Investigating Multilingual Abusive Language Detection: A Cautionary Tale

Kenneth Steimel, Daniel Dakota, Yue Chen, Sandra Kübler
Indiana University
{ksteimel,ddakota,yc59,skuebler}@indiana.edu

Abstract

Abusive language detection has received much attention in the last years, and recent approaches perform the task in a number of different languages. We investigate which factors have an effect on multilingual settings, focusing on the compatibility of data and annotations. In the current paper, we focus on English and German. Our findings show large differences in performance between the two languages.

We find that the best performance is achieved by different classification algorithms. Sampling to address class imbalance issues is detrimental for German and beneficial for English. The only similarity that we find is that neither data set shows clear topics when we compare the results of topic modeling to the gold standard. Based on our findings, we can conclude that a multilingual optimization of classifiers is not possible even in settings where comparable data sets are used.

1 Introduction

The last decade has seen a massive increase in user generated content in social media. While most people are interested in connecting with family and friends and in exchanging experiences, there is an increasing number of posts that cross the line from sharing negative opinions to becoming abusive. Since the data is too massive for manual filtering, automated methods to detect abusive language reliably are required. This has created a novel research area under the titles of abusive language detection, hate speech detection, flame or cyberbullying detection.

While most of the work on abusive language detection has focused on English (Schmidt and Wiegand, 2017; Park and Fung, 2017; Lee et al., 2018), there is some work on other languages, and first attempts have also been made to develop methods that work across different languages (Fehn Unsvåg and Gambäck, 2018).

Our interest also focuses on multilingual abusive language detection. However, before we engage in a full scale investigation of which methods work well across multiple languages, we need to know more about which factors have an effect on multilingual settings, including but not restricted to the compatibility of data and annotations, differences between languages, and topic effects. In the current paper, we focus on two languages, English and German, where we have access to similar data (see section 3 for more information). We approach the following questions as we investigate an approach across the two languages:

1. Do classifiers behave similarly across the two languages? I.e., can we establish the best classifier on one language and then use it successfully for the second language?

2. Can we determine which types of features are necessary for a classifier? Are the types of features and the number of features comparable across the two languages?

3. The data sets are skewed towards non-abusive language, and research in sentiment analysis has shown that over-sampling methods can improve results (Liu et al., 2014). Thus, do over-sampling methods show consistent results across both languages?

4. For tasks related to sentiment analysis, it is often the case that a classifier learns topic information rather that sentiment (Pang et al., 2002). We investigate whether the two languages show similar effects with regard to topics.

Proceedings of Recent Advances in Natural Language Processing, pages 1151–1160,
Varna, Bulgaria, Sep 2–4, 2019.
https://doi.org/10.26615/978-954-452-056-4_132

Our results show that the data sets differ in their answers to questions 1–3, only showing similarities with regard to topics, leading us to the the preliminary conclusion that we cannot transfer methodology across languages and data sets when the data sets for the individual languages have been collected opportunistically. Since it is highly unlikely that we can completely replicate the data collection methods from the "source" language, the implications of our findings are far reaching and necessitate further investigation into the issues of multilingual abusive language detection.

The remainder of the paper is structured as follows: We discuss related work in section 2 and the data sets in section 3. Then, we explain our experimental setup and feature sets in section 4. Section 5 presents the results, and section 6 draws conclusions and discusses future work.

2 Related Work

Research on abusive language detection has recently drawn much attention, as several recent shared tasks (Basile et al., 2019; Kumar et al., 2018; Wiegand et al., 2018; Zampieri et al., 2019) demonstrate. So far, research has mostly focused on English. For a comprehensive survey of NLP techniques to detect hate speech see (Schmidt and Wiegand, 2017; Fortuna and Nunes, 2018). Here, we will focus on issues relating to problems in multilingual abusive language detection.

There is work on abusive language detection in Arabic, Dutch, and German. For Arabic, Mubarak et al. (2017) developed a data set of abusive language by creating an abusive word list and then splitting their Twitter dump into abusive and non-abusive classes by user, where users are considered abusive if they have used at least one word from the list. They also created an algorithm to extend the list of abusive words.

Cyberbullying detection in Dutch social media was performed by Van Hee et al. (2015). They collected data from Ask.fm and annotated it with 7 categories including 'Threat', 'Insult', and 'Defamation'. For their classifier, they used a bag-of-words approach for word and character n-grams and a sentiment lexicon, with an SVM classifier. They found that sentiment features alone yield poor results, as does a substantial reduction in the number of bag-of-word features due to data sparsity.

The 2018 Germeval shared task focused exclusively on detecting abusive language in German tweets (Wiegand et al., 2018) (for a description of the data set, see section 3). While many of the best systems used neural architectures, the winning system (Montani and Schüller, 2018) used an ensemble method with classifiers trained on a subset of features (such as TF-IDF and character n-grams). The predictions were then combined with a maximum entropy model for a final prediction. They found that strategies such as feature selection, sampling, and extensive preprocessing ultimately reduced performance in their ensemble.

While most work focuses on identifying abusive language in a specific language, Fehn Unsvåg and Gambäck (2018) examined how a single approach can handle abusive language across multiple languages: English, Portuguese, and German. They used a number of existing twitter corpora including the corpus by Waseem and Hovy (2016) for English[1], the one by Ross et al. (2016) for German, and the one by Fortuna (2017) for Portuguese. They also incorporated "user features" (i.e., specific demographic information known about the author of the tweet) along with a standard set of word and character n-gram features using logistic regression. They noted slight improvements but only specific user features contributed improvements to a given language (e.g., network features boosting English and Portuguese).

Several systems for detecting abusive language in Hindi and English were developed as part of the 2018 *Trolling, Aggression and Cyberbullying* shared task (Kumar et al., 2018). This shared task used data from Facebook and Twitter. The systems had to label examples as 'Not Aggressive', 'Covertly Aggressive', and 'Overtly Aggressive'. Modha et al. (2018) achieved the best results on the Hindi side of the shared task using a convolutional neural network with fastText embeddings (Mikolov et al., 2018). Galery et al. (2018) identified abusive language in the English portion of the corpus as their initial survey found code-switching between Hindi and English. To address this code-switching, they used fastText embeddings for both languages and a SVD transformation of these two sets of embeddings to generate multilingual embeddings using sub-word units. These multilingual embeddings were used in a GRU-based recurrent neural network. They found that this approach was not effective on their particular data set due to

[1]This is the same English corpus used in the present work.

English	German
These girls are the equivalent of the irritating Asian girls a couple years ago. Well "done," 7 #MKR	@tagesschau Euere AfD Hetze wirkt. Da könnt ihr stolz sein bei #ARD-Fernsehen (Eng.: @tagesschau Your AfD baiting works. You can be so proud at #ARD-TV)
@ameedjadallah I read the entire Quran. Islam is what is wrong.	@welt Bla bla bla! Lügenpresse verkauft uns mal wider für Dumm! Alles gespieltes Theater!! (Eng.: @welt Blah blah blah! Lying press is taking us for fools again. It's all totally staged!!)

Table 1: Sample abusive tweets for English and German

	Abusive	Non-Abusive
English		
Train	4 460	9 683
Test	496	1 076
Total	4 956	10 759
Total in %	31	69
German		
Train	1 517	2 992
Test	171	329
Total	1 688	3 321
Total in %	34	66

Table 2: Distribution of binary class labels in the English and German data sets

the limited instances of code-switching present.

3 Data Sets

For our work, we chose two data sets that were as similar as possible without creating a new, tightly controlled bilingual data set. For English, we used the publicly available Twitter hate speech data set created by Waseem and Hovy (2016). The original corpus included approximately 16 000 tweets; however we were only able to retrieve 15 715 tweets using the Twitter API, the rest were unavailable or deleted. The English data set was manually annotated with three different labels: 'racism', 'sexism' and 'none', where none refers to non-hate speech. For more information regarding the annotation guidelines, see (Waseem and Hovy, 2016). For classification, we split the data set using 90% (14 143 tweets) as training data and 10% (1 572 tweets) as test set. Every tenth tweet was assigned to the test set and removed from the training set to ensure the data sets were drawn from the whole corpus.

For the German data, we used the 2018 GermEval shared task data set (Wiegand et al., 2018) with the annotations for task 1 of the shared task. The annotations are binary: examples are either labeled as 'offensive' or 'other'. For our work, we use the training set of the shared task, which consists of 5 009 samples, and split it into our training and test sets, taking every tenth example for the test set. We do not use the official GermEval test set since we basically perform optimization experiments, and any scores obtained on the test data should not be directly compared to other research.

Since the English data set consists of a ternary classification while the German set uses a binary classification, we simplify the English data in order to make the data sets consistent across the two languages: We group 'racism' and 'sexism' into an 'abusive' group while the 'none' labels are maintained. Examples of the abusive class are shown in table 1, and a summary of the class distributions in the two data sets is shown in table 2.

The imbalance between classes is clear given the data in table 2. The non-abusive data outnumber the abusive data in a ratio of about 2:1 for both languages. This will negatively affect the performance of the classifiers since statistical classifiers tend to predict the majority class.

4 Methodology

We have developed two pipelines for detecting abusive language in tweets: One pipeline is trained on German data and the other is trained on English data.

Classifiers. For the first set of experiments, we use a range of classifiers including random forest, SVM, XGBoost and a neural network approach. For the former three classifiers, we use scikit-learn (Pedregosa et al., 2011), for the neural network architecture tensorflow's Keras API (Abadi et al., 2016). Based on previous research on related tasks (Park and Fung, 2017; Badjatiya et al., 2017), we experiment with several promising architectures, including fully connected neural networks and convolutional neural networks, along with different word embeddings, with both BERT

and Flair embeddings (Devlin et al., 2018; Akbik et al., 2018), stand-alone and stacked respectively.

For the scikit-learn classifiers, we optimized hyper-parameters using grid search. For the neural networks, dropout and batch normalization techniques are applied, and architecture selection and hyper-parameter optimization are done in a non-exhaustive search.

Features. We use simple character n-grams along with stemmed word n-grams and dependency parse-derived features.

Stemming. Since we were unable to identify a good lemmatizer for German Twitter data, we decided to implement a stemmer for both the English and German data, in order to maintain compatibility between the two languages. The YASS stemmer (Majumder et al., 2007) is an unsupervised stemming algorithm that generates a minimum spanning tree of words based upon different string similarity distance metrics, cuts the hierarchy, and then stems a word by replacing the word with the centroid of the cluster the word belongs to. We use the YASS stemming method with a minor modification: While the YASS stemmer replaces all words that belong to a cluster with the cluster centroid, we replace all cluster members with the shortest member of the cluster. Stems are shorter than their morphologically related forms since affixes are not present. However, this step is not a departure from the YASS algorithm, instead it is an adaptation to increase the effectiveness of this algorithm on our particular domain. In addition, numbers, twitter handles, and URLs are removed from the data prior to stemming.

The distance metrics used by Majumder et al. (2007) rely heavily on the suffixing nature of German and English inflectional morphology. While these distance metrics fall flat for non-suffix based inflectional morphology like irregular past tense (primarily ablaut grades in words like 'sleep', 'slept'), they produce less spurious stems compared to other common metrics like the Levenshtein distance. We use distance metric 4 from the YASS stemmer.

Dependency parsing features. To extract dependency features for English, we use the Tweebo parser (Kong et al., 2014), which is designed to parse Twitter data and requires minimal preprocessing to obtain useful parses. Unlike English, German does not possess a Twitter domain specific parser. For this reason, we use the Mate parser pipeline (Björkelund et al., 2010). However, in order to maximize the usefulness of Mate, which expects a more standard text structure, we preprocess the German Twitter data.

Preprocessing steps for parsing include: removing one or more hashtags or retweets after punctuation at the end of tweet as well as removing initial hashtags and retweets, removing the # sign from any hashtag in the middle of the tweet, removing all emojis, and detaching punctuation from words. We also use a base list of abbreviations[2] and add additional ones to ensure that these are kept during the tokenization process.

We extract dependency triples consisting of (dependent, head, label) that occurred a minimum of five times as features. These features are Boolean valued, denoting their presence or absence in a tweet.

Sampling methods. Given the imbalanced nature of the data sets, we investigate sampling techniques to examine whether sampling can yield better performance. We use imbalanced-learn (Lemaître et al., 2017) to perform both under-sampling and over-sampling techniques. More specifically, we use four over-sampling and two under-sampling methods: SMOTE (Chawla et al., 2002) constructs synthetic examples of the minority class by averaging over two randomly chosen minority examples. Borderline SMOTE (Han et al., 2005) focuses on the area around the decision boundary to create new examples, SVM SMOTE (Nguyen et al., 2011) uses an SVM to determine the examples to average while ADASYN (He et al., 2008) concentrates on the areas of the minority class search space that are difficult. Edited Nearest Neighbors (ENN; Wilson, 1972) uses a k-NN approach to identify examples that are untypical for their neighborhood; these are then deleted. One sided selection (Kubat and Matwin, 1997) uses Tomek links to identify and delete noisy examples.

Topic modeling. We train a topic modeler on the two data sets, i.e., we create an LDA (Blei et al., 2003) topic modeler per language and use it to group tweets into two topics. We have preprocessed the tweets in the same manner that was

[2]Taken from Stefanie Dipper's Perl German Tokenizer and found at https://www.linguistics.ruhr-uni-bochum.de/~dipper/pub/software/abbrev.lex

Classifier	Prec	Rec	F-Score
majority class	34.22	50.00	40.63
RF	80.67	74.17	76.08
XGBoost	83.46	78.80	80.49
SVM	82.11	66.58	68.20
NN	34.22	50.00	40.63

Table 3: English results for a range of classifiers.

Classifier	Prec	Rec	F-Score
majority class	32.90	50.00	39.69
RF	66.00	66.50	66.50
XGBoost	68.50	60.00	59.50
SVM	74.41	70.97	72.01
NN	32.90	50.00	39.69

Table 4: German results for a range of classifiers

used for dependency parsing. We then use the top 100 words by frequency to produce two topics.

Evaluation. We report precision, recall, and F_1. Because of the skewing in the data set, all these measures are calculated as the macro average of precision, recall, and F_1, i.e., we calculate the measures per class and then average across both classes.

5 Results

5.1 Classifier Behavior across Languages

Here we investigate whether the classifiers show the same trends across both languages. This will allow us to decide whether we can select the classifier for one language and then keep the selection stable across further languages. For these experiments, we restrict the feature set to using only character n-grams of length 2-7 with a minimum frequency of 3, resulting in 137 434 features for German and 282 507 for English.

The results of the experiments for English are shown in table 3, the results for German in table 4. These results show that classifying all tweets as non-abusive, the majority class, results in a macro-F of around 40% for both languages. All classifiers but the neural networks perform better. All the neural network architecture/embedding combinations predict the majority class throughout.

For the other classifiers, we reach higher F-scores for English than for German: 80.49. vs. 72.01. This is not unexpected, since the English data set is larger than the German one. However, the best English results are reached by XGBoost while the same classifier only reaches 59.50 F for German. The highest results for German are reached by the SVM, which reaches a lower F-score on English (68.20). It is also interesting to see that for English, XGBoost reaches a recall that is more than 12 points higher than the SVM. In contrast, for German, XGBoost's recall is the lowest of all classifiers (except for the neural networks). For the SVM, recall is higher for German than for English, thus going against the general trend, but the difference is much less pronounced (> 4 points).

From these results, we draw the conclusion that we cannot choose a classifier for a new language based on experience with another language.

5.2 Feature Selection across Languages

The next set of experiments is concerned with the question of whether we can use the same features across languages, or if each language requires its own set of informative features. For these experiments, we decided to focus on SVMs since they show good performance and similar trends across the languages in the comparison of classifiers above and since they train much faster than XGBoost. We add two additional feature types into the vectors: stems and dependency features. For German, this results in a total number of features of 148 322 and for English, in 308 323.

We use Information Gain (IG) for feature selection and perform experiments using the set of features with the highest IG in incremental cutoffs. This results in a different amount of features for English and German, but this difference can be explained by the differences in the morphological complexity of the two languages and the data sizes. Results for English are reported in table 5, showing the overall results across both classes and specifically for the abusive class. Results for German are reported in table 6. The last row in each table repeats the results from the previous section, i.e., using all character n-gram features.

For English, a comparison of the two experiments with the character n-grams only as opposed to the full feature set including stems and dependency features shows that adding these features has a minimal negative effect, lowering the F-score from 68.20 to 67.70.

For the experiments on English using feature selection, we see that results with even 2 660 features improve significantly over the all features

		Overall			Abusive		
IG threshold	Num. IG features	Prec	Rec	F	Prec	Rec	F
0.000075	2660	79.38	72.62	74.49	77.95	52.02	62.39
0.00005	4232	80.29	0.74	0.76	78.95	54.44	64.44
0.000025	9305	80.72	74.27	76.18	79.59	55.04	65.08
0.00001	24350	82.26	76.48	**78.36**	81.21	59.27	**68.53**
0.0000075	33187	82.64	75.99	78.04	82.42	57.66	67.85
0.000005	60000	83.06	75.10	77.33	84.00	55.04	66.50
–	all features	81.87	66.18	67.70	87.31	34.68	49.64
–	only char n-grams	82.11	66.58	68.20	87.56	35.48	50.50

Table 5: English results with IG feature selection, overall and for the abusive class.

		Overall			Abusive		
IG threshold	Num. IG features	Prec	Rec	F	Prec	Rec	F
0.005	266	66.18	58.76	58.02	61.97	25.73	36.36
0.004	451	65.54	60.88	61.10	59.18	33.92	43.12
0.003	788	67.59	62.79	63.30	62.14	37.43	46.72
0.002	2 338	66.19	62.74	63.27	59.13	39.77	47.55
0.0016	4 071	67.42	65.19	65.80	59.70	46.78	52.46
0.0014	6 404	68.70	66.23	66.92	61.65	47.95	53.95
0.0011	9 690	69.68	67.40	**68.10**	62.77	50.29	**55.84**
0.0008	16 791	70.21	65.71	66.56	65.49	43.27	52.11
0.0006	26 801	69.62	66.26	67.06	63.71	46.20	53.56
0.0004	48 014	72.84	68.93	**69.95**	68.55	49.71	**57.63**
0.0002	69 541	75.21	72.28	**73.26**	70.80	56.73	**62.99**
0.0001	101 605	74.92	72.13	73.07	70.29	56.73	62.78
–	all features	74.71	71.13	72.20	70.77	53.80	61.13
–	only char n-grams	74.41	70.97	72.01	70.23	53.80	60.93

Table 6: German results with IG feature selection, overall and for the abusive class.

baseline, and they continue their upward trend until around 14 000 features. At this point, however, a downward trend begins, suggesting that for English, a lower number of important features for the SVM classifier is beneficial. The results for the minority class follow the same trend: They also reach their peak at around 14 000 features. This means that the classifier's performance on the minority class is the driving factor (which is partly a result of using the macro-averaged values).

For German, the addition of stems and dependencies has a minimal positive effect, increasing the F-score from 70.01 to 72.20. For the experiments on feature selection, we see a steady upward trend as the number of features increases until we reach around 70 000 features, at which point results start to decrease. Again, the results on the abusive class follow the same trend. This means that, at a certain point, we reach a saturation of features in terms of modeling the abusive class. If we add more features, the classifier suffers from irrelevant features.

When we compare the results across the two languages, we notice a discrepancy in that the stems and dependencies help for German while they are harmful for English. Both effects are minimal. One possible explanation may be that for English, the additional features only increase data sparsity without providing novel information. For German, which is morphologically richer and has freer word order. Introducing the stems and dependencies may help alleviate data sparsity to a certain degree. This requires further investigation.

The second discrepancy concerns the optimal number of features. For German, we achieve our best results using slightly less than half the features; for English, the best results are based on approximately 4.5% of the data. This means that the results differ in terms of absolute numbers and percentage.

Sampling method	Abusive Precision	Abusive Recall	Non-Abusive Precision	Non-Abusive Recall	F-score
No sampling	85.49	43.95	78.89	**96.56**	72.45
SMOTE	63.21	**76.21**	**87.89**	79.55	76.31
Borderline SMOTE	62.23	73.99	86.92	79.65	75.48
SVM SMOTE	62.46	73.79	86.82	79.55	75.34
ADASYN	61.19	74.40	86.89	78.25	74.75
Edit nearest neighbors	81.77	57.86	82.88	94.05	**77.94**
One sided selection	**85.60**	44.35	79.01	**96.56**	72.67

Table 7: Results for English using sampling (70 000 features).

Sampling method	Abusive Precision	Abusive Recall	Non-Abusive Precision	Non-Abusive Recall	F-score
No sampling	**70.80**	56.73	79.61	**87.84**	**73.26**
SMOTE	58.17	52.05	76.37	80.55	66.67
Borderline SMOTE	60.26	54.97	77.62	81.16	68.42
SVM SMOTE	60.26	54.97	77.62	81.16	68.42
ADASYN	57.32	52.63	76.38	79.64	66.43
Edit nearest neighbors	56.81	**70.76**	**82.58**	72.04	69.98
One sided selection	69.57	56.14	79.28	87.23	72.60

Table 8: Results for German using sampling (69 541 features).

Consequently, we again come to the conclusion that we cannot generalize across languages with regard to which feature types are useful nor to the amount of features that are useful.

5.3 Sampling Methods across Languages

In this section, we look into the effects of sampling methods. Since the problem of abusive language detection is inherently skewed towards non-abusive language, instance sampling on the training set can help make the classifier more sensitive towards the minority class. We investigate the use of over-sampling of the minority class and under-sampling of the majority class using the best performing number of IG features for German (69541) and approximately the number of top IG features for English (70,000). The results of these experiments are shown in table 7 for English and in table 8 for German.

For English, table 7 shows that we reach the highest precision for the abusive class using one-sided selection, an under-sampling method. The highest recall for the abusive class and the highest precision for the non-abusive class are reached by SMOTE, an over-sampling method. The highest recall for the non-abusive class is reached without any sampling. The highest F-score across both methods (77.94) is reached by using edit nearest neighbors, which reaches a somewhat lower precision on the abusive class than one-sided selection, but a recall that is about 12.5 points higher.

For German, table 8 shows a different picture: We reach the highest precision for the abusive class along with the highest recall for the non-abusive class and the highest overall F-score (73.26) in the experiments without any sampling. The highest recall for the abusive class along with the highest precision for the non-abusive class is reached by the edit nearest neighbors under-sampling method.

These results show that we again face a situation in which the two data sets behave completely differently. For English, we reach the best results with an under-sampling method while for German, all of the sampling methods perform worse that not using sampling at all. Additionally, while the experiments without sampling show the same trends – high precision for abusive language and high recall for non-abusive language – across both languages, this is not the case for edit nearest neighbors: Here the English results show a high precision for the abusive class, but the German results show high recall for the same class.

	Abusive		Non-Abusive		
Language	Precision	Recall	Precision	Recall	F-score
English	33.98	53.23	70.82	52.32	52.59
German	36.97	51.46	68.32	54.41	51.80

Table 9: LDA Topic Modeling Classifications

5.4 Topic Behavior across Languages

One important distinction that needs to be better understood during the classification process is whether our classifiers detect abusive language itself, or whether they detect topics that are more likely to induce abusive language. If certain topics are strongly associated with abusive language in the training data then the model obtained may focus on these topic associations and not model abusive language directly. More specifically, if this trend is more pronounced for one language, this may explain some of the differences in results across the two languages that we found in previous sections. In order to investigate this possibility, we perform topic modeling on the tweets of the two data sets. This experiment is based on the assumption that if the topic modeler groups tweets similar to the abusive, non-abusive classes, then we have evidence that the abusive language is closely associated with a content topic. We determine the overlap between the topic modeler and the abusive/non-abusive split by calculating precision, recall, and F_1 between the topic models decisions and the gold standard.

We perform topic modeling with the number of topics set to 2, parallel to the grouping into abusive and non-abusive language. However, the topic modeler does not tell us which of the two topics corresponds to abusive language. Thus we calculate precision and recall for both correspondences and choose the one with the higher F-score.

The outcomes of this comparison are presented in table 9. The table shows that both languages follow the same trend with an F-score slightly higher than chance (52.59 for English and 51.80 for German). For both languages, recall is around 50, and precision is higher for the non-abusive class and lower for the abusive one. We conclude from these results that there is no meaningful overlap between topics and abusive/non-abusive language. I.e., our SVM classifiers do learn characteristics of abusive and non-abusive language rather than characteristics of topics. However, this also means that the differences between the languages found in the previous experiments cannot be explained by differences in associating abusive language with specific topics, and we need to investigate further to determine the source of those differences.

6 Conclusion and Future Work

In this paper, we have started an in-depth investigation into two data sets for abusive language detection, one in English and one in German. While the data sets were collected independently and annotated with different classes, collapsing them into a binary annotation between abusive and non-abusive language resulted in data sets that superficially showed similar characteristics. However, our investigation has shown notable differences: For English, XGBoost provides the best performance, for German SVMs do. Stems and dependencies in addition to character n-grams help for German while they are harmful for English. For German, we need to use slightly less than half of all features while for English, we only need 4.5%. Even though the English data set is larger than the German one, the percentages translate into a much larger feature set for German than for English. Additionally, for English sampling improves results while for German, it does not. One hypothesis, namely that the differences may be related to a stronger topical effect in the abusive tweets, was rejected based on an experiment with a topic modeler.

Moving forward, we will investigate the differences between the German and English data sets in more detail. Finding the causes will allow us to better approach optimization of multilingual abusive language detection systems. The final goal of this work is the development of a system for detecting abusive language that can truly work in a multilingual fashion.

Acknowledgements

We are grateful to Noor Abo Mokh, Leah Schaede and Zuoyu Tian, who helped in initial stages of the project.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*. pages 265–283.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*. pages 1638–1649.

Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. 2017. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*. pages 759–760.

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, 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*. pages 54–63.

Anders Björkelund, Bernd Bohnet, Love Hafdell, and Pierre Nugues. 2010. A high-performance syntactic and semantic dependency parser. In *Proceedings of the 23rd International Conference on Computational Linguistics*. Beijing, China, pages 33–36.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research* 3:993–1022.

Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research* 16:321–357.

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* .

Elise Fehn Unsvåg and Björn Gambäck. 2018. The effects of user features on Twitter hate speech detection. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*. Brussels, Belgium, pages 75–85.

Paula Fortuna and Sérgio Nunes. 2018. A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)* 51(4):85.

Paula Cristina Teixeira Fortuna. 2017. *Automatic detection of hate speech in text: an overview of the topic and dataset annotation with hierarchical classes*. Master's thesis, Universidade de Porto.

Thiago Galery, Efstathios Charitos, and Ye Tian. 2018. Aggression identification and multi lingual word embeddings. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*. pages 74–79.

Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. 2005. Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pages 878–887.

Haibo He, Yang Bai, Edwardo A. Garcia, and Shutao Li. 2008. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *Proceedings of the 5th IEEE International Joint Conference on Neural Networks*. Hong Kong, pages 1322–1328.

Lingpeng Kong, Nathan Schneider, Swabha Swayamdipta, Archna Bhatia, Chris Dyer, and Noah Smith. 2014. A dependency parser for tweets. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar, pages 1001–1012.

Miroslav Kubat and Stan Matwin. 1997. Addressing the curse of imbalanced training sets: One-sided selection. In *Proceedings of the 14th International Conference on Machine Learning*. Nashville, TN, volume 97, pages 179–186.

Ritesh Kumar, Atul Kr Ojha, Shervin Malmasi, and Marcos Zampieri. 2018. Benchmarking aggression identification in social media. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*. pages 1–11.

Younghun Lee, Seunghyun Yoon, and Kyomin Jung. 2018. Comparative studies of detecting abusive language on Twitter. In *Proceedings of the Second Workshop on Abusive Language Online (ALW2)*. Brussels, Belgium, pages 101–106.

Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. 2017. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research* 18(17):1–5.

Can Liu, Sandra Kübler, and Ning Yu. 2014. Feature selection for highly skewed sentiment analysis tasks. In *Proceedings of the Second Workshop on Natural Language Processing for Social Media (SocialNLP)*. Dublin, Ireland, pages 2–11.

Prasenjit Majumder, Mandar Mitra, Swapan K Parui, Gobinda Kole, Pabitra Mitra, and Kalyankumar Datta. 2007. YASS: Yet another suffix stripper. *ACM Transactions on Information Systems (TOIS)* 25(4).

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

Sandip Modha, Prasenjit Majumder, and Thomas Mandl. 2018. Filtering aggression from the multilingual social media feed. In *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*. pages 199–207.

Joaquín Padilla Montani and Peter Schüller. 2018. TUWienKBS at GermEval 2018: German abusive tweet detection. In *Proceedings of GermEval 2018, 14th Conference on Natural Language Processing (KONVENS 2018)*. Vienna, Austria, pages 45–50.

Hamdy Mubarak, Kareem Darwish, and Walid Magdy. 2017. Abusive language detection on Arabic social media. In *Proceedings of the First Workshop on Abusive Language Online*. Vancouver, Canada, pages 52–56.

Hien M. Nguyen, Eric W. Cooper, and Katsuari Kamei. 2011. Borderline over-sampling for imbalanced data classification. *Journal International Journal of Knowledge Engineering and Soft Data Paradigms* 3(1):4–21.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the ACL Conference on Empirical Methods in Natural Language Processing*. pages 79–86.

Ji Ho Park and Pascale Fung. 2017. One-step and two-step classification for abusive language detection on Twitter. In *Proceedings of the First Workshop on Abusive Language Online*. Vancouver, Canada, pages 41–45.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Björn Ross, Michael Rist, Guillermo Carbonell, Benjamin Cabrera, Nils Kurosky, and Michael Wojatzki. 2016. Measuring the reliability of hate speech annotations: The case of the European refugee crisis. In *Proceedings of the Workshop on Natural Language Processing for Computer- Mediated Communication (NLP4CMC)*. Bochum, Germany, pages 6–9.

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*. Valencia, Spain, pages 1–10.

Cynthia Van Hee, Els Lefever, Ben Verhoeven, Julie Mennes, Bart Desmet, Guy De Pauw, Walter Daelemans, and Véronique Hoste. 2015. Detection and fine-grained classification of cyberbullying events. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*. pages 672–680.

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*. San Diego, CA, pages 88–93.

Michael Wiegand, Melanie Siegel, and Josef Ruppenhofer. 2018. Overview of the GermEval 2018 shared task on the identification of offensive language. In *Proceedings of GermEval 2018*. Vienna, Austria.

Dennis L. Wilson. 1972. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetrics* 2(3):408–421.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. SemEval-2019 task 6: Identifying and categorizing offensive language in social media (OffensEval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*.