On the relevance of simple language for natural language processing

Anonymous ACL-IJCNLP submission

Abstract

Simple language is a valuable tool for accessibility. We looked at the impact of German simple language in relation to natural language processing (NLP). We looked into two questions, first whether there might be room to specifically gear procedures towards simple language to achieve better results than with methods developed with conventional texts in mind, and second whether applying NLP tools to simple language without training on it has a significant impact on the performance. In both instances, it turned out that simple language differs enough from conventional language, that it warranted a specific approach, to achieve optimal results.

1 Introduction

The concept of simple language (Trescher, 2018) is a solution for creating accessible text for people without the capability to understand complex texts. Its used by news sites to provide valuable information to people regardless of their literacy (Deutschlandfunk, 2021).

In this work, we look at German simple language, or "Leichte Sprache". We are examining the implications simple language has on natural language processing. Simple language is distinct from conventional text, especially so in news context, where the language level is fairly high. Simple language aims to maintain high readability even on complex topics.

For this, we focus on two main experiments. Firstly, we answer the question if the common features used when classifying news articles work at a comparable level when applied to simple language, or if we can tune the features to obtain better results in classifying it. And secondly, is it worth it to train a classifier on simple language specifically, if we intend to use it for simple language, or does training on conventional text give a similar result.

The motivation for this is to figure out if there is a benefit in considering simple language when designing a classifier, or if it is close enough to conventional language to simply treat it as such.

2 Data

We are using datasets we gathered ourselves. We use news articles, as there is plenty of available data in simple language, and it is easily interpretable as a classification task by extracting features from the article and by using the general category as a label. We take our data from nachrichtenleicht.de (Deutschlandfunk, 2021) for simple language and from spiegel.de (SPIEGEL-Gruppe, 2021) for our conventional language. For our supervised training we require labels, but we are able to easily obtain them, as they are provided already, since each article is categorized by the source under "culture", "sport", "news" or for our purposes "politics" and "miscellaneous". We will focus on the former three categories, as that gives us a more clear distinction for each label.

We cleaned the data, formatting each article into a single line, with the document consisting of the headline, followed by a short description of the article, and then the body of the article itself. After cleaning we gathered some metadata, to give a better overview of our dataset, and to be able to judge some of the results accordingly. The result can be seen in Table 1.

Two aspects to point out are the clear and significant difference in the reading ease (Flesch, 1948), indicating that we chose data that is far apart in its style and language level, and is thus fit for researching our questions. The other point to consider is the comparable number of data points per labels, meaning we do not need to pay too much attention to possible problems arising from unbalanced data.

	simple language				normal language			
	culture	sport	politics	**total**	culture	sport	politics	**total**
# of articles	1304	1230	2020	**4554**	2357	2659	2233	**7249**
avg length of article (words)	151	137	174	**157**	93	93	89.8	**91.7**
avg reading ease	64	63	61.5	**62.77**	42	46.3	33	**40.6**
# of tokens	716642				714773			
# of unique words	47602				76126			

Table 1: Overview over the datasets

3 Approach

3.1 General

For implementation, we are using Python, with the packages pandas (pandas development team, 2020), spacy (Honnibal et al., 2020) and sci-kit learn (Pedregosa et al., 2011) as our main sources for premade vectorizers and classifiers. For our evaluation, we observe the standard metrics for our models; accuracy, precision and recall, and the F_1 score. We use these standard metrics since our goal is not to determine the actual quality of our models, but to compare them against one another. Therefore having a broader overview of general performance is enough to indicate a trend.

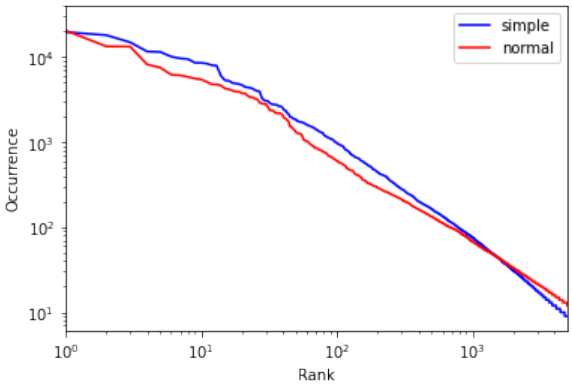

Figure 1: Rank Frequency curve.

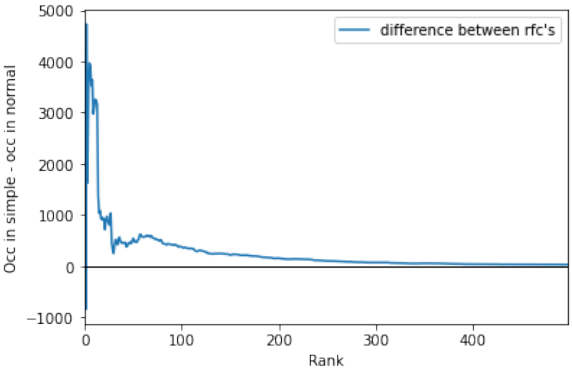

Figure 2: Difference between the two lines

3.2 Which features work best

In this experiment we use a Multilayer Perceptron as our classifier. We use it as it is reasonably fast and easy to train, and still provides a decent standard, being a neural network classifier. For our standard features we will look at TF-IDF vectors and word-embeddings, along with an experimental custom featurizer based on TF-IDF.

Our own feature essentially uses a TF-IDF vectorizer, but we're only including lemmas of nouns, verbs and adjectives. The intuition here is, that in simple language even more than in conventional texts, the rest of the words provide no benefit for labeling. This is supported by our analysis of the data, which shows that for the most commonly used words, simple language has significantly higher frequencies, as shown in Figure 2. For a further look into the plots, check the corresponding notebook.

Thus, the most common words will be even more present, and have even less value for our task than in a normal text, so instead of just diminishing their influence, we are outright removing them.

3.3 Is training on simple data worth it

We will use two very basic methods for feature extraction, simple count vectors and TF-IDF vectors. We use these fairly simple methods here since our focus is mainly on the general question of comparability between conventional and simple language. For this question, we deem simple and easily computable features sufficient.

To classify our data we use a Naive Bayes Classifier (NB), a Decision Tree Classifier (DT) and a Multilayer Perceptron (MLP) (Pedregosa et al., 2011). With this selection we hope to obtain a good overview. We might even get an idea of the degree of difference between the two languages, judging on the performance of the differently sophisticated classifiers.

4 Results

4.1 Which features work best

As expected, the control provided good results already, with the MLP classifier achieving 94% accuracy for simple and 97% for normal language, with F_1 scores over 90 throughout. When applying our modified feature, the results for the normal language stayed the same, with simple language increasing around 1 percentage point. While these results are not conclusive, they do show that our intuition was correct, and it shows promise for future development. It certainly shows, that you can remove large parts of the text for simple language, without losing any performance on the classifier, instead improving it.

4.2 Is training on simple data worth it

When training the classifier with conventional data, in all setups, the scores show a significant decrease when being applied to the simple language as opposed to the test set of conventional language. The difference in accuracy is around 5-7%, with F_1 scores dropping similar amounts. We show a full exemplary evaluation of metrics in Table 2 and Table 3, with the others being of a very similar nature. To properly judge the value of these results, it is important to remember a few facts. The test set was significantly smaller than the size of the simple language data. Another factor to consider is the possibility of overfitting on the conventional texts. The source for all conventional texts was the same, which could have the classifier pick up on that sources specific way of writing. This would not be present in the simple language. It is easy to argue however, that writing style and word choice are naturally less influential for texts written in simple language. A noteworthy property of our results is the noticeable difference between the categories. Not only does ”sport” do better for the test data, it also suffers less when classifying simple language, with the drop in F_1 score being less than for the other categories (see Table 1). To get a better view into our models performances, we gathered a sample of 75 predictions, in the attempt to find hints as to which features might have led to the models success or failure.

4.3 Manual review

When analysing the samples one thing becomes clear. The main reason ”sport” outperforms the other two categories is the clearer separation between the two. There is little overlap between a sport article, and an article about politics. With culture and politics, this is more often that case. This leads to the classifier often mislabeling ”culture” for ”politics” and vice versa, but rarely mixing up ”sport” with any of the two.

5 Discussion and outlook

The results of our experiments show, that the questions we raised in the beginning of the project were valid, and can in fact be answered with a clear yes. Simple language does differ enough from conventional texts for it to be its own domain. It has the possibility for specifically designed and chosen features. When classifying, there is a difference whether the training data contained any simple language or not. It is important to note, that while the performance for simple language worsened when only training on normal texts, the results were still decent. This shows that the connection between the two types of text is certainly relevant, but that simple language has certain special properties that push it just far away enough from conventional language to impact the results. It can only be expected, that features and classifiers that rely even more on individual style and small nuances will struggle more with simple language, as its core idea is simplifying through reducing complexity. In our project, we were able to give a small view into the implications simple language has on NLP, with the clear potential for further questions, that we could not include in this project. The clear challenges for our project were time and availability of data. In our relatively short time frame, setting up struc-

	precision	recall	f1 score	#articles
culture	0.90	0.93	0.91	708
politics	0.94	0.92	0.93	655
sport	0.98	0.96	0.97	812
accuracy		0.94		2175
macro avg	0.94	0.94	0.94	2175
weighted avg	0.94	0.94	0.94	2175

Table 2: MLP classifier, TD-IDF vectors, trained on conventional, tested on conventional

	precision	recall	f1 score	#articles
culture	0.74	0.86	0.80	1304
politics	0.89	0.83	0.86	2020
sport	0.98	0.93	0.96	1230
accuracy		0.87		4554
macro avg	0.87	0.87	0.87	4554
weighted avg	0.87	0.87	0.87	4554

Table 3: MLP classifier, TD-IDF vectors, trained on conventional, tested on simple lanuage

tures like language models or machine translation models would have required a lot larger workforce, with more experience in performing those tasks. For many applications we would have required significantly more data, but with our focus on news articles in German simple language, the sources are sparse to begin with. It would certainly be interesting to look into language models for simple language, as well as machine translation, as the requirements for a text in simple language are vastly different than for a standard text. Transforming the text precisely without altering the content, while simplifying complex issues might be a task that requires more general machine learning approaches, with conventional NLP approaches being targeted at too different topics. Overall, we believe simple language to grow in its use, and thereby forcing itself into the focus of more researchers, looking to improve tools for those who might need them.

6 Conclusion

Simple language as a concept is of relevance when applying NLP procedures. While theoretically compatible with conventional language, it is considerably different, to the point where it warrants its own separate approach. Not accounting for simple language in NLP applications can lead to a decrease in performance, so it is a factor to consider, when analysing the domain. With simple language becoming an increasingly important part of life, mostly in the form of public agencies and news organisations, this will only grow in relevance. We showed this using the example of multi-label classification, a very standard and general task, and with very basic and simple tools, but the results were striking. Alongside that, some small experimentation with adapting features for simple language showed that our tools can be easily and successfully adapted for simple language.

References

Deutschlandfunk. 2021. nachrichtenleicht.de. https://www.nachrichtenleicht.de/. Accessed: 26.05.2021.

Rudolph Flesch. 1948. A new readability yardstick. *Journal of applied psychology*, 32(3):221.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

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

SPIEGEL-Gruppe. 2021. spiegel.de. https://www.spiegel.de/. Accessed: 14.06.2021.

The pandas development team. 2020. pandas-dev/pandas: Pandas.

Hendrik Trescher. 2018. Kognitive beeinträchtigung und barrierefreiheit. *Eine Pilot-Studie. Bad Heilbrunn: Klinkhardt*.