Sentiment analysis on German articles: Comparison on different tools for creating labels on a dataset

Milena Gjorgjievska Perusheska

Faculty of Computer Science and Engineering
Ss. Cyril and Methodius University
Rugjer Boshkovikj 16, Skopje, Republic of North Macedonia
milena.gjorgjievska-perusheska@students.finki.ukim.mk

Abstract

The internet became a place where the freedom of media turned into hate speech together with improper information that can contribute to revolt about the ongoing life. On the other side, nowadays, technology offers many tools for analyzing the polarity of that kind of expression that impacts society. While a great deal of work has devoted to examining sentiment for the English language, significantly less effort has been placed into giving exact classification to the German language. This paper examines the capability of the existing tools and libraries to determine the impact of the words used in articles on society. Sentiment analysis tools use natural language processing (NLP) to analyze online conversations and determine deeper context - positive, negative, neutral. These tools mimic our brains, to a greater or lesser extent, allowing us to monitor the sentiment behind online content. The results obtained represent my approach to create sentiment labels to an unlabeled dataset that contains new articles and analyze the trained model outputs.

1 Introduction

News continues to contribute the way people understand the situations that are happening, many newspapers at least want to give an impression of objectivity so that journalists will often refrain from using positive or negative vocabulary. With the emergence of the Internet, web and mobile technologies, people have changed their way of consuming news. Traditional newspapers and magazines have been replaced by virtual versions like online news. That kind of websites have developed effective strategies to draw attention.

Online news expresses opinions regarding news entities, which may comprise of people, places or even things, while reporting on events that have recently occurred. For this reason, interactive emotion rating services are offered by various channels of several news websites, so that news can be positive, negative or neutral. They may resort to other means to express their opinion, such as embedding statements in a more complex discourse or argument structure, they may omit some facts and highlight others, they may quote other persons who say what they feel, etc. Automatically identifying sentiment that is not expressed lexically is rather difficult, but lexically expressed opinion can be found in news texts, even if it is less frequent than in product or film reviews. [1]

Another difference between reviews and news is that reviews frequently are about a relatively concrete object while news articles may span larger subject domains, more complex event descriptions and a whole range of targets. The sentiment value towards a person may be negative even if this person is attempting to act positively in the event. For these reasons, the focus is on dividing the opinion and have attempted to separate positive, neutral and negative sentiment from good and bad news.

Due to grammatical differences between the English and the German language, a classifier might be effective on an English dataset, but not as effective on a German dataset. The German language has a higher inflexion and long compound words are quite common compared to the English language. One would need to evaluate a classifier on multiple German datasets to get a sense of its effectiveness. [2]

Sentiment Analysis or Opinion Mining [3] is a way of finding out the polarity or strength of the opinion that is expressed in written text, in the case of this paper – a news article.

But what happens if our data in unlabeled? We can try crating those labels by tools that are offered online and often used for sentiment analysis. Trying to find labels and train a model, as well as getting results is an object of interest in this article. The main idea of this paper is to use tools so that from unlabeled data, supervised learning based on classification can be proceed.

The following section gives an overview of major research approaches to sentiment analysis. Brief description of the process taken for sentiment analysis is presented in Section 3. In Section 4, a discussion of the results obtained from the model for sentiment is presented, as well as analysis and summarization of the results of the independent study conducted to assess its effectiveness. Finally, there is a conclusion of the paper in Section 5.

2 Related work

Many researchers have contributed in news sentiment analysis using different approaches. Opinion mining, in this context, aims therefore at extracting and analyzing judgements on various aspects of given products. A brief discussion on the work done previously on sentiment analysis is provided in this section.

Most of the sentiment extraction methods have been developed for the English language. Yet, since English and German are closely related languages when compared grammatically, many preprocessing and feature extraction steps, like stop-word removal or creating n-grams can be applied to German with very little adaptations. The free placement of the negation word in the German language versus the fixed placement in English is one of these differences which have an effect on sentiment analysis. In the English language the negation word not' can only appear after the verb, for example, "I do not like this song" while in German the position of the negation word is not defined: "Mir gefällt nicht dieses Lied" versus "Mir gefällt dieses Lied nicht" – different word ordering but in both cases the same meaning. [2] Indeed, knowledge of the language, that the sentiment analysis is performed on, is essential. The person that is analyzing the data needs to be aware of the correctness of the results obtained, in the end of the process, so that further analysis can be done on reshaping and fitting the results.

[1] and [3] extend views on the analysis of sentiment in newspapers and how they affect society. Different machine learning models contribute to results that can use to regulate what is shared online for divergent purposes.

In [4], [5] and [6] there is explained how tools that are preprogrammed for sentiment analysis work and how they can help if the labels are missing, identical or random.

3 Methods

3.1 Dataset

The dataset of news articles was available online at [7] divided by categories, but the articles were not prelabeled with suitable sentiment values. The dataset counts 10273 articles, from which for evaluation purposes, 80% are randomly chosen for training, and the remaining 20% for testing, using Multinomial Bayes and Random Forest; and 90% for training and 10% for testing using Support Vector Machine (SVM).

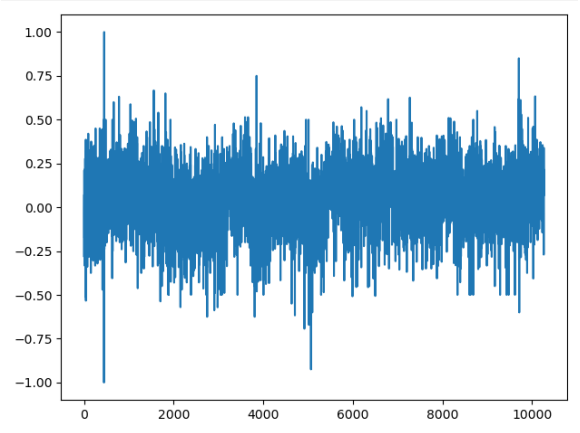
3.2 Creating sentiment labels

For the purpose of the dataset, two ways of calculating the sentiment and appending to each article were tried. The first was given by the following formula:

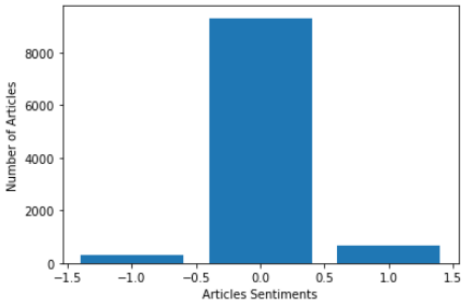
$$\text{Sentiment} = \frac{\#\text{ positive terms} - \#\text{ negative terms}}{\#\text{ all terms}} \quad (1)$$

Each article was tokenized, stemmed and cleaned from stop-words, then searching for the word from two lists with positive and negative polarity bearing words in German, which are available online at [8]. The sentiments obtained were all positive numbers which if proceeded, will give low results, as well as unbalanced data, which is a result of a big amount of same weighted data. Also, the process was time-consuming and I would not recommend it for further usage.

The second approach was with the tool called textblob-de, a German language support for TextBlob, which is a Python library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more [9] [10]. By passing an article to the processing method, there is an ongoing sentiment analysis which gives the polarity of the processed text in the range of [-1,1]. In order to be sure that the tool is calculating the sentiment in a right way, I have chosen random articles, read it and decided for the polarity. After that a comparison was made, with a conclusion that the sentiment that is produced by textblob-de matches the sentiment that can be perceived by a reader. The calculation gives the following statistic about the data (x-axis) and the sentiment (y-axis):



From this, it can be concluded that the data will be neutral in the range from [-0.25, 0.25]. When divided into negative, neutral and positive the data will have this representation:



3.3 Articles processing

After preprocessing, the statistical technique known as Term Frequency-Inverse Document Frequency (TF-IDF) has been used. In TF-IDF term frequency is counted. According to this technique words that occur frequently in a document are considered important and a weight is given to these words. Using TF-IDF important words or terms in a document were identified and assigned a weightage according to the occurrence of various words in the news article.
Other vectorization technique that is used is CountVectorizer which is used to convert a collection of text documents to a vector of term/token counts. It also enables the pre-processing of text data prior to generating the vector representation. This functionality makes it a highly flexible feature representation module for text.

3.4 Model training

The models that are trained for this purpose are three different classifiers: Multinomial Bayes, SVM and Random Forest. In order to train and evaluate the ensemble, each dataset was split into a training and a testing set.

4 Discussion of results

The further work will compare the trained models between the two vectorizers and the three models mentioned in the previous section.
The following table represents the results that were provided by Multinomial Bayes using both vectorizers:

TF-IDF MultinomialBayes				CountVectorizer MultinomialBayes			
Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
0.90	0.82	0.90	0.86	0.90	0.82	0.90	0.86

```
predicted
```

```
array([0, 0, 0, ..., 0, 0, 0])
```

```
predicted
```

```
array([0, 0, 0, ..., 0, 0, 0])
```

The predicted results obtained can show that the prediction is undergoing a problem, most possible the problem of unbalanced data, because every predicted value is 0, in this case neutral.
Next, the results that are presented are an output from training and testing on SVM:

```
Results for SVC(kernel=linear)
Training time: 96.472117s; Prediction time: 9.091183s
positive:  {'precision': 0.0, 'recall': 0.0, 'f1-score': 0.0, 'support': 69}
negative:  {'precision': 0.0, 'recall': 0.0, 'f1-score': 0.0, 'support': 29}
neutral:  {'precision': 0.9046692607003891, 'recall': 1.0, 'f1-score': 0.949948
9274770174, 'support': 930}
```

Another example where the unbalanced data can be spotted, is represented above, because the data is weighted to the neutral sentiment and the model is not actually learning from.
At last, Random Forest is trained and tested:

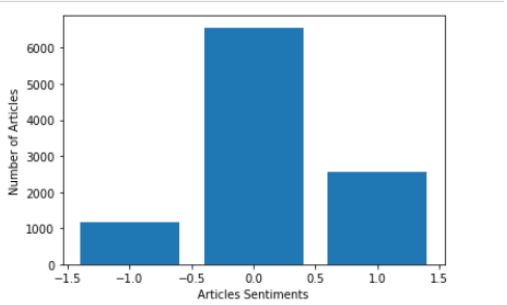
```
              precision    recall  f1-score   support

          -1       0.00      0.00      0.00        69
           0       0.91      1.00      0.95      1865
           1       0.00      0.00      0.00       121

    accuracy                           0.91      2055
   macro avg       0.30      0.33      0.32      2055
weighted avg       0.82      0.91      0.86      2055

0.9065693430656935
```

From the three models above, the main conclusion is the same – there is an appearance of unbalanced data. So, in this case there are three ways that can be adapted in order to get some improvements of the given results. Firstly, changing the range of the sentiment to be [-0.1, 0.1] as negative, neutral and positive. The data has the shape:



In this case, for Multinomial Bayes:

TF-IDF				CountVectorizer			
MultinomialBayes				MultinomialBayes			
Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
0.64	0.41	0.64	0.5	0.6	0.82	0.81	0.82

```
predicted2
array([0, 0, 0, ..., 0, 0, 0])
```

```
predicted1
array([ 0,  0, -1, ...,  0,  0,  0])
```

From the obtained predictions, it can be concluded that for every 100 articles there is 60% of right classification when using Multinomial Bayes with CountVectorizer.

For SVM the results, having again the problem of unbalanced data, are the following:

```
Results for SVC(kernel=linear)
Training time: 300.621496s; Prediction time: 29.352942s
positive:  {'precision': 0.0, 'recall': 0.0, 'f1-score': 0.0, 'support': 69}
negative:  {'precision': 0.0, 'recall': 0.0, 'f1-score': 0.0, 'support': 29}
neutral:  {'precision': 0.9046692607003891, 'recall': 1.0, 'f1-score': 0.949948
9274770174, 'support': 930}
```

For Random Forest, again with unbalanced data:

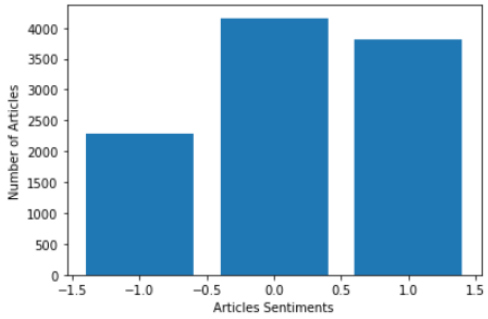
```
              precision    recall  f1-score   support

          -1       0.00      0.00      0.00       251
           0       0.64      1.00      0.78      1313
           1       0.00      0.00      0.00       491

    accuracy                           0.64      2055
   macro avg       0.21      0.33      0.26      2055
weighted avg       0.41      0.64      0.50      2055

0.6374695863746959
```

Secondly, because the problem was not solved with changing the range, a try with the range (-0.0, 0.0) is proceeded for negative, neutral and positive:



For Multinomial Bayes:

TF-IDF				CountVectorizer			
MultinomialBayes				MultinomialBayes			
Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
0.4	0.3	0.4	0.24	0.4	0.35	0.4	0.35

```
predicted01
array([0, 0, 0, ..., 0, 0, 0])
```

```
predicted0
array([ 1,  1, -1, ...,  0,  1,  0])
```

Again, it can be seen from the obtained results that every 100 articles are correctly classified with 40% of accuracy, which is less than the previous try with 60%.

Next, for SVM, there are some results, which show that out of all the articles the analyzer predicted as positive/negative/neutral, 38%/13%/42% of them were actually positive/negative/neutral articles.

```
Results for SVC(kernel=linear)
Training time: 183.080223s; Prediction time: 15.943849s
positive:  {'precision': 0.38317757009345793, 'recall': 0.42597402597402595, 'f1-score': 0.4034440344403444, 'support': 385}
negative:  {'precision': 0.13333333333333333, 'recall': 0.008888888888888889, 'f1-score': 0.016666666666666666, 'support': 225}
neutral:  {'precision': 0.4188034188034188, 'recall': 0.5861244019138756, 'f1-score': 0.4885343968095712, 'support': 418}
```

Finally, for this improvement, for Random Forest the results are the following, with overall accuracy of 39%:

```
              precision    recall  f1-score   support

          -1       0.24      0.02      0.03       453
           0       0.40      0.70      0.51       832
           1       0.37      0.26      0.31       770

    accuracy                           0.39      2055
   macro avg       0.33      0.33      0.28      2055
weighted avg       0.35      0.39      0.33      2055

0.3873479318734793
```

The third way of handling with the unbalanced data is sampling the data, which means choosing random n pairs of (article, sentiment) and pass them to the models. When previewing the data representation, it can be again seen the problem of domination of neutral articles – unbalanced data:

```
Counter({0: 64, 1: 22, -1: 14}), n=100

Counter({0: 325, 1: 118, -1: 57 }), n=500

Counter({0: 1636, 1: 535,  -1: 329}), n=2500

Counter({0: 3300, 1: 1072, -1: 628}) n=5000
```

As a summarization of this section, it can be concluded that Multinomial Bayes with Count Vectorizer and sentiment of the article in the range of [-0.1, 0.1] gives the most suitable results. In my opinion, the obtained results were expected, because of labelling an unlabeled data with a tool which is one of the small variety of choices for the German language. Most of the weak evaluation metrics are obtained as a result of having unbalanced data. Using a tool like texblob-de when you are a beginner, can help you experiment in many ways, not worrying if the data is prelabeled or not. [11] However, the performance of all analyzers used may be improved by feature engineering and cleaning text data first before analyzing them.

5 Conclusion

The present research explores an approach to the problem of obtaining the sentiment of articles and how to use tools for creating labels if the dataset is unlabeled. Following the results of previous research in this field, different techniques and methods examined. TextBlob, actually provided a very easy interface which helped creating the labels and supported further processing of the dataset. Working with NLTK, textblob-de is a tool that offers support for German language, which based on results from this

kind of researches, as well as lexicons, can be optimized in order to give better results in the future. German is a tricky language and we need more good quality libraries for the preprocessing and text analysis. In order to have a good intuition, in which way should a Machine learning project that is working with foreign language go, you need to truly understand the data you are working with.

References

[1] Sentiment Analysis in the News, Conference Paper, https://www.researchgate.net/publication/220746038_Sentiment_Analysis_in_the_News

[2] Sentiment Analysis for German Facebook Pages, https://www.researchgate.net/publication/304020973_Sentiment_Analysis_for_German_Facebook_Pages

[3] Sentiment Analysis of News Articles: A Lexicon based Approach, Conference Paper, https://www.researchgate.net/publication/330880816_Sentiment_Analysis_of_News_Articles_A_Lexicon_based_Approach

[4] Natural Language Processing for beginners: Using TextBlob, https://www.analyticsvidhya.com/blog2018/02/natural-language-processing-for-beginners-using-textblob/

[5] Evaluating Unsupervised Sentiment Analysis Tools Using Labeled Data, https://heartbeat.fritz.ai/evaluating unsupervised-sentiment-analysis-tools-using-labeled-data-8d4bb1336cee

[6] Stimmungsanalyse (Sentiment Analysis) auf deutsch mit Python, https://machinelearningblog.de/2019/06/03/stimmungsanalyse-sentiment-analysis-auf-deutsch-mit-python/

[7] Ten Thousand German News Articles Dataset, https://tblock.github.io/10kGNAD/

[8] German Sentiment Analysis, https://www.kaggle.com/rtatman/german-sentiment analysis-toolkit?select=SentiWS_v1.8c_Negative.txt

[9] TextBlob: Simplified Text Processing, https://textblob.readthedocs.io/en/dev/

[10] textblob-de Documentation, https://readthedocs.org/projects/textblob-de/downloads/pdf/latest/#:~:text=TextBlob%20is%20a%20Python%20(2,classification%2C%20translation%2C%20and%20more

[11] NLP for beginners using textblob, https://www.analyticsvidhya.com/blog/2018/02/natural-language-processing-for-beginners-using-textblob/