribution of Postings Contain *Bitch* and *Fuck*.

For the word *'bitch'*, we observe that – in the training data – its occurrence is strongly correlated (the probability is about 0.8) with the class hate speech. In the test set, however, this correlation is considerably weaker. As a result, it is very likely that our classifier will learn that *'bitch'* is a strong evidence for hate speech. As the correlation is different in the test data, this heuristic is likely to lead to misclassification. We conclude that our classifier, which makes strong use of lexical features, is too sensitive to such distributions. Note, that we do not find such a shift for the word *'fuck'*.

5 Conclusion

We present **ltl.uni-due** our submission to SemEval 2019 Task 5 *Multilingual Detection of Hate*. For building our system, We systematically compare a wide range of approaches – including neural network approaches such as LSTMs and BERT and approaches which are based on feature engineering. In our experiments a comparably simple classifier – a SVM equipped with lexico-semantic features (n-grams and word embeddings) – outperforms all other approaches. A comparison between performance on training and test data as well as a quantitative analysis of the dataset shows that our comparably simple classifier is prone to over-fitting, but nevertheless delivers competitive performance in this highly challenging task.

Acknowledgments

This work was supported by the Deutsche Forschungsgemeinschaft (DFG) under grant No. GRK 2167, Research Training Group "User-Centred Social Media".

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. Software available from tensorflow.org.

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.

Darina Benikova, Michael Wojatzki, and Torsten Zesch. 2017. What Does This Imply? Examining the Impact of Implicitness on the Perception of Hate Speech. In *International Conference of the German Society for Computational Linguistics and Language Technology*, pages 171–179. Springer.

Chih-Chung Chang and Chih-Jen Lin. 2011. Libsvm: A Library for Support Vector Machines. *Acm Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A Scalable Tree Boosting System. In *Proceedings of the 22Nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, pages 785–794. ACM.

François Chollet et al. 2015. Keras. https://keras.io.

Johannes Daxenberger, Oliver Ferschke, Iryna Gurevych, and Torsten Zesch. 2014. DKPro TC: A Java-based Framework for Supervised Learning Experiments on Textual Data. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 61–66. Association for Computational Linguistics.

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*.

Tobias Horsmann and Torsten Zesch. 2018. DeepTC – An Extension of DKPro Text Classification for Fostering Reproducibility of Deep Learning Experiments. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, pages 2539 – 2545.

Diederik Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint:1412.6980*, pages 1–13. Accessible at https://arxiv.org/abs/1412.6980; last accessed November 27 2018.

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)*.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith. 2013. Improvedppart-of-Speech Tagging for Online Conversational Text with Word Clusters. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–390.

Björn Ross, Michael Rist, Guillermo Carbonell, Benjamin Cabrera, Nils Kurowsky, and Michael Wojatzki. 2017. Measuring the Reliability of Hate Speech Annotations: The Case of The European Refugee Crisis. *arXiv preprint arXiv:1701.08118*.

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.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642.

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.

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.

Michael Wiegand, Josef Ruppenhofer, Anna Schmidt, and Clayton Greenberg. 2018. Inducing a Lexicon of Abusive Words–a Feature-Based Approach. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 1046–1056.

Ian H Witten, Eibe Frank, Mark A Hall, and Christopher J Pal. 2016. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann.

Michael Wojatzki, Tobias Horsmann, Darina Gold, and Torsten Zesch. 2018. Do Women Perceive Hate Differently: Examining the Relationship Between Hate Speech, Gender, and Agreement Judgments. In *Proceedings of the Conference on Natural Language Processing (KONVENS)*, pages 110–120.