Seminar für Sprachwissenschaft
Eberhard-Karls-Universität Tübingen

Cross-lingual sentiment analysis for Russian and German

Anastasia Vostrova

Bachelor Thesis

Supervisor: Prof. Kurt Eberle

Contents

Abstract

With the increasing amount of textual information on the internet, the field of sentiment analysis grew more and more popular. However, research in the field mainly focuses on the analysis of English while other languages are less explored. Therefore, in this thesis we perform sentiment classification of Russian tweets and their German translation using two algorithms: Support Vector Machine and Naive Bayes. We find that regardless of the differences in the grammatical sentence structure of German and Russian, the classification delivers very similar accuracy scores for both languages. In particular, we see that for both languages the Support Vector Classifier yields better accuracy scores than the Naive Bayes Classifier. Furthermore, unigrams as document features are more effective than bigrams and lemmatization has a very similar impact on classification results of tweets in Russian and German.

1 Introduction

As social media grew, people started expressing their opinion on the internet more and more on any topic they like. That is why the interest in extracting sentiment from internet-based corpora has emerged and Sentiment Analysis became popular. The main goal of sentiment analysis is to determine the polarity of a text, which is not a trivial task, because often even humans can not reach an agreement on the sentiment of a text.

There are multiple reasons why sentiment analysis is in great request. First of all, big companies are interested in analyzing people's reviews of their product with the goal of adapting it to people's needs (R.M, 2001). Sentiment analysis can also help to conduct sociological studies, which help to determine, for instance, which political views the majority of people have Mullen and Malouf (2006), identify social issues (Karamibekr and Ghorbani, 2012) or analyze people's reactions to significant events (Burnap et al., 2014). Moreover, it can help to make predictions about, for example, movie sales (Mishne et al., 2006) or the stock market behavior (Grigoryan, 2017).

Sentiment analysis belongs to a class of text classification tasks and existed before social networks gained their popularity. Now, to deal with a big amount of data, automatic methods for sentiment extraction became widespread. The aim of automatic methods is to detect the sentiment of a text according to its features that can be represented as, for instance, unigrams or bigrams. Automatic sentiment extraction can involve classification of texts into three or more classes, for example, into positive, negative and neutral, which is called multiclass classification or into just two classes like positive and negative, which is called binary classification. In this thesis we are going to explore the second one.

After carrying out the classification, the important step is evaluation. To evaluate the results of sentiment analysis, they are compared with the results of another more reliable classification ("gold standard"), for example, with human judgment. Evaluation metrics like Precision, Recall, F-score

and Accuracy are used and in some cases are calculated separately for each classification class.

There are also more complex variations of sentiment analysis like aspect-based sentiment analysis, the goal of which is not only to identify the sentiment of the text, but also to specify aspect terms, to which this sentiment belongs. For example, Rybakov and Malafeev (2018) analyzed Russian hotel reviews with the aspect terms: room, location and service. This type of sentiment analysis will not be covered in the thesis.

In our experiment we perform binary sentiment classification of tweets in Russian and German using two classifiers. The aim of the experiment is to find possible differences of Russian and German languages that affect the classification results. Further goals are to discover which document features improve sentiment analysis results, which classifier works best for the analysis of each Twitter dataset and in which way lemmatization affects the classification.

This thesis is structured as follows: in Section 2 different approaches to sentiment analysis and studies that are relevant for this thesis are introduced. Section 3 provides a description of the data that is used for the experiment of this thesis. Section 4 details the sentiment analyzer and the setup of the experiment. In Section 5 the results of the experiment and in section 6 the summary and possible future work are presented.

2 Related work

There are multiple methods for automatic sentiment classification, all of which can be grouped into lexicon based, machine learning based and mixed methods. Lexicon based methods require previously composed sentiment dictionaries, according to which the sentiment analysis is carried out, while machine learning based methods do not require a lexicon, instead they use a labeled dataset to train a sentiment classifier. Mixed methods combine both, lexicon based and machine learning based.

2.1 Machine learning based methods

The common machine learning approaches include Random Forests, NB (Naive Bayes), MaxEnt (Maximum Entropy), SVMs (Support Vector Machines) and ANNs (Artificial Neural Networks). MaxEnt is the oldest approach, known since the first half of 19th century (Cramer, 2002). ANN was proposed in 1943 (McCulloch and Pitts, 1943), SVM and NB are known since 1960s (Maron, 1961; Chervonenkis, 2013) and Random Forest is the newest method, which was proposed in 1995 by Ho (1995). They can be applied in multiple fields. For example, all of them can be applied in medicine: MaxEnt for mortality prediction in injured patients (Boyd et al., 1987), NB for medical diagnosis determination (Rish et al., 2001), ANN for classification of lung sounds (Sengupta et al., 2016), SVM for medical image analysis(Cuingnet et al., 2011), Random Forest for determining an adequate drug dose for patients (Rahman et al., 2019).

2.1.1 Random Forest

A random forest is a set of decision trees (Correia and Schwartz, 2016). A decision tree is a tool of tree-like structure, that is used for making predictions. In case of sentiment classification, each tree node checks if the input has the feature characteristics for a specific class. The classification proceeds to the next node via the edge, that corresponds to the correct answer.

A Random Forest processes the classification results of the decision trees and outputs the most frequent value or the mean of the values produced by the decision trees. While individual trees may not perform to a satisfactory degree, in combination they perform quite well. Random Forests are used in multiple studies in the field of natural language processing and reached impressive 91% of accuracy on a movie review dataset in Parmar et al. (2014).

2.1.2 Naive Bayes Classifier

This classifier uses the so-called bag-of-words principle. It considers only the frequency of words and disregards their position in a document. In the context of a Naive Bayes classifier it does not matter if the word is in the first, second or any other position, its impact on the classification is the same. For example, for the sentences "He walks out of the house." and "Out of the house he walks." or even the ungrammatical sentence "Walks he the house out of." NB yields the same classification result. Hence, the reason for the classifier to be called "naive" is that it makes the "naive" assumption that the word order does not affect the classification. Naive Bayes classifier makes use of Bayes' Theorem, by which we can calculate the probability of an event given that some other related event has already happened. The theorem represents a conditional probability with the help of three other probabilities:

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)}$$

where d is a document and c is a class, for example, negative or positive. So in our case we count the probability of a document d belonging to a class c. Since we are computing a label for one document, the divisor will stay constant, so that we can drop it. Hence, we get the formula $P(c|d) = P(d|c)P(c)$. We can also represent a document as a set of features: $(f_1, f_2, ..., f_n)$, so that $P(d|c) = P(f_1, f_2, ..., f_n|c)$. Following the naive assumption that features are independent from each other, we can multiply the probabilities as $P(f_1|c) \cdot P(f_2|c) \cdot ... \cdot P(f_n|c)$. However, some of the probabilities can be equal to zero, so we use so-called add-one Laplace smoothing. We add $+1$ to every probability count, so that there is no multiplier that equals zero. We compute $P(f_1, f_2, ..., f_n|c)P(c)$ for each of the classes and after that we take the class with the biggest probability as our result. Hence, the final formula looks as follows:

$$c_{NB} = \arg\max_{c \in C} P(c) \prod_{f \in F} P(f|c)$$

where C is a set of classes, F is a set of document features and c_{NB} is the document class we are trying to find (Jurafsky and Martin, 2014).

2.1.3 Maximum Entropy Classifier

The MaxEnt classifier is, like NB, probabilistic and linear, but it estimates probabilities in a different way. MaxEnt also does not assume that document features are independent and can deliver better results than NB when feature dependence plays a more significant role in certain tasks (Pang et al., 2002). Given a class c, the NB algorithm tries to predict the features of the input document d, calculating the probability $P(d|c)$. Then the Bayes theorem is used to compute $P(c|d)$.

In case of MaxEnt, the algorithm identifies the possible values of the class c and computes $P(c|d)$ directly. MaxEnt algorithm multiplies each feature by a weight and sums them up:

$$P(c|d) = \frac{exp\left(\sum_{i=1}^{N} \lambda_i f_i(c,d)\right)}{\sum_{c' \in C} exp\left(\sum_{i=1}^{N} \lambda_i f_i(c',d)\right)}$$

where N is the number of classes, C is the set of classes, $f_i(c,d)$ is a feature i for class c given the document d, λ is a weight vector that determines how significant the feature in the classification is (Kharde et al., 2016; Go et al., 2009; Jurafsky and Martin, 2014).

2.1.4 Support Vector Machine

Support vector classification is carried out using an algorithm called SVM (Support Vector Machine). It can perform linear as well as non-linear classification. In case of linear classification the classifier receives a set of training examples as an input. The SVM algorithm generates a model that aims at classifying new unseen samples into one of two categories. At first, as the algorithm gets input examples, it represents the examples as datapoints in

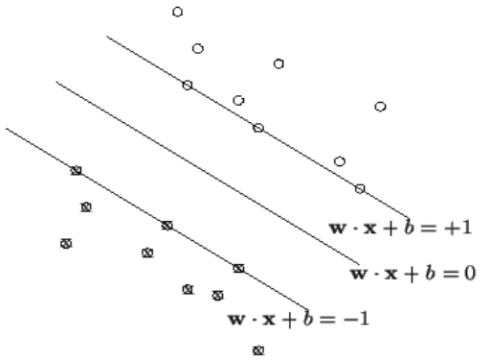

Figure 1: Hyperplanes and support vectors (Khan et al., 2010).

space. Then it finds two parallel hyperplanes that divide the input examples into two classes. These hyperplanes should lie as far from each other as possible and the distance inbetween is called *margin*. We are looking for the hyperplane that lies in the middle of the margin, the so-called "maximum-margin hyperplane". The vectors that are nearest to the maximum-margin hyperplane on both sides are called support vectors. Figure 1 depicts three hyperplanes and six support vectors. The class of the new input example is predicted according to the side of the hyperplane it falls on.

The main drawback of SVMs is that they require a lot of computational resources: the processing time is not only longer than for most other algorithms, but it grows quadratically relative to the training set size (Colas and Brazdil, 2006). In addition, it consumes a huge amount of memory in contrast to many other algorithms used for sentiment analysis (Khan et al., 2010).

2.1.5 Artificial Neural Networks

The most sophisticated algorithm is used by ANNs, being originally built analogous to biological neural networks. ANNs are structured as layers of neurons, which are simple processing units. The bigger the number of neurons the more complicated the problems are that can be solved by an ANN. The first layer is called "input layer", because it receives and stores the input of an ANN. The last layer is called "output layer", because it produces and stores the output of a neural network. Between them there are hidden layers, where

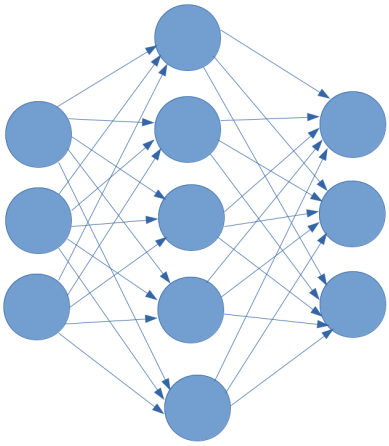

Figure 2: Artificial neural network structure with one hidden layer.

most of the work of the artificial intelligence is done. The input goes through all hidden layers while being transformed in each of them. The neurons of each layer are connected to the neurons in the previous layer. Simpler versions of ANNs can contain no hidden layers or just one. In Figure 2 the structure of an ANN with one hidden layer is depicted.

The disadvantages of ANNs include high computing costs and the impossibility for most users to fully understand what processes take place in the hidden layers (Khan et al., 2010).

2.1.6 Relevant studies

The most popular type of Neural Networks used for sentiment analysis is Convolutional Neural Networks. One of the main differences of the CNN structure compared to a regular ANN is that the neurons of one layer are not fully connected to all the neurons in the next layer. CNNs are used for sentiment analysis of different languages, including German and Russian (von Grünigen et al., 2018; Smetanin and Komarov, 2019).

Nevertheless, ANNs do not always outperform other methods in sentiment analysis. Recently proposed ANN algorithms show rather good results in non-binary classification tasks, but for binary ones there are other algorithms that work better as demonstrated in (Wiegand et al., 2018). In Wiedemann et al.

(2018) a model with neural structure yielded the best results for multiclass classification. However, the binary classification of German tweets in Montani (2018) performed best with a method that was not based on ANNs. Moreover, he mentions that the Neural Network even worsened the F1 score. Hence, sometimes it is more sensible to choose classifiers like NB, SVM or MaxEnt.

One of the first papers about binary text classification with the help of machine learning techniques in sentiment classification is by Pang et al. (2002), who try to discriminate between positive and negative English movie reviews. For this purpose three classifiers are used: SVM, MaxEnt and NB. Their data suggests that the best results are achieved with an SVM. Since then there have been more studies for English than for any other language, which is why these are mostly mentioned in this thesis. For example, Go et al. (2009) perform a binary English tweets classification with NB, SVM and MaxEnt, concluding that SVM outperformed other models with an accuracy of 82.2%. Both studies (Pang et al., 2002; Go et al., 2009) also use different document features like unigrams and bigrams, which influence the classification results greatly. Therefore, unigrams and bigrams are also used as document features in the experiment of this thesis.

There have been much less studies for German and even fewer for Russian, yet there are some worth mentioning: Yussupova et al. (2012) perform a binary classification of Russian bank reviews with NB and SVM classifiers. The SVM showed better results than NB achieving an accuracy score of 88.30% while NB yielded a score of 87.69% accuracy. However, as pointed out by Medagoda et al. (2013) a weakness of the study is an unbalanced corpus with 304 positive and 850 negative reviews. Interestingly, they also tested their approach on an English dataset and the classification results for this dataset also provide evidence for the advantage of the SVM algorithm over NB. In Chetviorkin and Loukachevitch (2013) the best accuracy scores for a 2-class classification task are yielded by the solutions that employed an SVM in combination with other methods like dictionaries and rule-based systems, which will be described in the next section. Loukachevitch and Rubtsova

(2015) describe the results of the competition SentiRuEval, the main aim of which is to analyze different sentiment classifiers that are used for the analysis of Russian tweets for binary classification. According to this paper, the most popular and most effective classifier is SVM. So, it seems that for Russian sentiment analysis SVMs continuously outperform other classification methods. For German, however, it is not always the case. For example, Xi et al. (2018) found that in a 2-class classification F1 score was higher for NB than for SVM while Precision was higher for SVM than for NB.

There are also studies, results of which appear to be more complex and from which it is hard to conclude which classifier works better for linear classification problems. For instance, Colas and Brazdil (2006) use an English E-mail dataset as training data to compare SVM to KNN and NB on a binary classification task. Here all three classifiers achieve a rather similar performance with a bigger training corpus. However, with a corpus of a smaller size the NB classifier performs best. Concerning the training time, the SVM classifier needs the most time for training. In Christiana et al. (2017) for binary classification the accuracy score for NB is higher on a dataset of movies, products and restaurant reviews and delivers almost the same results as the SVM on an SMS spam collection dataset. However, the best results are delivered by combining SVM and NB, which is consistent with the results of the experiment by Jain and Mishra (2016) who also obtain the best score with a model that combines two algorithms: NB with a modification of the MaxEnt classifier. However, Christiana et al. (2017) report that building the combined model takes eight times longer than for an SVM.

Unfortunately, for machine learning methods training and testing data are needed. By using training data a classifier learns to distinguish between classes while testing data is used for performance validation. Hence, the downside of these methods is that they require a labeled corpus and often significant computational power.

2.2 Lexicon based methods

Lexicon based methods use sentiment lexicons[1] that were composed in advance. The basic idea of these methods is to count the number of positive and negative words in a text and classify it according to the number that is bigger. If a text has more positive than negative words, then it is classified as positive and the other way around: if it has more negative words, it is classified as negative. In this case a sentiment analyzer is based on rules that are formed using if-else structures. Booster and negation words pose a challenge, as they can complicate the sentiment assignment especially if there is a combination of both types of words in a single sentence. For example, the sentences "The movie was very interesting." and "The movie was not very interesting." should be assigned different sentiments as well as the sentences "The movie was less interesting than the previous one." and "The movie was more interesting than the previous one."

A rule-based approach to German sentiment analysis is proposed in Momtazi (2012). He created a new opinion dictionary for German based on a translated English opinion dictionary. He uses social media comments (mostly YouTube comments) for data annotation. The texts are annotated by three people and as a result the human agreement for positive messages is 57% while for negative messages the agreement is 60%, which shows once again that sentiment analysis is a challenging task, even for humans. The author performs a binary classification as well as a fine-grained classification, which is an advanced version of regular binary classification. As a result of a fine-grained classification each text is assigned a score that indicates the degree of positivity or negativity. In a regular, "coarse-grained", binary classification sentences "The movie is good." and "The movie is excellent." are assigned the same label "positive" while in fine-grained classification the positivity score of the second sentence would be higher. Momtazi (2012)'s model outperformed NB and SVM classifiers in both, fine-grained and coarse-grained binary classification. There have been attempts to build rule-based sentiment

[1] A sentiment lexicon is a list of words that was annotated with sentiment labels.

classifiers for Russian for binary classification, for example by Kan (2012) as a part of a Russian Information Retrieval Seminar submission (Chetviorkin and Loukachevitch, 2013), but it was not as successful as other participants' algorithms based on machine learning methods. One of the best Russian sentiment analysis tools based on tonal dictionaries is by Pazelskaya and Solovyev (2011). Their algorithm reached an accuracy of 93%.

The main problem with lexicon based methods is that it is hard to create a universal dictionary, that can be applied to any context needed, because a sentiment lexicon does not capture the ambiguity of words, which makes these methods domain-dependent. They are language-dependent as well, due to different languages having different grammatical structures, hence the system needs to be adapted to another language and a new sentiment lexicon needs to be provided. Lexicon based methods are especially difficult to use for the sentiment extraction of texts that are written in languages with a flexible word order like Russian. In such cases it is hard to detect a pattern in the word order of the sentence and hence, the position of negation and booster words.

3 Data

We use the corpus RuTweetCorp, which includes 111,923 tweets labeled as negative and 114,991 tweets labeled as positive. The classification of tweets in the corpus was performed using the method described by Rubtsova (2015). We delete all metadata such as dates of publications, number of retweets, etc. and leave only information that is relevant for our experiment: the text of the tweet and its label (negative or positive).

For the purposes of this thesis we use 2,000 Tweets: 1,000 tweets labeled negative and 1,000 tweets labeled positive, due to the lack of memory on the computer that is used for the experiment. The parameters of the operating system, memory and processor used can be found in Table 1. The 2,000 tweets were translated to German with the help of Amazon Translate. The assump-

Operating system	Ubuntu 18.04.2
Memory	15.6 GiB
Processor	Intel Core i7-4510U 2.00GHz x 4

Table 1: The parameters of the operating system, memory and processor used for the experiment.

tion is that the translation will not greatly affect the sentiment analysis with regards to the meaning of most words that are important for classification. As confirmed by Al-Shabi et al. (2017), who use Arabic translation of English texts as training data, machine translation nowadays can deliver reliable results for semantic analysis purposes. Also, Shalunts et al. (2016) shows that sentiment analysis using machine translation of German, Russian and Spanish training data to English results in a maximum 5% performance decline compared to the sentiment analysis performed on the original dataset.

Overall we end up with two datasets: one with Russian tweets and one with their German translation. The preprocessing of the datasets includes the deletion of all URLs (e.g. www.abc.com), targets (@username) and stop words with the help of German and Russian models from the spaCy library. However, negations in the tweets are preserved due to their importance for the detection of sentiment.

4 Sentiment analyzer and experiment setup

The classification is carried out with the help of the sentiment analyzer from python module *nltk.sentiment.sentiment_analyzer* (NLTK Sentiment Analyzer). This package module provides tools for sentiment analysis tasks and allows to import any classifier needed for the analysis. The sentiment analyzer tokenizes the input, identifies and extracts the features and passes them to a classifier. We are interested in linear classifiers, because we treat tweets as either negative or positive, hence, our training data is linearly separable. In this thesis we compare Naive Bayes and Support Vector Machine classifiers. The

Classifier	Document features	Lemmatization
SVM	Bigrams	yes
SVM	Unigrams	no
NB	Bigrams	yes
NB	Unigrams	no

Table 2: Sentiment analyzer variations used in the experiment.

NB classifier is taken from python module *nltk.classify.naivebayes.NaiveBayesClassifier* and is not changed for the experiment (NLTK Naive Bayes).

The SVM classifier, namely Linear Support Vector Classifier (LinearSVC), is taken from scikit-learn library (Pedregosa et al., 2011). We choose an SVM with a linear kernel, because studies show its effectiveness in binary classification in comparison to other kernels (Christiana et al., 2017; Colas and Brazdil, 2006). An SVM with a linear kernel also needs less time for prediction in binary classification (Pahwa and Sinwar, 2015). All parameters of the classifier are left at default settings which are described here: LinearSVC

We experiment with using unigrams and bigrams as document features. We use all unigrams while we take into consideration only bigrams that occur 3 times or more.

We perform one classification with preliminary lemmatization of the dataset and one without it. Lemmatization brings different inflected forms of one word to their infinitive form. For example, the words “бежит” (eng: runs) and “бегут” (eng: run, plural, third person) are brought to their basic form: “бежать” (eng: run). We perform Russian lemmatization with the help of the pymorphy2 library (Korobov, 2015) and German lemmatization with the help of the spaCy library. The sentiment analyzer variations, that are used in the experiment are specified in Table 2.

As the input of the sentiment analyzer we take tweets labeled as positive and negative. 80% of the data is used for training the classifier and 20% for testing. After that 5-fold cross validation is used for the estimation of the obtained results. The evaluation metrics that we use are F1 score and

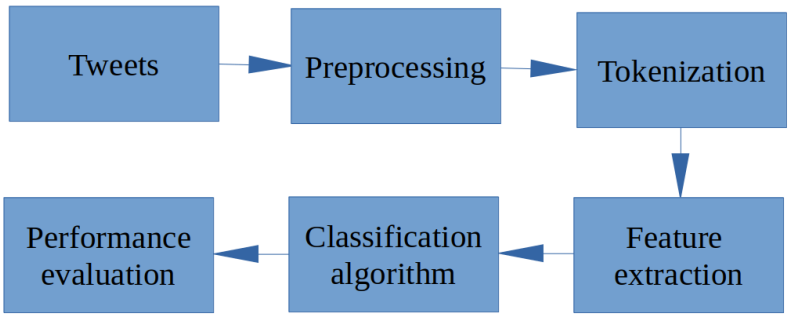

Figure 3: Sentiment analysis processing pipeline

	Russian	German
Vocabulary length with stopwords	25,048	28,658
Vocabulary length without stopwords	19,264	17,124
Lemmatized unigrams	5,011	4,246
Unigrams	7,000	4,887
Lemmatized bigrams	426	442
Bigrams	305	372

Table 3: Vocabulary length and number of features with 2,000 samples

Accuracy. We calculate F1 score for negative and positive classes separately while Accuracy is calculated for the whole testing data.

The pipeline of the whole sentiment classification process can be seen in Figure 3.

The code for the experiment can be found on GitHub.

5 Results and Discussion

First, let us consider Table 3, in which we can see that due to the deletion of stop words the length of the German vocabulary decreases by about 11,500 words while in Russian the difference between the length of vocabulary with stop words and without them is only about 5,700 words. The reason for that can be the extensive usage of articles and auxiliary verbs in German. They are considered to be stop words, since they do not carry much semantic mean-

	LinearSVC		**Naive Bayes**	
	Russian	German	Russian	German
Lemmatized unigrams	**0.993**	**0.989**	0.982	0.977
Unigrams	0.992	0.987	0.982	0.976
Lemmatized bigrams	0.852	0.871	0.862	0.864
Bigrams	0.857	0.877	0.858	0.856

Table 4: Five-fold cross-validation accuracies with 1,000 samples

	LinearSVC		**Naive Bayes**	
	Russian	German	Russian	German
Lemmatized unigrams	0.995	**0.992**	0.988	0.979
Unigrams	**0.996**	**0.992**	0.988	0.979
Lemmatized bigrams	0.868	0.883	0.893	0.886
Bigrams	0.867	0.888	0.881	0.885

Table 5: Five-fold cross-validation accuracies with 2,000 samples

ing and are deleted in the preprocessing stage. The structure of a Russian sentence does not require articles or auxiliary verbs, which is why the list of stop words in German used by spaCy library is 544 words long and the list of Russian stop words contains just 264 words.

$$\begin{array}{cccc} \text{Ich} & \text{bin} & \text{ein} & \text{Student.} \\ \textit{I} & \textit{am} & \textit{a} & \textit{student.} \end{array} \tag{1}$$

$$\begin{array}{cc} Я & \text{студент.} \\ \textit{I} & \textit{student.} \end{array} \tag{2}$$

For example, for German sentence in (1) “bin” and “ein” have to be deleted since these are stop words. Meanwhile, the Russian translation in (2) does not contain any stop words to be deleted.

The accuracy scores with 2,000 samples are illustrated in the Table 5. LinearSVC yields near-perfect results for both languages: 0.992 for German and 0.996 for Russian. The reason for such a high accuracy score can be the

way the corpus was constructed. All tweets that contained both negative *and* positive emotions were filtered out and only obviously positive or obviously negative tweets were left. Similar results were presented in the paper by (Christiana et al., 2017), where both SVM and NB classifiers reached an accuracy of more than 98%.

In general, the accuracy is higher for Russian language, which is possibly due to some sentiment information of German tweets being lost during translation. However, the difference between the best results for Russian and German is not very big: 0.996 and 0.992 respectively.

The most effective document features for both classifiers and languages turned out to be unigrams. These results are remarkable, because bigrams are thought to capture context better than unigrams (Kharde et al., 2016).

Das	Wetter	ist	schön	heute.
The	*weather*	*is*	*nice*	*today.*

(3)

Das	Wetter	ist	nicht	schön	heute.
The	*weather*	*is*	*not*	*nice*	*today.*

(4)

For example, tweet (3) belongs to the positive class and tweet (4) to the negative class. If we use unigrams as features, we get the unigram "schön" (eng: nice) in both tweets, hence, the unigram does not indicate which sentiment the tweet corresponds to. However, if we use bigrams, then we can determine the negative sentiment of the tweet (4) with the help of the bigram "nicht schön" (eng: not nice).

Nevertheless, bigrams seem not to capture context well enough in our experiment setup, which is consistent with the results of Pang et al. (2002). They also came to the conclusion that both classifiers, NB and SVM, perform better using unigrams. It may be due to the sparseness of bigrams which reduces overall accuracy (Go et al., 2009). In their experiment Go et al. (2009) provide evidence that unigrams as features for twitter data are more

effective than bigrams. However, there are some studies, that report different results: Al-Shabi et al. (2017) perform the cross-lingual classification from English to Arabic and come to the conclusion that the NB classifier achieves better results with bigrams than with unigrams.

In Tables 4 and 5 we can see that in our experiment lemmatization has a greater effect on the classification using bigrams than unigrams. In case of unigrams it has either no or little effect. The lemmatized bigrams, on the other hand, improve the accuracy score for the NB classifier for both languages when it is trained on 1,000 samples, but has a more significant positive effect on Russian when training with 2,000 samples. However, the classification results for LinearSVC are different: lemmatization of bigrams makes the score worse for both languages with the exception of a slight improvement on the Russian dataset with 2,000 samples.

Nevertheless, the results for Russian sentiment classification with lemmatization of unigrams are slightly better than without lemmatization on the dataset with 1,000 samples in case of LinearSVC, which is consistent with Yussupova et al. (2012). They explain this phenomenon with the fact that lemmatization causes grouping of cognate words with the same semantic but different endings. The datasets they used contains 1,154 samples, but the results of our experiment show that when the dataset gets bigger (up to 2,000 samples) the results for LinearSVC are better without lemmatization. It is worth mentioning that Yussupova et al. (2012) also performed an English sentiment analysis on a dataset of 2,000 samples and the lemmatization worsened the classification accuracy score.

Surprisingly, the effect of lemmatized unigrams is very similar for Russian and German, although the effect was expected to be greater for Russian. The reason for that expectation is that there are more inflected forms of one word in Russian than in German or English. The usage of multiple affixes for providing grammatical information is one of the distinguishing features of inflectional languages like Russian (VanWagenen and Pertsova, 2014). For example, in sentences (5) and (6), verb forms “была” and “был” and adjective

		LinearSVC		**Naive Bayes**	
		Russian	German	Russian	German
Lemmatized unigrams	F1 score [pos]	0.995	0.992	0.988	0.980
	F1 score [neg]	0.995	0.992	0.988	0.979
Unigrams	F1 score [pos]	0.996	0.992	0.989	0.979
	F1 score[neg]	0.996	0.992	0.988	0.979
Lemmatized bigrams	F1 score [pos]	0.873	0.874	0.899	0.893
	F1 score [neg]	0.862	0.890	0.887	0.879
Bigrams	F1 score [pos]	0.868	0.880	0.889	0.893
	F1 score[neg]	0.864	0.896	0.871	0.877

Table 6: Five-fold cross-validation F1 score for positive and negative classes of tweets for 2,000 samples.

forms “одна” and “один” look the same in German translation as well as in English translation. Hence, as can be seen in the Table 3 the number of unigrams after lemmatization in Russian decreased from 7000 to 5011 while in German the number only went from 4887 to 4246. However, the reduced number of unigrams did not affect the classification greatly.

With bigrams the situation is different. Their number increases with lemmatization, because there are more lemmatized bigrams that occur at least 3 times and hence they are taken into account by the sentiment analyzer.

Она	была	одна	дома.	
Sie	war	allein	zu Hause.	(5)
She	*was*	*alone*	*at home.*	

Он	был	один	дома.	
Er	war	allein	zu Hause.	(6)
He	*was*	*alone*	*at home.*	

Table 6 shows that for the majority of the cases NB classifier works better for the prediction of positive tweets, which is consistent with results in Momtazi (2012), but in our experiment setup the SVM classifier delivers a

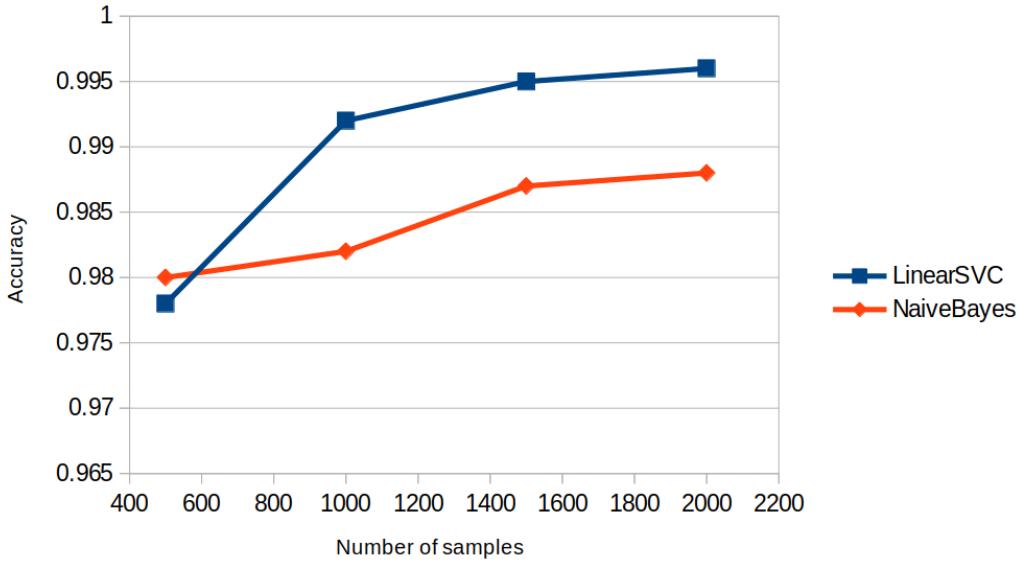

Figure 4: Accuracy results of the classification of Russian tweets using unigrams as document features according to the number of samples in the corpus.

better prediction for negative German tweets and positive Russian tweets if used with bigrams as document features. In Momtazi (2012), however, SVM showed better results for positive German tweets.

It is also important to note that many methods often benefit from more data (Colas and Brazdil, 2006; Fatima and Srinivasu, 2017) and our experiment is not an exception. As can be seen in the Figures 4 and 5 the accuracy grew with the number of samples, but surprisingly, the accuracy for the NB classifier did not grow when the number of used samples grew from 1,500 to 2,000 in the German dataset. With 500 tweets, the NB classifier performed better on both, Russian and German datasets. Therefore, it appears that there is an advantage in using NB for small sample sizes, as also stated in Colas and Brazdil (2006).

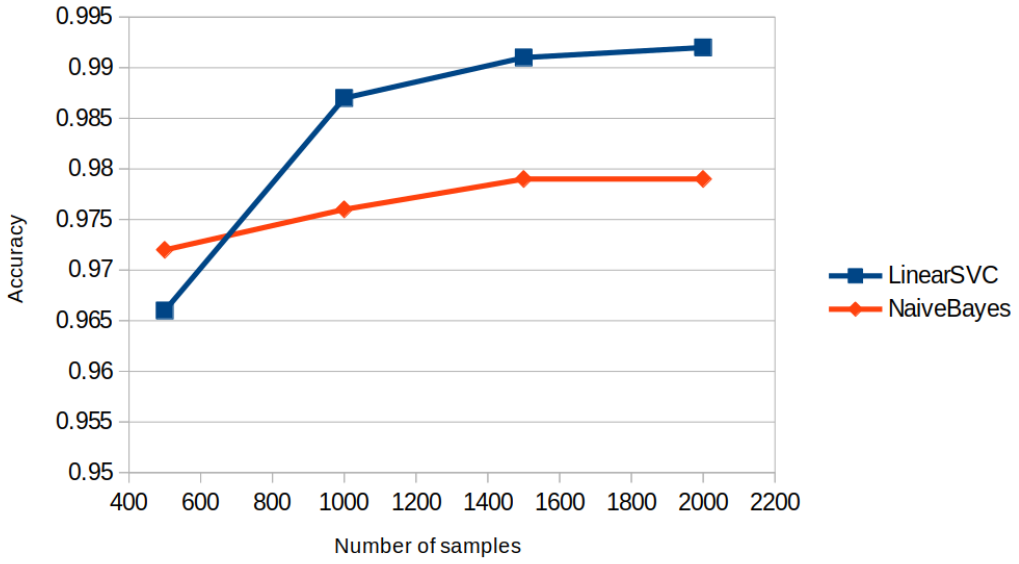

Figure 5: Accuracy results of the classification of German tweets using unigrams as document features according to the number of samples in the corpus.

6 Conclusion

In this thesis we have performed a sentiment classification of Russian tweets and their German translation using Support Vector and Naive Bayes classifiers. While the SVM classifier performed best on both datasets, NB also reached satisfactory results. The experiment performed provides evidence that regardless of the differences in Russian and German, the classification results appear to be rather similar: the document features, that improve the classification are the same, namely unigrams, and accuracy scores do not differ much as well. Additionally, we examined the effects of lemmatization on the classification performance. The effect was substantial for the classification with bigrams but not with unigrams for both languages examined.

However, because of the lack of computational power, we did not have an opportunity to experiment with classifier parameters, which can be a good extension of the thesis. Future work could also experiment with more training data samples, try out different limits of unigram and bigram numbers or use POS tags as document features.

References

Adel Al-Shabi, Aisah Adel, Nazlia Omar, and Tareq Al-Moslmi. Cross-lingual sentiment classification from english to arabic using machine translation. *Int. J. Adv. Comput. Sci. Appl.(IJACSA)*, 8(12):434–440, 2017.

Amazon Translate. URL `https://aws.amazon.com/translate/`.

Carl R Boyd, Mary Ann Tolson, and Wayne S Copes. Evaluating trauma care: the triss method. trauma score and the injury severity score. *The Journal of trauma*, 27(4):370–378, 1987.

Pete Burnap, Matthew L Williams, Luke Sloan, Omer Rana, William Housley, Adam Edwards, Vincent Knight, Rob Procter, and Alex Voss. Tweeting the terror: modelling the social media reaction to the woolwich terrorist attack. *Social Network Analysis and Mining*, 4(1):206, 2014.

Alexey Ya Chervonenkis. Early history of support vector machines. In *Empirical Inference*, pages 13–20. Springer, 2013.

Ilia Chetviorkin and Natalia Loukachevitch. Evaluating sentiment analysis systems in russian. In *Proceedings of the 4th biennial international workshop on Balto-Slavic natural language processing*, pages 12–17, 2013.

Abikoye Oluwakemi Christiana, Omokanye Samuel Oladeji, and Aro Taye Oladele. Binary text classification using an ensemble of naïve bayes and support vector machines. *Computer Science & Telecommunications*, 52(2), 2017.

Fabrice Colas and Pavel Brazdil. Comparison of svm and some older classification algorithms in text classification tasks. In *IFIP International Conference on Artificial Intelligence in Theory and Practice*, pages 169–178. Springer, 2006.

Artur Jordao Lima Correia and William Robson Schwartz. Oblique random forest based on partial least squares applied to pedestrian detection. In

2016 IEEE International Conference on Image Processing (ICIP), pages 2931–2935. IEEE, 2016.

Jan Salomon Cramer. The origins of logistic regression. 2002.

Rémi Cuingnet, Charlotte Rosso, Marie Chupin, Stéphane Lehéricy, Didier Dormont, Habib Benali, Yves Samson, and Olivier Colliot. Spatial regularization of svm for the detection of diffusion alterations associated with stroke outcome. *Medical image analysis*, 15(5):729–737, 2011.

Shugufta Fatima and B Srinivasu. Text document categorization using support vector machine. *International Research Journal of Engineering and Technology (IRJET)*, 4(2):141–147, 2017.

GitHub. URL `https://github.com/vostrova/Cross-lingual-sentiment-analysis`.

Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N project report, Stanford*, 1(12):2009, 2009.

Hakob Grigoryan. Stock market trend prediction using support vector machines and variable selection methods. In *2017 International Conference on Applied Mathematics, Modelling and Statistics Application (AMMSA 2017)*. Atlantis Press, 2017.

Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.

A Jain and RD Mishra. Text categorization: by combining naïve bayes and modified maximum entropy classifiers. *Int. J. Adv. Electron. Comput. Sci*, pages 122–126, 2016.

Dan Jurafsky and James H Martin. Speech and language processing. vol. 3, 2014.

Dmitry Kan. Rule-based approach to sentiment analysis at romip 2011. 2012. URL Availablefrom:http://www.dialog-21.ru/digests/dialog2012/materials/pdf/Kan.pdf.

Mostafa Karamibekr and Ali A Ghorbani. Sentiment analysis of social issues. In *2012 International Conference on Social Informatics*, pages 215–221. IEEE, 2012.

Aurangzeb Khan, Baharum Baharudin, Lam Hong Lee, and Khairullah Khan. A review of machine learning algorithms for text-documents classification. *Journal of advances in information technology*, 1(1):4–20, 2010.

Vishal Kharde, Prof Sonawane, et al. Sentiment analysis of twitter data: a survey of techniques. *arXiv preprint arXiv:1601.06971*, 2016.

Mikhail Korobov. Morphological analyzer and generator for russian and ukrainian languages. In Mikhail Yu. Khachay, Natalia Konstantinova, Alexander Panchenko, Dmitry I. Ignatov, and Valeri G. Labunets, editors, *Analysis of Images, Social Networks and Texts*, volume 542 of *Communications in Computer and Information Science*, pages 320–332. Springer International Publishing, 2015. ISBN 978-3-319-26122-5. doi: 10.1007/978-3-319-26123-2_31. URL http://dx.doi.org/10.1007/978-3-319-26123-2_31.

LinearSVC. URL https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html#sklearn.svm.LinearSVC.

Natalia V Loukachevitch and Yuliya Rubtsova. Entity-oriented sentiment analysis of tweets: Results and problems. In *Proceedings of the International Conference DAMDID/RCDL-2015.—Obninsk*, pages 499–507, 2015.

Melvin Earl Maron. Automatic indexing: an experimental inquiry. *Journal of the ACM (JACM)*, 8(3):404–417, 1961.

Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4): 115–133, 1943.

Nishantha Medagoda, Subana Shanmuganathan, and Jacqueline Whalley. A comparative analysis of opinion mining and sentiment classification in non-english languages. In *2013 International Conference on Advances in ICT for Emerging Regions (ICTer)*, pages 144–148. IEEE, 2013.

Gilad Mishne, Na<to be read again>talie S Glance, et al. Predicting movie sales from blogger sentiment. In *AAAI spring symposium: computational approaches to analyzing weblogs*, pages 155–158, 2006.

Saeedeh Momtazi. Fine-grained german sentiment analysis on social media. In *LREC*, pages 1215–1220. Citeseer, 2012.

Joaqun Padilla Montani. Tuwienkbs at germeval 2018: German abusive tweet detection. In *14th Conference on Natural Language Processing KONVENS*, volume 2018, page 45, 2018.

Tony Mullen and Robert Malouf. A preliminary investigation into sentiment analysis of informal political discourse. In *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, pages 159–162, 2006.

NLTK Naive Bayes. URL `https://www.nltk.org/api/nltk.classify.html#nltk.classify.naivebayes.NaiveBayesClassifier`.

NLTK Sentiment Analyzer. URL `https://www.nltk.org/api/nltk.sentiment.html#module-nltk.sentiment.sentiment_analyzer`.

Supriya Pahwa and Deepak Sinwar. Comparison of various kernels of support vector machine. *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, 3(VII), 2015.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the*

ACL-02 conference on Empirical methods in natural language processing-Volume 10, pages 79–86. Association for Computational Linguistics, 2002.

Hitesh Parmar, Sanjay Bhanderi, and Glory Shah. Sentiment mining of movie reviews using random forest with tuned hyperparameters. In *International Conference on Information Science. Kerala*, 2014.

AG Pazelskaya and AN Solovyev. A method of sentiment analysis in russian texts. In *Proceedings of the Dialog 2011 the 17th International Conference On Computational Linguistics, Moscow region, Russia*, 2011.

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. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Raziur Rahman, Saugato Rahman Dhruba, Souparno Ghosh, and Ranadip Pal. Functional random forest with applications in dose-response predictions. *Scientific reports*, 9(1):1–14, 2019.

Irina Rish et al. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001.

Tong R.M. An operational system for detecting and tracking opinions in on-line discussion. *Working Notes of the ACM SIGIR 2001 Workshop on Operational Text Classification (pp. 1-6). New York, NY: ACM.*, 2001.

Y Rubtsova. Constructing a corpus for sentiment classification training. *Software and Systems*, (109):72–78, 2015.

RuTweetCorp. URL `http://study.mokoron.com/`.

Valery Rybakov and Alexey Malafeev. Aspect-based sentiment analysis of russian hotel. 2018.

Nandini Sengupta, Md Sahidullah, and Goutam Saha. Lung sound classification using cepstral-based statistical features. *Computers in biology and medicine*, 75:118–129, 2016.

Gayane Shalunts, Gerhard Backfried, and Nicolas Commeignes. The impact of machine translation on sentiment analysis. *Data Analytics*, 63:51–56, 2016.

Sergey Smetanin and Mikhail Komarov. Sentiment analysis of product reviews in russian using convolutional neural networks. In *2019 IEEE 21st Conference on Business Informatics (CBI)*, volume 1, pages 482–486. IEEE, 2019.

spaCy. URL `https://spacy.io/models`.

Sarah VanWagenen and Katya Pertsova. Asymmetries in priming of verbal and nominal inflectional affixes in russian. *UCLA working papers in linguistics: Connectedness. Papers by and for Sarah VanWagenen. Los Angeles, CA: University of California Los Angeles*, 2014.

Dirk von Grünigen, Fernando Benites, Pius von Däniken, Mark Cieliebak, Ralf Grubenmann, and AG SpinningBytes. spmmmp at germeval 2018 shared task: Classification of offensive content in tweets using convolutional neural networks and gated recurrent units. In *14th Conference on Natural Language Processing KONVENS 2018*, page 130, 2018.

Gregor Wiedemann, Eugen Ruppert, Raghav Jindal, and Chris Biemann. Transfer learning from lda to bilstm-cnn for offensive language detection in twitter. *arXiv preprint arXiv:1811.02906*, 2018.

Michael Wiegand, Melanie Siegel, and Josef Ruppenhofer. Overview of the germeval 2018 shared task on the identification of offensive language. 2018.

Jian Xi, Michael Spranger, and Dirk Labudde. Cnn-based offensive language detection. In *14th Conference on Natural Language Processing KONVENS 2018*, page 125, 2018.

Nafissa Yussupova, Diana Bogdanova, and Maxim Boyko. Applying of sentiment analysis for texts in russian based on machine learning approach. In *Proceedings of Second International Conference on Advances in Information Mining and Management*, pages 8–14, 2012.