Towards Sentiment Analysis on German Literature

Albin Zehe, Martin Becker, Fotis Jannidis, and Andreas Hotho

University of Würzburg, 97074 Würzburg, Germany
{zehe,becker,hotho}@informatik.uni-wuerzburg.de
fotis.jannidis@uni-wuerzburg.de

Abstract Sentiment Analysis is a Natural Language Processing-task that is relevant in a number of contexts, including the analysis of literature. We report on ongoing research towards enabling, for the first time, sentence-level Sentiment Analysis in the domain of German novels. We create a labelled dataset from sentences extracted from German novels and, by adapting existing sentiment classifiers, reach promising F1-scores of 0.67 for binary polarity classification.

Und sie lebten glücklich bis ans Ende ihrer Tage. (German fairy tales)

1 Introduction and Related Work

The above quote is a common ending in German fairy tales. If you can not tell whether or not this ending is happy, you have already come across the problem this paper is concerned with: For Sentiment Analysis (SA), a task in Natural Language Processing, there exists a multitude of solutions tailored to specific datasets of English texts, but few for other languages - and none for our domain: German novels. We aim to change this, as SA can help to achieve a very interesting goal in the context of literature: a computer-readable representation of a story. One viable approach for story representation is the use of *sentiment trajectories* that describe the emotional state over the course of a novel. For example, a wedding could be recognised in such a trajectory by a spike in positive emotions, while the death of a protagonist would be accompanied by negative words. Such representations have previously been used in [2, 4, 12]. Similarly, [22] use sentiment trajectories to recognise one core element of a story's plot: the presence or absence of a Happy Ending.

Most of these previously used representations rely on relatively crude SA, using only word-level sentiment. In order to take into account negation, intensification etc., more complex systems have to be used. Such systems have been a field of active research for a while. For example, [13] proposes an SVM-based classifier relying on bag-of-words and syntactic features, as well as some manually crafted emotion features. Recently, Neural Network-based approaches have become increasingly popular, redefining the state-of-the-art in SA. One milestone was the introduction of RNTN [20] along with the Stanford Sentiment Treebank (SST), which has since been used as a standard evaluation dataset. RNTN reached an accuracy of 85.4 % on the SST. [5] and subsequently [6] proposed systems based on Convolutional Neural Networks, reaching accuracies of up to 88.1 %. Paragraph Vectors [7] have also been reported to yield accuracies of up to 87.8 % on the SST. Very recently, a system using a fundamentally different approach with unsupervised pre-training has achieved accuracies of 91.9 % on the SST [17].

While working very well, these systems have only been used on English texts. In contrast to that, our goal is to introduce sophisticated methods for SA into the domain of German literature. To this end, we verify results from previously published approaches and evaluate their performance in our domain. To make this possible, we create the German Novel Dataset (GND), a set of sentences extracted from novels in German language, labelling these sentences with sentiment information using crowdsourcing. We adapt three state-of-the-art SA methods [6,7,13] forming a good basis for adaptation to German data, as they rely on a straightforward and understandable model. [17] is currently unsuitable for our evaluation, as its pre-training does not transfer well to other domains and is too time intensive to retrain. Overall, our work is an important step towards building a repository of advanced methods that can be used for the analysis of German literature.

The remainder of this paper is structured as follows: In Sect. 2, we define Sentiment Analysis and adapt existing approaches to German texts. Sect. 3 describes the English reference datasets and introduces our GND. Sects. 4 and 5 report and discuss our findings. A summary and future work are given in Sect. 6.

2 Sentiment Classification

Generally, Sentiment Analysis refers to the task of assigning a label, called *polarity*, to a segment of text, describing whether it induces positive, negative or neutral feelings in a human reader. In this work, SA is defined as a sentence classification task, enabling classifiers to account for the effect of negation etc. Assume a corpus $C \subset S \times L$ of sentences $s \in S$ and polarity labels $l \in L$. For example, $c = (s, l) = (\text{"I love you."}, 1)$ represents a sentence s with a positive polarity l. We perform two classification tasks, distinguished by the set of possible polarities: (a) Binary classification: $L_{\text{bin}} = \{-1, 1\}$ and (b) Ternary classification: $L_{\text{ter}} = \{-1, 0, 1\}$.[1] A classifier of any kind is trained to predict the correct label given a sentence, that is, to learn a function $f\colon S \to L$ with $f(s) = l$.

We compare two different classifiers for our SA task: Support Vector Machines (SVMs) [3] and Convolutional Neural Networks (CNNs) [8]. To represent sentences for the SVM, we use two different feature generation methods, the *NRC Representation* [13] and *Paragraph Vector* [7]. For the CNN, we use the model from [6], which we refer to as *S-CNN*.

SVM Classifier. In this paragraph, we give a short overview of the sentence representations used as input for the SVM.

NRC Representation. For the NRC Representation, a sentence is represented as the concatenation of different kinds of n-gram and syntactic features in

[1] Ternary labels are transformed into binary labels by omission of the neutral class (0).

combination with a set of sentiment features constructed manually from the EmoLex [14]. We use the TreeTagger [19] to get part-of-speech-tags required for the syntactic features. The representation employs a basic negation detection relying on a set of English negation words. In order to be applicable to German text, this list had to be translated.[2] Lemmatisation was not employed in n-gram generation, as it did not improve results, but was used for the lookup of words in the sentiment lexicon. For a full description of the features, we refer to [13]. Note that we do not use the full set of features described there, as we consider the following features to be irrelevant outside the context of tweets: all-caps words, hashtags, multiple punctuation marks, emoticons and elongated words. Also, using Brown Clusters has been shown to be ineffective in [13].

Paragraph Vector. The Paragraph Vector-Framework extends the word2vec-Framework [10, 11] to create an embedding of a piece of text, for example a sentence, in a low-dimensional space. Following [7], we directly use this embedding as a feature vector for SA with Logistic Regression or other suitable classifiers/regressors. We use the implementation of Paragraph Vector provided by gensim [18] to train sentence embeddings on our GNC (see Sect. 3).

S-CNN. The S-CNN is a sentence classification method based on a Convolutional Neural Network. We use the variant referred to as "cnn-nonstatic" in [6]. While [6] uses word2vec embeddings pre-trained on a very large corpus of English news articles[3], to our knowledge, no embedding trained on such a large corpus is available for German. Again, we used gensim to train 300-dimensional word2vec embeddings on our corpus of German novels. An implementation of S-CNN is provided by the original author.[4] Only small changes had to be made to the code to make it compatible with German text, specifically the inclusion of the character "ß" and German umlauts to the regular expression used for pre-processing.

3 Datasets

In this section, we describe the datasets we use in our experiments. We use English reference corpora to verify results of the existing algorithms. Additionally, we extract a dataset of sentences from German novels and label it by crowdsourcing to evaluate the classifiers in our domain.

English SA Datasets. In order to validate the selected approaches as well as existing results, we use two standard English datasets. The first one is a dataset of tweets [15] used in [13]. It is downloadable via Twitter's API. Some differences may arise depending on the time of the download, but the distribution over the polarity labels was mostly unchanged from that in [13] (14 % negative, 49 % neutral and 37 % negative for the training set of 6128 samples.). The second dataset is the Stanford Sentiment Treebank (SST) [20].

[2] We use the words "nicht" (not), "kein" (no), "ohne" (without), "nie" (never), "niemals" (never), "nirgends" (nowhere), "niemand" (nobody), and "keiner" (nobody) as negation markers.

[3] `https://code.google.com/archive/p/word2vec/`

[4] `https://github.com/yoonkim/CNN_sentence`

German Novel Dataset and Corpus. The *German Novel Dataset* (GND) was generated using a crowdsourcing approach. It contains 270 labelled sentences extracted from our German Novel Corpus (GNC) of over 600 novels in German language from the TextGrid Digital Library[5]. Of these sentences, 89 are labelled as "negative", 124 as "neutral" and 57 as "positive". The dataset is released along with this paper.[6]

Labelling Process. To create the GND, we extracted all sentences from the GNC containing at least three words and no more than 30 words or 1500 characters. These sentences were then ranked using the ratio $r = {}^{e^n}/_{w}$, where e is the number of words in a sentence associated with emotions by the EmoLex [14] and w is the total number of words in the sentence. We evaluated $n \in \{1, 2, 3\}$ and selected $n = 3$ because it led to sentences that were emotional, but did not consist only of emotional words.[7] We selected the 210 highest ranked sentences and 90 additional sentences by random choice, resulting in 300 sentences for annotation. We developed a web interface for the annotation process.[8] The sentences were annotated for ternary polarity and the eight basic emotions defined in [16] using *Microworkers* and *CrowdFlower*.[9] An Inter Annotator Agreement was calculated and annotators were dropped, including all of their annotations, if they failed to meet a defined threshold. We kept only sentences with at least five annotations after filtering. The polarity of a training sample was selected by majority vote, while a sentence was marked as conveying an emotion if at least two annotators selected the emotion. For details on the annotation and selection process, see [21].

4 Results

Here, we briefly report the findings on the English datasets and give a more detailed description of our results on the German Novel Dataset (GND).

Validating Classifiers on English Datasets. To validate our implementations, we evaluated all classifiers on the Twitter Dataset and the SST, reproducing the results from [13] and [20]. On both datasets, the S-CNN gave better results (F1-score of 0.86 for binary classification), but was much more sensitive to parameter selection. We were not able to reproduce the results from [7], using Paragraph Vectors as input for an SVM or Logistic Regression. Note that others were also unable to reproduce them and one author of [7] considers them to be invalid [9].

Evaluating Classifiers. All classifiers were evaluated on the GND and large parameter studies as well as some feature analysis were performed.

Results Using an SVM. We start by describing the findings from the SVM-based methods. For both the NRC features and the Paragraph Vectors, we used a linear

[5] `https://textgrid.de/digitale-bibliothek`

[6] `https://www.dmir.org/datasets/german_novel_dataset`

[7] We also evaluated other selection schemes, but found that random selection yielded too many unemotional sentences, while $r = e$ preferred very long ones.

[8] Available on `http://dmir.org/senticrowd/senticrowd`. Login is possible with both "Microworkers-ID" and "Kampagnen-ID" set to "demo" in the upper form.

[9] `http://www.microworkers.com` and `https://www.crowdflower.com`

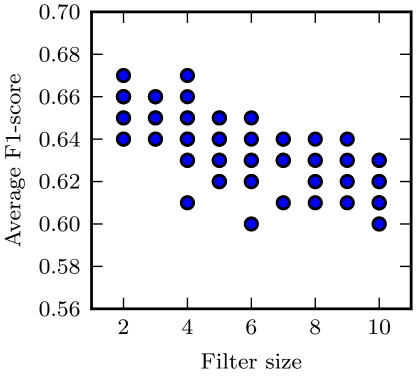

Figure 1: Dependency of the S-CNN on the filter size for binary polarity classification on the GND. A datapoint corresponds to a specific combination of the three tuned hyper-parameters.

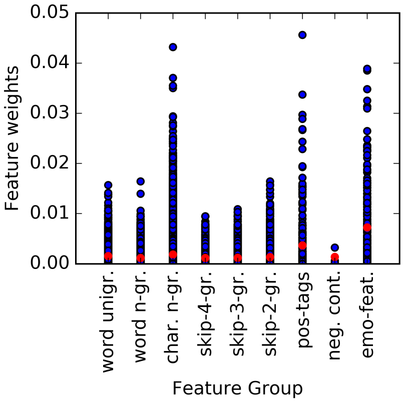

Figure 2: Absolute weights assigned to feature groups by a linear SVM. Blue dots are individual feature weights, red dots group averages. Weights for all classes plotted together.

kernel SVM. For the Paragraph Vectors, we additionally employed an RBF-kernel. To optimise parameters, we did a grid search over 20 values for C evenly spaced on the log-scale from 10^{-2} to 10^{2} and (where applicable) $\gamma \in \{0.01, 1, 10\}$. We report micro-averaged F1-scores[10] over all classes in the respective task. All scores are calculated as the average over 10 independent runs of 10-fold cross-validation. There were no major differences between runs (usually much less than 5 %).

We reached an F1-score of 0.43 for ternary and 0.67 for binary classification respectively using a linear SVM trained on NRC features. Using Paragraph Vectors as input to an SVM led to similar results. While these scores are not on par with those on English datasets, they are far above those a majority baseline would achieve (F1-score ≈ 0.5 for binary classification). There was no dependency of the linear SVM on the value for C in the range we searched. Using an RBF kernel made the classifier much more dependent on hyper-parameter selection, but did not improve the results overall.

Results using S-CNN. We use random search [1] to jointly optimise Dropout rate d, number of filters n per filter size and filter size s. We draw about 80 parameter combinations uniformly from $d \in [0.0, 0.5]$, $n \in [100, 500] \subset \mathbb{N}$ and $s \in [2, 10] \subset \mathbb{N}$, as recommended in [23]. After sampling a filter size s, we use $s-1$, s and $s+1$ in parallel for the CNN. The only parameter with clear influence on the results is s. Using smaller values clearly outperformed larger ones, as shown in Fig. 1. Generally, the S-CNN performed comparably or slightly worse than an SVM trained on NRC features, with F1-score up to 0.67 for binary classification.

[10] `http://scikit-learn.org/0.17/modules/generated/sklearn.metrics.f1_score`

5 Discussion

On the German Novel Dataset (GND), all three methods yielded comparable results. However, the SVM trained on NRC features is much less dependent on hyper-parameter selection and requires less time to train than the S-CNN. While training an SVM on Paragraph Vectors is also fast, the training of these embeddings requires much time. We therefore consider the SVM trained on NRC features to be the most suitable classifier for our task at the current state.

To gain some further insight into the relevance of individual features, we plotted the weights assigned to all NRC features by a linear SVM trained on the full GND. Fig. 2 shows the resulting plot comparing the average weight of our feature types during classification. The manually constructed emotion features (*emo-feat.*) obviously are most important to the classification. Similarly, the *pos-tag* features play an important role, which may be due to their ability to capture specific sentence structures. Word n-grams ($n > 1$) and skip-n-grams (except, to some degree, $n = 2$) have only small influence, which is not surprising considering their sparsity in the small GND. Unigrams have some relevance, but are, interestingly, much less important than character-n-grams. We assume this is due to the fact that character n-grams can actually group together different words with the same stem, helping generalisation. Assuming that the filters in the S-CNN capture information similar to n-gram features[11], these findings are consistent with those for the CNN. There, smaller filter sizes (i.e., corresponding to lower order n-grams) performed best.

While the results achieved on the GND are certainly not on par with those on the English datasets, this can most likely be explained by the training set that is at least an order of magnitude smaller than the English sets. The performance being far above that of simple baselines and the interpretability of the feature weights show that the NRC features are able to capture information that is useful for polarity classification on German literary text.

6 Conclusion and Outlook

In this paper, we have presented first steps towards introducing more complex SA methods to the domain of German literature. This is a prerequisite for plot representation and other interesting tasks in the Digital Humanities. To this end, we introduced a unique dataset of sentences extracted from German novels that have been manually labelled with polarity information and basic emotions. Our annotation interface can easily be used to extend this dataset in the future. While the results are not on par with English SA, we have shown that the features and classifiers are generally applicable in our domain and can recognise signals that are useful for SA on German novels.

In future work, we will expand our dataset to enable training of more expressive models, possibly also creating French and Spanish datasets, and introduce domain-specific features to improve classification accuracy.

[11] http://www.wildml.com/2015/11/understanding

References

1. Bergstra, J., Bengio, Y.: Random search for hyper-parameter optimization. JMLR 13, 281–305 (2012)
2. Elsner, M.: Abstract representations of plot struture. LiLT 12 (2015)
3. Joachims, T.: Text categorization with Support Vector Machines: Learning with many relevant features, pp. 137–142. Springer, Berlin, Heidelberg (1998)
4. Jockers, M.L.: A novel method for detecting plot (Jun 2014)
5. Kalchbrenner, N., Grefenstette, E., Blunsom, P.: A convolutional neural network for modelling sentences. In: Proceedings of the 52nd ACL. pp. 212–217
6. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on EMNLP. pp. 1746–1751 (2014)
7. Le, Q.V., Mikolov, T.: Distributed representations of sentences and documents. In: ICML. vol. 14, pp. 1188–1196 (2014)
8. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. In: Proceedings of the IEEE. vol. 86, pp. 2278–2324 (1998)
9. Mesnil, G., Mikolov, T., Ranzato, M., Bengio, Y.: Ensemble of generative and discriminative techniques for sentiment analysis of movie reviews (2014), cite arxiv:1412.5335
10. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space (2013), cite arxiv:1301.3781
11. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NIPS (2013)
12. Mohammad, S.: From once upon a time to happily ever after: Tracking emotions in novels and fairy tales. In: LaTeCH. pp. 105–114. LaTeCH '11, Association for Computational Linguistics, Stroudsburg, PA, USA (2011)
13. Mohammad, S.M., Kiritchenko, S., Zhu, X.: Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. SemEval-2013 (2013), cite arXiv:1308.6242
14. Mohammad, S.M., Turney, P.D.: Crowdsourcing a word–emotion association lexicon. Computational Intelligence 29(3), 436–465 (2013)
15. Nakov, P., Kozareva, Z., Ritter, A., Rosenthal, S., Stoyanov, V., Wilson, T.: Semeval-2013 task 2: Sentiment analysis in twitter (2013)
16. Plutchik, R.: A general psychoevolutionary theory of emotion. Theories of emotion 1, 3–31 (1980)
17. Radford, A., Jozefowicz, R., Sutskever, I.: Learning to generate reviews and discovering sentiment (2017), cite arxiv:1704.01444
18. Řehůřek, R., Sojka, P.: Software Framework for Topic Modelling with Large Corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks. pp. 45–50. ELRA, Valletta, Malta (May 2010)
19. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: New methods in language processing. p. 154 (2013)
20. Socher, R., Perelygin, A., Wu, J.Y., Chuang, J., Manning, C.D., Ng, A.Y., Potts, C., et al.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the EMNLP. vol. 1631, p. 1642 (2013)
21. Zehe, A.: Sentiment Analysis on German Novels. Master's thesis (2017)
22. Zehe, A., Becker, M., Hettinger, L., Hotho, A., Reger, I., Jannidis, F.: Prediction of happy endings in german novels. In: Cellier, P., Charnois, T., Hotho, A., Matwin, S., Moens, M.F., Toussaint, Y. (eds.) DMNLP@PKDD/ECML. pp. 9–16 (Jul 2016)
23. Zhang, Y., Wallace, B.: A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification (2015), cite arxiv:1510.03820