Hate Speech Detection on Twitter: Feature Engineering v.s. Feature Selection

David Robinson[1], Ziqi Zhang[2(✉)], and Jonathan Tepper[1]

[1] Nottigham Trent University, Nottingham, UK
david.robinson2015@my.ntu.ac.uk, jonathan.tepper@ntu.ac.uk
[2] University of Sheffield, Sheffield, UK
ziqi.zhang@sheffield.ac.uk

Abstract. The increasing presence of hate speech on social media has drawn significant investment from governments, companies, and empirical research. Existing methods typically use a supervised text classification approach that depends on carefully engineered features. However, it is unclear if these features contribute equally to the performance of such methods. We conduct a feature selection analysis in such a task using Twitter as a case study, and show findings that challenge conventional perception of the importance of manual feature engineering: automatic feature selection can drastically reduce the carefully engineered features by over 90% and selects predominantly generic features often used by many other language related tasks; nevertheless, the resulting models perform better using automatically selected features than carefully crafted task-specific features.

1 Introduction

In recent years, social media has been increasingly exploited for the propagation of hate speech and the organisation of hate based activities [1]. Although social media companies are spending millions of euros every year on manually reviewing online contents and deleting offensive materials [3,7], they are still criticised for not doing enough and facing increasing pressure to address this issue.

The pressing situation has attracted increasing research using semantic content analysis techniques based on Natural Language Processing (NLP) and Machine Learning (ML) [1–4,8] to develop scalable, automated methods of hate speech detection. Substantial effort has been spent on developing novel, effective features (**feature engineering**) that better capture hate speech on the social media [1,2,6,7]. However, little work is done to understand how these distinctive features have - or not - contributed to the task and whether a **feature selection** process can further enhance the performance of such methods. This work fills this gap by analysing the effect of automatic feature selection on state-of-the-art ML based methods for hate speech detection using Twitter as a case study. We show surprising insights that challenge our existing perception of the importance of feature engineering. We prove that on this specific task, the automatic feature

A. Gangemi et al. (Eds.): ESWC 2018 Satellite Events, LNCS 11155, pp. 46–49, 2018.
https://doi.org/10.1007/978-3-319-98192-5_9

selection algorithm drastically reduces the carefully engineered feature space by over 90%, but improves ML algorithms to perform better using automatically selected features that are predominantly generic and used by many other tasks.

We structure the remainder of this work as follows: related work in Sect. 2, methodology in Sect. 3, experiments and conclusion in Sects. 4 and 5.

2 Related Work

State-of-the-art typically cast hate speech detection as a supervised text classification task [5]. These can be either **classic methods** that rely on manually engineered features consumed by ML algorithms such as SVM [1,2,6,7]; or **deep neural networks** (**DNN**) based methods that automatically learn multi-layers of abstract features from raw data [3,4,8]. While our earlier work [8] looked at DNN based methods, here we study the effects of feature engineering in classic methods.

Feature engineering is the process of analysing and designing predicative features for classifying hate speech. A wide range of features have been summarised in [5]. In short, these can include *simple surface features* such as word n-grams; *word generalisation* using, e.g., word clusters; *lexical resources* such as lists of abusive words; *linguistic features* such as Part of Speech (PoS) and dependency relations; *knowledge-based features* such as stereotypical concepts in a knowledge base; and *multimodal information* such as image captions. However, it is unclear how these different types of features contribute to the performance of the classifier. Most methods simply 'use them all', which creates high-dimensional, sparse feature vectors - particularly for short texts such as Tweets - that are prone to over-fitting.

3 Methodology

Our method is based on a state-of-the-art linear SVM based hate speech classifier introduced in [2]. It uses a number of different types of features, which are: (1) *surface* features, including word unigrams, bigrams and trigrams each weighted by TF-IDF, and filtered by a minimum frequency of 5; number of mentions (**#mentions**), and hashtags (**#hashtags**); number of characters, and words; (2) *linguistic* features, including Part-of-Speech tag unigrams, bigrams (i.e., two consecutive PoS tags), and trigrams, also weighted and filtered the same way as above; number of syllables; Flesch-Kincaid Grade Level (**FKGL**) and Flesch Reading Ease (**FRE**) scores to measure the 'readability' of a document; and (3) *sentiment* feature in terms of sentiment polarity scores of the tweet.

Extending this, we add additional surface based features as follows: the ratio between the number of misspelled words and the number of all words in the tweet; the number of emoji's (based on regular expressions); the number of special punctuations such as question and exclamation marks as they can be used as an expletive; the percentage of capitalised characters; and the lowercase hashtags from tweets.

We use **SVM** to denote the model using the **Original** feature set, and **SVM+** to denote that using both the original and extended feature sets (**enhanced**). Next, we use a state-of-the-art feature selection process based on Logistic Regression with L1-regularization as the estimator on the training data[1]. This calculates a 'feature importance' score for each feature, which is discarded if its score is below the default threshold. We use $\mathbf{SVM_{fs}}$ and $\mathbf{SVM_{fs}}$+ to denote the SVM and SVM+ model with feature selection respectively.

4 Experiment

We use a total of 7 **public datasets** compiled in [8]. Briefly, **WZ-L** contains over 16k Tweets annotated for 'sexism', 'racism', and 'neither' [7], **WZ-S.amt** and **WZ-S.exp** contain the same set of some 6k Tweets annotated for the same classes by different groups of people [6]; **WZ-S.gb** merges WZ-S.amt and WZ-S.exp [3]; **WZ-LS** merges WZ-L and WZ-S.exp [4]; **DT** [2] and **RM** [8] each contains some 24k and 2k Tweets classified into hate or non-hate. We also use the CNN+GRU deep learning model described in [8] as state-of-the-art reference. For each dataset, we split it into 75:25 to use 75% for parameter tuning using 5-fold cross-validation experiments, and test the optimised model on the 25% held-out data. We report our results using in micro F1 in Table 1.

Table 1. Comparing micro-F1 on the different models (best figures in **bold**). The shaded columns show the percentage of features retained after feature selection

Dataset	**SVM**	$\mathbf{SVM_{fs}}$	%Features	**SVM+**	$\mathbf{SVM_{fs}}$+	%Features	**CNN+ GRU** [8]
WZ-L	0.74	0.81	5.1%	0.74	0.81	5.1%	**0.82**
WZ-S.amt	0.86	0.87	3.4%	0.91	0.90	3.1%	**0.92**
WZ-S.exp	0.89	0.90	3.9%	0.90	0.91	3.9%	**0.92**
WZ-S.gb	0.86	0.91	3.4%	0.87	0.90	3.2%	**0.93**
WZ-LS	0.72	0.81	4.4%	0.73	0.81	4.0%	**0.82**
DT	0.87	0.89	4.4%	0.86	0.90	3.8%	**0.94**
RM	0.86	0.89	0.7%	0.88	0.89	0.6%	**0.92**

Table 1 shows that, comparing $\mathbf{SVM_{fs}}$ against SVM, or $\mathbf{SVM_{fs}}$+ against SVM+, clearly feature selection can further enhance the performance of the linear SVM classifier on this task. Sometimes the improvement due to feature selection can be quite significant (e.g., WZ-LS). Although none of the SVM based classifiers can outperform the CNN+GRU model, on the WZ-L, WZ-S.exp and WZ-LS datasets, the feature selected models can get very close to state-of-the-art performance. Table 1 also shows that after applying feature selection, the majority of both the Original and Enhanced features are discarded. In some

[1] http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectFromModel.html.

cases (e.g., RM), the reduction is quite extreme. This has however, improved classification accuracy. Further analysis shows that, out of the Original feature set, features such as #mentions, #hashtags, FRE, and FKGL are completely discarded on all datasets. Word and PoS n-grams are the most predictive features as they are selected on all datasets. Other feature types appear to be only useful on isolated cases (i.e., 1 or 2 datasets). Similar situation is found for the Enhanced feature set, with only 2 out of the 5 **added** feature types selected for at least one dataset. This raises a controversial question that is whether the practice of feature engineering found to be fundamental to classic methods is really worthwhile. As it appears that with generic features such as word and PoS n-grams combined with feature selection, the systems can even outperform using a sophisticated sets of unselected features.

5 Conclusion

This work studied the effect of feature selection on the task of hate speech detection from Twitter. We have shown feature selection to be a very powerful technique as it is able to select a very small set of the most predictive features that are often generic and widely used in many other language-related tasks, to achieve much better results than models using carefully engineered features. In future, we will analyse the effect of feature selection in other tasks.

References

1. Burnap, P., Williams, M.L.: Cyber hate speech on Twitter: an application of machine classification and statistical modeling for policy and decision making. Policy Internet **7**(2), 223–242 (2015)
2. Davidson, T., Warmsley, D., Macy, M., Weber, I.: Automated hate speech detection and the problem of offensive language. In: Proceedings of ICWSM 2017 (2017)
3. Gambäck, B., Sikdar, U.K.: Using convolutional neural networks to classify hate-speech. In: Proceedings of the Workshop on Abusive Language Online, pp. 85–90 (2017)
4. Park, J.H., Fung, P.: One-step and two-step classification for abusive language detection on Twitter. In: ALW1: 1st Workshop on Abusive Language Online (2017)
5. Schmidt, A., Wiegand, M.: A survey on hate speech detection using natural language processing. In: Proceedings of the Workshop on Natural Language Processing for Social Media, pp. 1–10. Association for Computational Linguistics (2017)
6. Waseem, Z.: Are you a racist or am i seeing things? Annotator influence on hate speech detection on Twitter. In: Proceedings of the Workshop on NLP and Computational Social Science, pp. 138–142. Association for Computational Linguistics (2016)
7. Waseem, Z., Hovy, D.: Hateful symbols or hateful people? Predictive features for hate speech detection on Twitter. In: Proceedings of the NAACL Student Research Workshop, pp. 88–93. Association for Computational Linguistics (2016)
8. Zhang, Z., Robinson, D., Tepper, J.: Detecting hate speech on Twitter using a convolution-GRU based deep neural network. In: Proceedings of ESWC 2018 (2018)