Using SketchEngine and Python to perform text analysis on German's national dialects (specifically the usage of the German language in Austria, Germany, Liechtenstein, and Switzerland)

Muhammad Ahmad
University of Leeds
sc20ma2
@leeds.ac.uk

Tristan Cheah
University of Leeds
sc20tkkc
@leeds.ac.uk

Raphael Nash
University of Leeds
sc20rn
@leeds.ac.uk

Flavio Schuricht
University of Leeds
sc20fs
@leeds.ac.uk

1 Introduction

The data mining community has created numerous resources for English linguistic analysis, for example, The Brown Corpus (Francis & Kučera, 1967) and the ICE International Corpus of English. It has been shown that, in general, the larger the size of the training corpus for a classifier model, the better the classifier's performance (Banko & Brill, 2001).

With this in mind, corpora for other languages with several national dialects could benefit from further attention. This paper describes the collection, preparation and text classification tasks we have performed on four national dialects of the German language: Deutsch, Austrian, Swiss, Liechtensteinisch.

2 Related Work

The University of Leeds has made similar contributions to the research of Arabic dialects. One paper by Alshutayri and Atwell (2017) collected 210,915 tweets from five Arabic dialects - another of their papers did the same broadly across social media (2018) - while a great source for informal text, we preferred to use another one of their methods by collecting newspapers of each national dialect (Alshutayri and Atwell, 2019).

Alshutayri and others have used WEKA as their principal tool for text analytics (Alshutayri et al., 2016). However, for the purpose of Deep Learning, we found Python to be better suited for our task.

3 Problem Definition, Objectives and Requirements

The problem statement is as follows: by collecting reasonably sized sub-corpora, each for four national dialects of German, is it possible to train (a) classifier(s) to distinguish between the dialects we have chosen?

The objectives of this paper are:

1. to collect texts of similar size but of a variety of different contexts, authors and levels of formality for each subcorpus (Section 4),
2. to understand the linguistic features of these dialects to understand what makes each distinct,
3. to sanitise text of non-relevant features like HTML tags (Section 5),
4. to train several classifiers on a large portion of the sub-corpora/'in-sample data'; this portion is referred to as the training set (Section 6), and
5. to test those classifiers on a smaller portion of the sub-corpora (the test set), and against several out-of-sample examples; to evaluate classifiers based on their precision, recall and F1-scores; to determine which model (that we have tested) is best for solving the problem (Section 7).

The requirements are:

1. to ensure that the sub-corpora are representative of a wide variety of contexts, authors and levels of formality in that national dialect.
2. to sanitise text to prevent non-German words/features leaking into the vocabulary.
3. to correctly portion and shuffle the training data so that the model can train on the set with a reduced opportunity for bias.
4. to strictly keep training data and test data separate.

5. to correctly interpret evaluation metrics and form an objective summary of the performance of each model.

4 Data Collection

We decided upon the types of texts we should be collecting and agreed to collect texts from identical keywords, 'immer', 'indes', and 'innen', as well as texts based on key terms specific to our countries of choice but of the same contexts, for example, 'newspaper articles from that country'.

We chose the following categories for such context-based selection: **National Newspapers** (*formal, articles*), **Music** (*informal, lyrics*), and **Universities** (*formal, research*).

Where possible texts were also collected from the categories of 'Literature' (novels, for example).

Our sub-corpora sizes were: **Austria** (*68,261 words*), **Germany** (*65,705 words*), **Liechtenstein** (*67,831 words*), **Switzerland** (*68,460 words*).

We believe that this process satisfies the requirement to ensure texts are representative of a wide variety of contexts.

5 Data Preparation

Before feeding data into the models for training, the data went through the following process:

- Remove all HTML/XML tags
- Remove whitespace at the beginning of a line
- Remove empty lines in the file
- Replace multiple spaces with just one space

For each dialect, the relevant data were then organised into a document-term matrix, giving us n-gram-document frequencies.

Then the document-term matrix was taken for each training example (sanitised and prepared as above), and each n-gram from the training example was considered. If the n-gram existed in a national dialect's n-gram data, its relative frequency was added to a list for that dialect (relative frequency's logarithmic value, to be precise).

6 Modelling

We experimented with different models with increasing complexity, trying: Levenshtein Distance model, Cosine Similarity model, Naive Bayes model, Simple Neural Network model and Ensemble Neural Network model. We studied features at character and word n-gram level, to investigate which level of detail boosts performance.

```
Text length: 487
Labels: ['DE', 'CH', 'AT', 'LI']
Cosine Similarity Prediction: CH
Naive Bayes Prediction: AT
```

Figure 1: Example output during an out-of-sample test. The text being classified was Swiss; the **Cosine Similarity** model classified the text correctly, while the **Naive Bayes** model did not.

6.1 Linguistic Analysis

Before trying the above classifiers, our group researched linguistic similarities and differences between the dialects. One obvious orthographic difference in written language is the "eszett" ("sharp s" or "ß"), only used in Austria and German, whenever the diphthong before the sharp s is pronounced long (Ammon, 2011). As Germany and Austria have a related history there are no grammatical differences, but several variations in the vocabulary (see Table 4 in the appendix). Furthermore, at the Swiss, Austrian and Liechtenstenisch border, the languages begin to mix, according to an interview we conducted with a student from the "Universität Liechtenstein".

6.2 Levenshtein Distance

We thought that there could be slight differences in the letter frequencies used by each dialect which we could exploit. To explore this we worked out frequency counts of each letter by dialect and made bar charts of frequency counts in descending order: we could not see any major differences.

To confirm this observation, we used Levenshtein Distance to measure similarity. For each dialect, we ordered letters by descending order of frequency, giving us a string shorthand for capturing the occurrences of each letter in each dialect corpus, with letters at the start occurring most frequently.

When classifying a new text, we applied the same method to get a similar string, where we worked out the Levenshtein Distance between that string and each dialect's string; the dialect with the shortest Levenshtein Distance was selected.

6.3 Cosine Similarity

For the training of this model, we worked out the frequencies of each **unit** (word or character n-gram) in a frequency table, dividing each of these values by the total number of units in the subcorpus, making a probability table.

When classifying text, the model constructed one vector for each dialect: for each unit in the example text, its probability (relative to the dialect's subcorpus) was appended to the dialect's vector.

Probabilities for each unit in the example text were then calculated relative to the example text itself, producing a vector for the text with the same length of that of each dialect. The dialect whose vector is 'closest to' the example text's vector by the cosine similarity measure was then taken as the class of the text.

6.4 Naive Bayes

For this model, we used the standard scikit-learn classification toolkit and developed a Naive Bayes classifier model. We tested a range of word and n-gram models.

6.5 Neural Network (Ensemble Method)

We performed tests using unseen in-sample data and labelled out-of-sample data (from random websites) which the classifier hadn't seen; we sanitised each test example and input them into each classifier to determine whether their predictions were correct.

We noticed that classifiers generally had a harder time differentiating between Swiss and Liechtensteinich than for Austrian and Deutsch, so we attempted to create a shallow neural network based decision tree that first made the distinction between whether a piece of text belongs to 'Deutsch or Austrian' German category or 'Swiss or Liechtensteinich' German category. Once this distinction was made, one of two different classifiers would run to classify the text to the correct dialect.

7 Evaluation

Out of the classifiers we tried, it seemed that Deep Learning models seemed to perform best: in particular, they seemed to work well when used together.

In general, all models tended to perform worse on out-of-sample data. We believe that this is because they over-fitted on the training data; hence, their performances on in-sample data were better as such data came from the same sample distribution.

7.1 Cosine Similarity

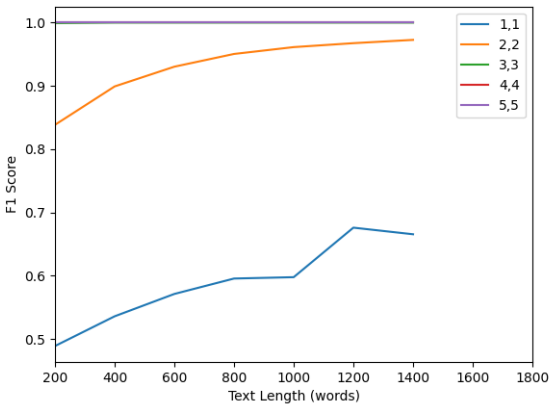

Figure 2: In-sample Cosine Similarity performance for multiple **character** n-gram ranges. 'Text Length' refers to the length of the test example in words. The sidebar shows the 'ngram range', where, for clarity, only n-grams of a fixed size are graphed (unigram, bigram, trigram, etc)

The cosine similarity model performed better for larger text sizes and larger character n-gram sizes, as in Figure 2. It performed very well on in-sample data for larger n-gram sizes, but performed with F1 scores of at most 0.71742 on out-of-sample data (see Table 1).

One unexpected result is that for word n-grams, increasing the n-gram size harms performance. While bigrams and unigrams appear to have the most success, trigrams show a decrease in performance for small text sizes; 4-grams and 5-grams don't do much better than random guessing (assuming a uniform distribution).

n-gram range	Accuracy	F1 Score
1-5	0.76471	0.71742
2-5	0.76471	0.71742
3-5	0.76471	0.71742
4-5	0.76471	0.70752
5-5	0.70588	0.66205

Table 1: Out-of-sample Cosine Similarity scores on character n-grams. Scores given to 5 d.p., and only n-gram configurations for each 'starting n-gram' size with highest F1-score shown.

7.2 Naive Bayes

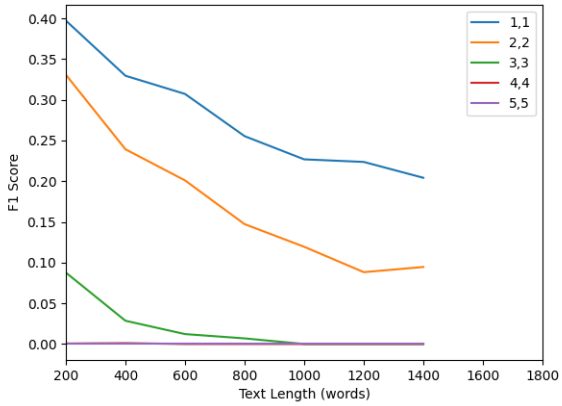

Figure 3: In-sample Naive Bayes performance for multiple **character** n-gram ranges. 'Text Length' refers to the length of the test example in words. The sidebar shows the 'n-gram range', where, for clarity, only n-grams of a fixed size are graphed (unigram, bigram, trigram, etc)

The Naive Bayes model performed terribly on in- and out-of-sample data (see Figure 3); however, Naive Bayes performs better using character bigrams on in-sample data than any other character n-gram size for small text sizes; for large text sizes (1000 words and beyond), unigrams seem to perform better.

A bizarre observation is that the Naive Bayes model has an F1 score very close to 0% on word n-grams. As each dialect is so linguistically similar, word n-grams may not be enough to determine the national dialect of a text based on probability.

Based on these results, probabilistic models like the Naive Bayes classifier do not seem to perform well on German dialect classification.

7.3 Neural Network (Ensemble Method)

The constituent parts of the Deep Learning model performed far better than previous methods on the test set, as seen in Table 2. This was promising, but did not guarantee performance for out of sample data.

The reason for this seemed to either be that the deep learning model was over-fitting on the training data or had too little data to generalise well. The solution for both of these problems would be to increase the amount of data available. We attempted to see if a larger corpus would make a difference - while not submitted as part of this paper, we make note that a 100,000 word Swiss-German subcorpus (as Swiss was the class that was getting the worse predictions) worked very well, giving out of sample accuracy of more than 90% on the test set and out of sample tests.

Model	Feature	Layers	Accuracy	Loss
1	char	2	0.943	0.111
2	char	2	0.989	0.036
3	word	2	0.978	0.093

Table 2: Performance on test-set data of the 3 constituent Deep Learning models based on hyperparameter configuration. Accuracy and Loss figures given to 3 decimal places.

The slight faults in performance of each classifier individually affected the ensemble classifier, particularly on out-of-sample data as can be seen in Table 3. However, we were impressed that our Deep Learning ensemble model performed with an F1-score of 0.694 on out-of-sample data, and an accuracy of 73.4%. This is not state-of-the-art, but using the multi-layer perceptron model has allowed us to get a 'reasonably' accurate classifier.

Accuracy	Recall	Precision	F1 Score
0.734	0.842	0.734	0.694

Table 3: Out of sample scores for the ensemble model, all given to 3 decimal places.

8 Conclusion

While traditional methods of classifying text have proven successful for many other tasks, cosine similarity and deep learning approaches seem to be the best for classifying German dialects. German-speaking countries have a lot of shared history and influence from other languages (French for example), which makes classification trickier. However, predicting class to a relatively high degree of accuracy still seems possible with a deep learning/ensemble approach. These novel methods seem very promising to handle NLP tasks with relatively small corpora sizes, and we believe that while our results do not indicate perfect classifier results, they indicate appropriate processes for data collection, data preparation and modelling.

References

1. Banko, M. and Brill, E. 2001. Scaling to very very large corpora for natural language disambiguation. In: *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics, 6-11 July 2001, Toulouse, France*. United States: Association for Computational Linguistics. pp.26–33.

2. Francis, W. N. and Kučera, H. 1967. *Computational Analysis of Present-Day American English*. Providence: Brown University Press.

3. Ammon, U. 2011. *Die deutsche Sprace in Deutschland, Österreich und der Schweiz: Das Problem der nationalen Varietäten*. Berlin, New York: De Gruyter.

4. Lingoda, T. 2020. Deutsch in Deutschland, Österreich und der Schweiz: Hauptunterschiede des Vokabulars. 6 October. *Lingoda*. [Online]. [Accessed 20th April 2022]. Available from: https://blog.lingoda.com/.

5. Alshutayri, A. and Atwell, E. 2019. *Classifying Arabic dialect text in the Social Media Arabic Dialect Corpus (SMADC)*. In: Proceedings of the 3rd Workshop on Arabic Corpus Linguistics, Cardiff, United Kingdom. Association for Computational Linguistics, pp. 51–59.

6. Alshutayri, A., Atwell, E., Alosaimy, A., Dickins, J., Ingleby, M. and Watson, J. 2016. *Arabic Language WEKA-Based Dialect Classifier for Arabic Automatic Speech Recognition Transcripts*. In: Proceedings of the Third Workshop on NLP for Similar Languages, Varieties and Dialects (VarDial3). Osaka, Japan: The COLING 2016 Organizing Committee, pp. 204–211.

7. Alshutayri, A. and Atwell, E. 2017. Exploring Twitter as a source of an Arabic Dialect Corpus. *International Journal of Computational Linguistics (IJCL)*. **8**(2), pp. 37-44.

8. Alshutayri, A. and Atwell, E. 2018. Creating an Arabic Dialect Text Corpus by Exploring Twitter, Facebook, and Online Newspapers. In: Al-Khalifa, H, Magdy, W, Darwish, K and Elsayed, T, (eds.) *OSACT 3 Proceedings, OSACT 3 The 3rd Workshop on Open-Source Arabic Corpora and Processing Tools, co-located with LREC 2018, 08 May 2018, Miyazaki, Japan.*. [no place]: LREC, pp. 54-61.

Appendix

Linguistic Differences

Austrian German	High German	English
Jänner	Januar	January
Heuer	Dieses Jahr	This year
Stiege	Treppe	Stairs
Erdapfel	Kartoffel	Potato
Faschiertes	Hackfleisch	Minced Meat

Table 4: vocabulary differences Austria Germany (Lingoda, 2020)

Swiss German	High German	English
Trottoir	Gehsteig	Pavement
Coiffeur	Frisör	Hairdresser
Fasnacht	Fasching	Carnival
Velo	Fahrrad	Bicycle
Portmonee	Geldbeutel	Purse

Table 5: vocabulary differences Switzerland Germany (Lingoda, 2020)