Investigating the Influence of Selected Linguistic Features on Authorship Attribution using German News Articles

Manuel Sage[1], **Pietro Cruciata**[2], **Raed Abdo**[3], **Jackie Chi Kit Cheung**[4] and **Yaoyao Fiona Zhao**[1]

[1]Department of Mechanical Engineering, McGill University
[2]Department of Mathematics and Industrial Engineering, Polytechnic de Montreal
[3]Department of Electrical Engineering, McGill University
[4]School of Computer Science, McGill University
{*manuel.sage@mail., raed.abdo@mail., jcheung@cs., yaoyao.zhao@*}*mcgill.ca*
{*pietro.cruciata@polymtl.ca*}

Abstract

In this work, we perform authorship attribution on a new dataset of German news articles. We seek to classify over 3,700 articles to their five corresponding authors, using four conventional machine learning approaches (naïve Bayes, logistic regression, SVM and kNN) and a convolutional neural network. We analyze the effect of character and word n-grams on the prediction accuracy, as well as the influence of stop words, punctuation, numbers, and lowercasing when preprocessing raw text. The experiments show that higher order character n-grams (n = 5,6) perform better than lower orders and word n-grams slightly outperform those with characters. Combining both in fusion models further improves results up to 92% for SVM. A multilayer convolutional structure allows the CNN to achieve 90.5% accuracy. We found stop words and punctuation to be important features for author identification; removing them leads to a measurable decrease in performance. Finally, we evaluate the topic dependency of the algorithms by gradually replacing named entities, nouns, verbs and eventually all tokens in the dataset according to their POS-tags.

1 Introduction

When the author of a text is the subject of particular interest, there exist three main approaches in the field of natural language processing (NLP): author profiling, authorship verification, and authorship attribution. They mean respectively, aiming at detecting details of the author such as age or gender, measuring the similarity between an author's work and a text in question, and trying to identify the author of a text given a group of potential authors. All approaches are based on the assumption that individuals have unique writing styles and habits (Stamatatos, 2009). In this project, we focus on authorship attribution (AA), a task popular in many areas such as literary studies, history and forensic linguistics (Evert et al., 2017). Anonymity and potential content creation under false name on the internet have recently increased the interest in AA (Aborisade and Anwar, 2018), (Rocha et al., 2017).

Working on a new dataset of 3,700 German news articles written by five authors, we carry out a multiclass classification using different machine learning (ML) models and linguistic features. As ML models, we test multinomial naïve Bayes (NB), logistic regression (LR), support vector machine (SVM) and k-nearest-neighbors (kNN) using scikit-learn implementations, and a convolutional neural network (CNN) for text classification using PyTorch. We experiment with word n-grams, character n-grams and fusions of both, as well as punctuation, numbers, stop words, and lowercasing. We further seek to evaluate the topic-dependency of the algorithms by gradually replacing named entities, nouns, verbs and all tokens in the dataset with the help of their part-of-speech (POS) tags. This study aims to quantify the effect of linguistic features on AA using a new source of German news articles.

2 Related Work

Previous approaches to AA greatly vary regarding applied linguistic features, implemented ML models and investigated languages. The combination of character n-grams and traditional ML models such as naïve Bayes has been deployed for many publications on AA, such as in Amasyalı and Diri (2006) on Turkish news articles, Markov et al. (2017) on Portuguese news articles, or Oppliger (2016) on instant messages in Swiss German. Punctuation n-grams have been used as stylometric features for English, French, Italian and Spanish (Martín-Del-Campo-Rodríguez et al., 2019). In addition, Khan (2018) and Schwartz (2016) demonstrated the importance of stop words for AA with English language data.

Originally developed for computer vision tasks, CNNs have been successfully applied in various text classification tasks lately, including AA. Ruder et al. (2016) presented state-of-the-art results in large-scale AA on various social media datasets. The authors of the paper implemented different multi-channel word and character CNNs and create hybrid models of both. On average, the char-CNN outperformed not only previous models, but also word CNNs and hybrids. In the study of Shrestha et al. (2017), a CNN with three convolutional layers was trained on character n-grams and outperformed traditional models as well as recurrent neural networks (RNN) on recognizing the authors of tweets.

3 Dataset

We collect a new dataset of newspaper articles published by Main-Post, a German newspaper. In cooperation with the newspaper, we decide to choose the articles by five journalists from the regional department in Schweinfurt. In their daily work, the journalists cover similar topics, mostly local news in and around the city. With this choice, we hope to alleviate the likelihood of classifying authors by topic specific vocabulary. The newspaper reaches a circulation of around 40,000 readings per day in Schweinfurt. All articles (mostly behind a paywall) can be accessed via the company's online presence.[1] In a first step, we collect the weblinks to all articles for each author. Then, a second script opens each link and extracts the corresponding text into a csv-file. We clean the collected articles by removing:

- Author names in the text where indicative of the writer (e.g. comments);
- Articles written by multiple authors;
- Articles listed more than once (e.g. regional and trans-regional versions);
- Non-text elements such as image or video boxes that were downloaded due to variations in the webpages' html-structures.

The final dataset consists of 3,717 articles by five different journalists, written between May 2013 and October 2019. The number of articles per author is imbalanced and varies from 331 to 972. The average length of an article is 455 words and it comprises 24 sentences and 7.5 paragraphs. The shortest article measures 26 words, the longest 2299. The overall corpus size is 1.6 Million words. Compared to other author attribution datasets in literature, our dataset contains fewer authors, but larger available data per author. This facilitates the prediction task but allows to draw more meaningful conclusions about the effect of analyzed linguistic features. The dataset is not publicly available but can be obtained from the authors of this paper on reasonable request.

4 Methodology

4.1 Preprocessing & Model Implementations

As a first step, we split off a stratified test set containing 20% of each author's available articles. All final models are evaluated by their prediction accuracy on this test set. Since this is the first work on a new dataset, we start by establishing two baseline models. Due to their decent performance in many applications, we chose naïve Bayes and logistic regression, a generative and a discriminative approach, respectively. We process the raw text using word unigrams obtained through splitting by whitespaces and test for the following linguistic features:

- Punctuation (keep/remove);
- Numbers (keep/remove);
- Stop words (keep/remove), using NLTK's list of 232 German stop words;
- Lemmatization, using Spacy's implementation for German;
- Stemming, using NLTK's German Snowball-Stemmer;
- Lowercasing.

[1] https://www.mainpost.de/regional/schweinfurt/

We further define a minimum document frequency of 5 and tune the model-specific hyperparameters via grid search. For NB, LR, SVM and kNN, we vectorize counting raw token frequencies using scikit-learn's CountVectorizer. As a result of the baseline experiments, the best parameter combinations achieved 81.85% on the test set for naïve Bayes and 90.59% for logistic regression. During our work on the baselines, we observed an average performance decrease of 2.8 percentage points for lemmatization and 3.9 percentage points for stemming, along with a drastically increasing runtime. Thus, both techniques were excluded from the following experiments. Instead, we focus on the effect of punctuation, numbers, stop words, and lowercasing on the prediction accuracy.

We expand our work with SVM and kNN models, and add bigrams and trigrams, as well as character n-grams of different lengths (n = 3 – 6). Finally, for each model, we combine the best word n-gram vectorizer with the best character n-gram vectorizer in a fusion model. For every combination of model and word/character n-gram, a random search with 100 iterations tests the beforementioned linguistic features and the model specific hyperparameters by averaging the results of a 5-fold cross-validation. Then, each model's best configuration is trained on the whole training set and its performance on the test set is reported.

The implementation of CNN is based on the work of (Trevett, 2019) on CNNs for multi-class sentiment analysis, that we adjust for our experiments. In all set-ups, the network consists of an embedding layer, at least one convolutional layer, and a fully connected output layer. The model is fed with a pretrained German word embedding, trained on two million Wikipedia articles, with a disk size of 6.4 GB (Cieliebak et al., 2017).[2] To obtain results comparable to the other ML models trained on word/character n-grams, we use one-layer convolutional filters of size n after tokenizing the text on word and character level (e.g. filter of size 2 for bigrams). Instead of fusions/hybrids between word and characters, we optimize the CNN by adding multiple convolutional layers to achieve higher accuracies. In addition to testing on the linguistic features described above, we experiment with different values for filter sizes, number of filters and layers, dropout regularization, max-pooling, and the size of vocabulary. We select Adam as optimizer and cross-entropy as loss-function. After splitting off the test set, 20% of the remaining data is used for validation. During training, the performance is validated after every epoch and the overall best model parameters are saved. For testing, we load the best parameters and run the algorithm on the test set.

4.2 Experiments with reduced topic-dependency

The five authors in the dataset work in the same regional department. Nevertheless, each author has special topics, for example certain cultural events or news from particular villages. With this information in the training data, the models might learn to predict authors based on specific words appearing in an article and not through each author's writing style. We seek to quantify this assumption and evaluate how much the performance of the established models depends on the vocabulary used. Therefore, we replace all tokens from different part-of-speech categories with abbreviations and create the following four variations of the dataset (including test set):

- Replace all named entities by 'NE';
- Additionally, replace all nouns by 'NN';
- Additionally, replace all verbs (including auxiliary verbs) by 'VB';
- Finally, replace all tokens by their corresponding Treebank POS-tag.

Then, we run the best performing version for each machine learning model on these variations of the dataset and report the accuracy on the hold out test set. We utilize Spacy's pretrained German tagger with a reported accuracy of 96.3% for POS-tagging.

5 Results

5.1 Preprocessing & Model Implementations

The left side of Table 1 shows the best results obtained for the five implemented ML models. Overall, we note high performances above 80% for all models except the kNN approach. Logistic regression and SVM performed the best and achieved almost same results in all tested variations. Except for kNN, word n-grams slightly outperformed character n-grams. On word level, adding bigrams and trigrams outperformed unigrams, with only marginal differences between the models. On

[2] https://www.spinningbytes.com/resources/

Model	Word n-grams			Character n-grams				Fusion	Reduced topic-dependency, replace			
	n=1	n=2	n=3	n=3	n=4	n=5	n=6		NE	+ Nouns	+ Verbs	All
NB	81.9*	86.0	**86.2**	77.2	80.2	82.3	82.1	83.0	87.0	86.4	86.6	77.6
LR	90.6*	91.4	91.4	88.6	89.9	90.2	91.3	**91.7**	91.53	89.8	89.5	76.3
SVM	90.5	91.5	91.1	87.6	89.7	90.5	91.1	**91.9**	92.2	89.8	88.2	77.4
kNN	65.7	62.4	60.0	69.0	**70.3**	65.9	64.1	70.2	72.2	67.7	65.7	64.3
CNN	88.7	89.7	88.7	82.5	85.7	88.8	88.7	**90.5**✣	88.6	87.3	86.3	78.8

Table 1: Test set accuracies in %. Note that higher order n-grams always include all lower order combinations. For the CNN, n refers to the filter size using a single convolutional layer after word/character tokenization.
*Baseline models ✣Best result for multi-layer CNN, obtained with word tokenization

character level, higher order n-grams improved the results. Combining the best character and word n-grams in fusion models led to higher accuracies for LR and SVM. Here, SVM delivered the best predictions with 91.9% (macro-averaged F1-score = 0.898). In all variations, SVMs with linear kernels were more accurate than those with polynomial kernels. On average, the performance of the CNN was two percentage points below the SVM. Both, character and word n-gram features achieved accuracies in the high 80s. The best performance of the CNN was 90.5% (macro-averaged F1-score = 0.855), applying two convolutional layers of filter sizes 2 and 3, with 500 filters each, after word level tokenization. Using smaller vocabulary sizes of 5k and 10k improved results, as well as high dropout values of 0.5.

Besides word and character n-grams, the conducted experiments allow us to quantify the effect of stop-words, punctuation, numbers and lowercasing on the performance for this dataset. The corresponding values are displayed in Table 2 and obtained by taking each model's best implementation, retraining it either with or without the feature in question, and finally averaging the performance difference over all models for each feature. For all 28 random searches and 8 CNN-configurations, removing stop words or punctuation decreased the prediction accuracies. For the most accurate implementations, this resulted in an average decrease of 1.23 and 1.06 percentage points, respectively. Lowercasing and removing numbers on the other hand barely influenced the results.

Feature	Average effect on performance
Removing stop words	− 1.23
Removing punctuation	− 1.06
Removing numbers	− 0.09
Lowercasing	+ 0.09

Table 2: Average effect of tested features on prediction accuracy in percentage points.

5.2 Experiments with reduced topic-dependency

The results of the experiments with reduced topic-dependency are presented in Table 1. Replacing named entities by 'NE' did not affect the performance negatively as expected. Instead, kNN, SVM and naïve Bayes slightly improved their performances and the SVM reached 92.2% (macro-averaged F1-score = 0.901), the highest accuracy in the project. Replacing nouns and then verbs decreases the performance for all models yet still allows accuracies in high 80s (except kNN). Finally, replacing the whole text with corresponding Treebank POS-tags led to poorer yet reasonable results above 76%, again excluding kNN.

6 Discussion & Conclusion

With logistic regression, SVM and CNN, three models reached prediction accuracies over 90%, while kNN was found less applicable on this task. The CNN did not outperform traditional approaches in this work. However, the innumerable network structures and tunable (hyper)parameters of this model leave room for improvement. We are in line with Sanchez-Perez et al. (2017) showing that higher orders of character n-grams outperform lower orders. Combining word and character n-grams also improved results. Therefore, exploring longer character n-grams (n = 7, 8, . . .) could extend this study.

Regarding text preprocessing, we conclude that most changes in the raw text, despite being useful in other NLP domains, decrease the performance on this task. Removing stop words reduces the accuracy measurably, this confirms the consent in literature. In the study of Arun et al. (2009), stop words play an essential role in authorship attribution on English text documents. We detected a similar importance for punctuation whereas numbers and lowercasing barely affected performance. Due to a lower accuracy of NB and LR after lemmatization and stemming, we assume that both techniques disguise characteristics in an author's writing style. However, this assumption requires further evaluation. In the experiments with reduced topic-dependency, the models did not depend on certain keywords, but more on the overall structure and writing style of an author's work. Replacing named entities improved predictions. This contradicts the initial hypothesis and other researcher‘s work, such as Sanchez-Perez et al. (2017), where accuracies dropped by approximately 2-3 percentage points. We assume that the intersection between the authors' work is too large to allow models to classify based on named entities. Instead, removing them could reduce the variance of the vectorizer and help to focus on more meaningful writing patterns. More experiments, e.g. with POS n-grams, could further improve results.

Overall, this work demonstrated the importance of stop words, punctuation, and fusions of word and character n-grams for AA on German news articles. It further revealed the potential of POS-tags as meaningful features for this task.

Acknowledgments

We would like to thank Main-Post GmbH for their cooperation in finding adequate authors and free access to all articles.

This research work is supported by National Sciences and Engineering Research Council of Canada Discovery Accelerator Supplements RGPAS-2018-522708.

The fourth author is supported in part by the Canada CIFAR AI Chair program.

References

O. Aborisade and M. Anwar. 2018. Classification for Authorship of Tweets by Comparing Logistic Regression and Naive Bayes Classifiers. In *2018 IEEE International Conference on Information Reuse and Integration (IRI)*, pages 269–276.

Fatih Amasyalı and Banu Diri. 2006. Automatic Turkish text categorization in terms of author, genre and gender. In *International Conference on Application of Natural Language to Information Systems*, pages 221–226. Springer.

R. Arun, V. Suresh, and C. E. V. Madhavan. 2009. Stopword Graphs and Authorship Attribution in Text Corpora. In *2009 IEEE International Conference on Semantic Computing*, pages 192–196.

Mark Cieliebak, Jan Milan Deriu, Dominic Egger, and Fatih Uzdilli. 2017. A twitter corpus and benchmark resources for German sentiment analysis. In *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, pages 45–51.

Stefan Evert, Thomas Proisl, Fotis Jannidis, Isabella Reger, Steffen Pielström, Christof Schöch, and Thorsten Vitt. 2017. Understanding and explaining Delta measures for authorship attribution. *Digital Scholarship in the Humanities*, 32(suppl_2):ii4–ii16.

Jamal Ahmad Khan. 2018. A Model for Style Breach Detection at a Glance: Notebook for PAN at CLEF 2018. In *CLEF (Working Notes)*.

Ilia Markov, Jorge Baptista, and Obdulia Pichardo-Lagunas. 2017. Authorship attribution in Portuguese using character n-grams. *Acta Polytechnica Hungarica*, 14(3):59–78.

Carolina Martín-Del-Campo-Rodríguez, Daniel Alejandro Pérez Alvarez, Christian Efraín Maldonado Sifuentes, Grigori Sidorov, Ildar Batyrshin, and Alexander Gelbukh. 2019. Authorship attribution through punctuation n-grams and averaged combination of SVM notebook for PAN at CLEF 2019. In *CEUR Workshop Proceedings*, volume 2380.

Rahel Oppliger. 2016. Automatic authorship attribution based on character n-grams in Swiss German. *Bochumer Linguistische Arbeitsberichte*, (16):177–185.

A. Rocha, W. J. Scheirer, C. W. Forstall, T. Cavalcante, A. Theophilo, B. Shen, A. R. B. Carvalho, and E. Stamatatos. 2017. Authorship Attribution for Social Media Forensics. *IEEE Transactions on Information Forensics and Security*, 12(1):5–33.

Sebastian Ruder, Parsa Ghaffari, and John G Breslin. 2016. Character-level and multi-channel convolutional neural networks for large-scale authorship attribution. *arXiv preprint arXiv:1609.06686*.

Miguel A Sanchez-Perez, Ilia Markov, Helena Gómez-Adorno, and Grigori Sidorov. 2017. Comparison of character n-grams and lexical features on author, gender, and language variety identification on the same Spanish news corpus. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 145–151. Springer.

Maxwell B Schwartz. 2016. An examination of cross-domain authorship attribution techniques. *CUNY Academic Works*. https://academicworks.cuny.edu/gc_etds/1573.

Prasha Shrestha, Sebastian Sierra, Fabio González, Manuel Montes, Paolo Rosso, and Thamar Solorio. 2017. Convolutional Neural Networks for Authorship Attribution of Short Texts. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 669–674, Valencia, Spain. Association for Computational Linguistics.

Efstathios Stamatatos. 2009. A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, 60(3):538–556.

Ben Trevett. 2019. Pytorch sentiment analysis. `https://github.com/bentrevett/pytorch-sentiment-analysis`. Accessed: 2019-12-04.