Proceedings of the 15th Conference on Natural Language Processing (KONVENS 2019)

FoSIL - Offensive language classification of German tweets combining SVMs and deep learning techniques

Florian Schmid, Justine Thielemann, Anna Mantwill, Jian Xi, Dirk Labudde, Michael Spranger
University of Applied Sciences Mittweida
Technikumplatz 17
09648 Mittweida
spranger@hs-mittweida.de

Abstract

In this paper an approach for the automatic detection of offensive language in German twitter posts, so called tweets, based on a data set provided by the organizers from the GermEval2019 contest is presented. Two different approaches were used. The first one is based on a document-term-matrix and the second one uses fastText to represent tweets as numerical vectors. Additionally, some text based features, e.g. sentiment analysis of the text and emojis were added. Further, some statistic features were calculated, e.g. the number of special characters, hashtags and mentions. As a classifier a support vector machine with radial kernel function was utilized. The best f1-macro values for subtask 1 of 0.7978, subtask 2 of 0.5957 and for subtask 3 of 0.7055, validated by a ten-fold cross validation, were achieved by using a self-trained unsupervised fastText model to vectorize the tweets.

1 Introduction

Social media platforms like Twitter have become increasingly popular in the past ten years (Twitter, 2019). People of nearly all generations, especially teenagers and young adults, are using them to communicate with friends, connect with people around the world or to state their opinion about current topics (Faktenkontor, 2019). Unfortunately, the increasing number of people using social media platforms results in a growth of posts with offensive content. Therefore, the automatic detection of offensive language on these platforms is a very important task to effectively fight e.g. hate speech, hateful or insulting comments, cyber mobbing or cyber bullying.

The detection of offensive language is a typical task in sentiment analysis, which is in turn a subtask in text classification, that focuses on the contextual mining of texts related to some specific objects. Furthermore, sentiment analysis is especially useful to find out the public opinion concerning highly sensitive political topics, as was shown in the study by Backfried et al. (2016), in which Twitter texts were analysed in order to detect tendencies that are inter-related to real world events in the European refugee crisis. Usually, sentiment analysis involves methods from different disciplines such as natural language processing and machine learning (Pang et al., 2002).

The challenge by the organizers of the GermEval 2019 Task 2 focuses on detecting offensive language in tweets and is subdivided into three smaller tasks: The first task is to detect texts containing offensive language in Twitter messages. The second task is the fine-grained categorization of tweets into one of the categories neutral, profanity, insult or abuse. Finally, the third task is to distinguish offensive tweets to be explicit or implicit.

In this paper, for the first subtask two systems were used and compared. The first system uses SVM with a radial kernel function as classifier incorporating different lexical resources. This approach forms the baseline. The second system extends the first one by vectorizing the data with a self-trained fastText model based on nearly 30 million tweets. Due to better results being achieved with the second system, it was used for the other two subtasks.

The paper is organized as follows: in Section 2 an overview of the data is given. In Section 3 the methods used are described and in Section 5 the results are presented. Finally, in Section 6 a short conclusion is given.

2 Data

The data for all subtasks consisted of tweets provided by the organizers of the GermEval 2019 Task 2. For the first two subtasks the dataset contained 12,536 manually labelled tweets. As can be seen in Table 1, the dataset was highly imbalanced. There is double the amount of tweets in the category OTHER compared to the category OFFENSE and even for the fine-grained classification task the number of tweets in each category varies greatly.

For the third subtask an additional dataset was provided, consisting of only 1958 tweets. Again, the dataset was imbalanced (see Table 1), with the category EXPLICIT having more than five times as many tweets as the category IMPLICIT. The tweet's content was neither preprocessed nor cleaned and therefore contained hashtags, user mentions, emoticons and other text patterns that are typical for social media platforms (GermEval, 2019).

subtask	category	# tweets
Subtask 1	OFFENSE	4177
	OTHER	8359
Subtask 2	ABUSE	2305
	INSULT	1601
	PROFANITY	271
	OTHER	8359
Subtask 3	EXPLICIT	1699
	IMPLICIT	259

Table 1: Number of tweets in each category.

3 Methods

In this paper, two different systems are presented for the classification, each based on a SVM with a radial kernel function and the preprocessed tweets as described in the following Section 3.1. For the first system a document-term-matrix (DTM) built on a pruned vocabulary was used that holds the following condition: $1 \leq tf(w) \leq 50$, where $tf(w)$ is the term frequency of each single word from the preprocessed tweets. Further statistical features, sentiment scores and lexical resources were used as additional features. In contrast, in the second system the preprocessed tweets were vectorized using a self-trained unsupervised fastText model.

Some more detailed information is given in the following subsections.

3.1 Preprocessing

Before any further steps were taken to normalize the tweets some statistical features were calculated. An overview is given in Table 2. Afterwards, the tweets were changed to lower case, all special German characters were converted, the punctuation marks removed and the words lemmatized using TreeTagger (Schmid, 1995).

Feature	values
tweets containing emojis	1011
tweets containing hashtags	2355
tweets containing mentions	12,536
average no. of words per tweet	18.22
average no. of punct. marks per tweet	6.44

Table 2: Statistical features for both datasets.

Because hashtags are potentially important to capture the real message or sentiment of a tweet, only the #-sign at the beginning of a hashtag was removed, yet the hashtag itself was kept as part of the tweet.

As no further information about users or groups was given, the mentions in all tweets were removed completely. Moreover, stop words were removed using the list provided by Diaz (2016). However, this list was modified, because some stop words may give important information regarding the sentiment of a tweet. For instance, it makes a huge difference whether an adjective is preceded by a negation word or not. Furthermore, personal or possessive pronouns may indicate that someone is addressed personally. Consequently, negation words as well as personal and possessive pronouns were not removed. Finally, a document-term-matrix was created.

3.2 Feature Modelling

Sentiment Analysis on Texts and Emojis

To get the sentiment of a tweet, a combined score was calculated from the words and emojis in the tweet. In order to get a sentiment score for the words SentiWS (Remus et al., 2010) was used to assign a positive or negative polarity value between -1 and 1 to each word. Emojis were taken into account, because, usually, a large number of tweets contain emojis (Gotzner, 2013) and because, in some cases they can indicate the mood or clarify the meaning of an expression. In order to calculate a sentiment score for the them, the Emoji Sentiment Ranking (Kralj Novak et al., 2015) from the Department of Knowledge in Slovenia was used. First, the emojis were extracted, converted to their unicode sequence (e.g. <U+263>) and then a score between -1 and 1 was assigned.

Finally, the single scores were summed up for each tweet.

Lexical Lookup

Due to the young age of twitter users, colloquial words or teenage-slang-words are often included in tweets. Several teenage-slang-words are offending either a single individual or groups of them. To detect these words a lexicon of youth language was used, which was created by Helmut Hehl (Hehl, 2006). However, some phrases were removed manually because they were not relevant for detecting offensive language.

Additionally, in order to detect swearwords in tweets a comprehensive lexicon containing offensive nouns, adjectives and also verbs was created. The nouns were obtained from the "HyperHero Schimpfwortliste", a huge list with 11,300 swearwords (HyperHero, nd). The adjectives with an abusive connotation were manually extracted from the website `www.wortwuchs.net` (Willing and Goldschläger, nd). The verbs were added manually because there was no suitable list available. Finally, some words, which were significant for the data were added manually in their lemmatized form and all lists were combined to our comprehensive lexicon. For each tweet, a binary decision was made whether the tweet contains offensive or slang words from our lexicons or not.

Vectorization

The language used in social media is strongly related to currently discussed topics. Therefore, each sample drawn from social media captures only a limited amount of the vocabulary used. To overcome this limitation, a huge amount of tweets were collected to capture as much of the vocabulary as possible incorporating different topics in order to build a fastText model (Joulin et al., 2017).

FastText is able to capture the context of words instead of simply checking if a word is in a tweet or not. In order to train a suitable fastText model a crawler was set up that automatically collects German tweets from the twitter API. This way, an additional dataset was created consisting of finally 29.6 million unique German tweets each preprocessed as described in Subsection 3.1. This slowly growing corpus formed the basis for training different unsupervised fastText models to subsequently create a vectorized text representation of the data provided by the organizers. At different points in time models were created in order to analyse the performance depending on the number of tweets used for the fastText model. As can be seen in Figure 1 with a growing number of tweets the results also increase until the number of tweets reaches 24.9 million. Afterwards, the macro F1 measure slightly decreases. For the two submitted runs the decision was made to use the fastText model that achieved the best result in the ten-fold cross-validation with 24.9 million unique German tweets and the fastText model with the final amount of 29.6 million tweets.

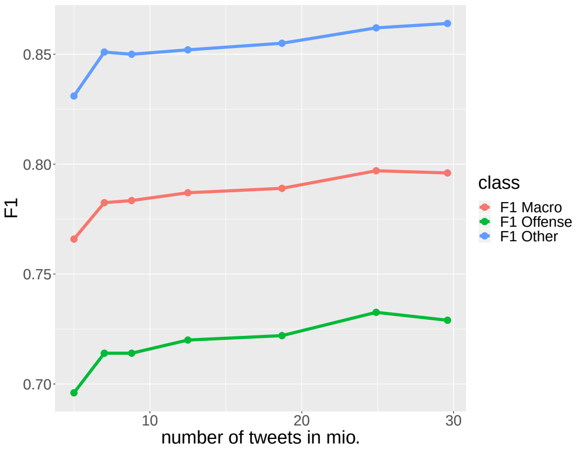

Figure 1: The plot shows the overall performance of system 2 with increasing numbers of tweets used to train the fastText model.

As parameters 50 epochs, 300 dimensions, a window size of 5, char N-grams with a length from 2 to 6 and a learning rate of 0.05 were used. The usage of char N-grams make the model more robust against unseen words. The models were calculated using a continuous-bag-of-words and skipgram technique as well as a hierarchical softmax function and negative sampling.

4 System Descriptions

In this paper, two different systems were used for the classification. In the following the different systems are described.

System 1 - DTM and SVM (radial kernel)

The first system is based on the preprocessed tweets and a document-term-matrix (DTM) which was built with the pruned vocabulary from the training data set (min tf = 1 , max tf = 50). Additionally, some statistical features, sentiment scores and lexical resources were added. As a classifier a support vector machine with a radial kernel function was used.

System 2 - fastText and SVM (radial kernel)

The second system is based on the preprocessed tweets which were vectorized by a self trained unsupervised fastText model. As for the first system, statistical features, sentiment scores and lexical resources were added. Again, a support vector machine with a radial kernel function was used. For the 1st run a fastText model built on 24.9 million unique tweets was used, whereas for the 2nd run a fastText model built on 29.6 million unique tweets was used.

5 Results

The following tables show the results for each subtask and system. All results are based on the training data set and a ten-fold cross-validation to prevent overfitting of our models.

The results in Table 3 show the best achieved scores with system 1, which represent the start of development. This system formed the baseline for the further work and results were not submitted to the contest.

Run	Category	P	R	F1
-	OFF.	0.5640	0.4096	0.4742
	OTHER	0.7405	0.8415	0.7877
	Mac. avg.	**0.6522**	**0.6255**	**0.6386**

Table 3: Results for subtask 1 with system 1 (not submitted).

The following Tables 4 to 6 show the best scores achieved with system 2 and the different runs as described in Section 4. Both runs were submitted to the contest.

Run	Category	P	R	F1
1st	OFF.	0.7193	0.7467	0.7326
	OTHER	0.8710	0.8543	0.8625
	Mac. avg.	**0.7952**	**0.8005**	**0.7978**
2nd	OFF.	0.7291	0.7302	0.7291
	OTHER	0.8650	0.8637	0.8643
	Mac. avg.	**0.7967**	**0.9770**	**0.7968**

Table 4: Results for subtask 1 with system 2.

With the first model, several problems occurred. The large vocabulary of more than 22,000 unique words led to a high sparsity of the DTM, which in turn caused different computational problems. Therefore, the vocabulary was pruned, as described in the former section, in order to reduce its size to around 1,100 words. However, pruning the vocabulary also means that a lot of information from the tweets gets lost. As the results in Table 3 clearly show, with the first system it was not possible to detect much of the offensive language in the dataset. As can be seen in Table 4 the results for subtask one clearly improved using the fastText model. As might be expected, the results are worse for the second subtask (see Table 5). The results clearly show that it is really difficult to detect profanity in the tweets, whereas for the category ABUSE the best results were achieved. However, the results also coincide with the number of tweets available. For PROFANITY the number of tweets was the lowest, while for ABUSE it was much higher. Furthermore, the results in Table 6 indicate that it is more difficult to detect implicit abusive language in comparison to explicit abusive language. Yet again, the bad results can be partly explained with the available number of tweets. Interestingly, a greater number of tweets for the training of the fastText

Run	Category	P	R	F1
1st	ABUSE	0.5567	0.6130	0.5822
	INSULT	0.4875	0.4866	0.4865
	PROF.	0.6669	0.2839	0.3950
	OTHER	0.8602	0.8526	0.8563
	Mac. avg.	**0.6428**	**0.5586**	**0.5975**
2nd	ABUSE	0.5372	0.6239	0.5769
	INSULT	0.4629	0.5391	0.4978
	PROF.	0.5851	0.3134	0.4033
	OTHER	0.8739	0.8200	0.8460
	Mac. avg.	**0.6148**	**0.5741**	**0.5935**

Table 5: Results for subtask 2 with system 2.

Run	Category	P	R	F1
1st	IMPLIC.	0.3582	0.6678	0.4653
	EXPLIC.	0.9418	0.8164	0.8744
	Mac. avg.	**0.6500**	**0.7421**	**0.6929**
2nd	IMPLIC.	0.3660	0.7143	0.4825
	EXPLIC.	0.9489	0.8081	0.8725
	Mac. avg.	**0.6575**	**0.7612**	**0.7055**

Table 6: Results for subtask 3 with system 2.

model does not improve the results for the first two subtasks. However, the difference between the two runs is minimal for all three subtasks.

6 Conclusion

As pointed out in the discussion section, it can be clearly seen how modern techniques for word representations like fastText can help achieve better results in natural language processing tasks. Using a radial SVM and a fastText vectorization as a feature, for the first subtask an F1-measure of 0.7978 was achieved, whereas for the second and third subtask the F1-measure was 0.5975 and 0.7055, respectively.

The deep learning technology used for fastText enables the transformation of most of the context into numerical vectors with a moderate number of dimensions. This led to an increase of the overall performance of our model in the second system. Besides fastText there are many different implementations for modern word embeddings like word2vec, sent2vec or doc2vec. It might be interesting to use different word embedding techniques for the text vectorization as well as classifier chains.

References

Gerhard Backfried and Gayane Shalunts. 2016. Sentiment analysis of media in german on the refugee crisis in europe. In Paloma Díaz, Narjès Bellamine Ben Saoud, Julie Dugdale, and Chihab Hanachi, editors, *Information Systems for Crisis Response and Management in Mediterranean Countries*, pages 234–241, Cham. Springer International Publishing.

Gene Diaz. 2016. Collection of Stopwords for multiple languages. `https://github.com/stopwords-iso/stopwords-iso`. Accessed: 27.03.2019.

Faktenkontor. 2019. Anteil der befragten Internetnutzer, die Twitter nutzen, nach Altersgruppen in Deutschland im Jahr 2017. Statista. `https://de.statista.com/statistik/daten/studie/691593/umfrage/anteil-der-nutzer-von-twitter-nach-alter-in-deutschland/`. Accessed: 24.07.2019.

GermEval. 2019. Germeval Task 2, 2019 — Shared Task on the Identification of Offensive Language. `https://projects.fzai.h-da.de/iggsa/germeval/`. Accessed: 01.07.2019.

Peter Gotzner. 2013. Herzchenzähler analysiert das Twitter-Gefühlsleben. Accessed: 24.07.2019.

Helmut Hehl. 2006. Lexikon der Jugendsprache. Accessed: 14.06.2019.

HyperHero. n.d. HyperHero Schimpfwortliste. `http://www.hyperhero.com/de/insults.htm`. Accessed: 03.04.2019.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, volume 2, pages 427–431.

Petra Kralj Novak, Jasmina Smailović, Borut Sluban, and Igor Mozetič. 2015. Sentiment of Emojis. *PLOS ONE*, 10(12):1–22, 12.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of EMNLP*, pages 79–86.

Robert Remus, Uwe Quasthoff, and Gerhard Heyer. 2010. SentiWS - a publicly available German-language resource for sentiment analysis. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May. European Languages Resources Association (ELRA).

Helmut Schmid. 1995. Improvements In Part-of-Speech Tagging With an Application To German. In *In Proceedings of the ACL SIGDAT-Workshop*, pages 47–50.

Twitter. 2019. Anzahl der monatlich aktiven Nutzer von Twitter weltweit vom 1. Quartal 2010 bis zum 1. Quartal 2019 (in Millionen) [Graph]. Statista. `https://de.statista.com/statistik/daten/studie/232401/umfrage/monatlich-aktive-nutzer-von-twitter-weltweit-zeitreihe/`. Accessed: 24.07.2019.

Rebekka Willing and Jonas Goldschläger. n.d. Wortwuchs Adjektivliste. `https://wortwuchs.net/adjektivliste/`. Accessed: 14.06.2019.