Improving Cross-domain Authorship Attribution by Combining Lexical and Syntactic Features

Notebook for PAN at CLEF 2019

Martijn Bartelds* and Wietse de Vries*

* the authors contributed equally to this work

University of Groningen, The Netherlands
{m.bartelds.2, w.de.vries.21}@student.rug.nl

Abstract Authorship attribution is a problem in information retrieval and computational linguistics that involves attributing authorship of an unknown document to an author within a set of candidate authors. Because of this, PAN-CLEF 2019 organized a shared task that involves creating a computational model that can determine the author of a fanfiction story. The task is cross-domain because of the open set of fandoms to which the documents belong. Additionally, the set of candidate authors is also open since the actual author of a document may not be among the candidate authors. We extracted character-level, word-level and syntactic information from the documents in order to train a support vector machine. Our approach yields an overall macro-averaged F1 score of 0.687 on the development data of the shared task. This is an improvement of 18.7% over the character-level lexical baseline. On the test data, our model achieves an overall macro F1 score of 0.644. We compare different feature types and find that character n-grams are the most informative feature type though all tested feature types contribute to the performance of the model.

1 Introduction

Authorship attribution is an established research area in computational linguistics that aims to determine the author of a document by taking the writing style of the author into account. Typically, a system assigns a candidate author to an anonymous text by comparing the anonymous text to a set of possible author writing samples. Currently, the field of authorship attribution can be considered as a topic of pivotal interest as the authenticity of information presented in the media is often questioned. Following this, any successfully attempt in revealing the authors behind a text will result in improved transparency and ideally removes any uncertainty with respect to the validity of information presented. As a consequence, authorship attribution can be determined as being closely related to research tailored to the privacy domain, law, cyber-security, and social media analysis.

In the PAN-CLEF 2019 shared task on authorship attribution, a cross-domain authorship attribution task was proposed based on fanfiction texts. More specifically, fanfiction texts are written by admirers of a certain author and these fanfiction texts are known to substantially borrow characteristics from the original work. This task can be considered cross-domain, since the documents of known authorship are not necessarily collected within the same thematic domain or genre. Moreover, this task is extended beyond closed-set attribution conditions, as the true author of a given text in the target domain is not by definition included in the set of candidate authors.

In this work, we present the methodology and results of our submission to the PAN-CLEF 2019 cross-domain authorship attribution task. We developed our approach with respect to the documents of all four languages provided in the PAN-CLEF 2019 data set. These languages include: English, French, Italian, and Spanish. Previous research showed the effectiveness of textual features such as character-level n-grams to authorship attribution problems, since these are capable of representing more abstract level writing style characteristics rather than generating a representation that is purely related to the content of the document [18, 19, 21]. However, limitations of these features arise together with the observation that they often result in sparse representations for documents of insufficient size. Therefore, we aimed to create an approach in which we create textual representations across different levels inside the training documents. As such, we intended to prevent overfitting by developing a model that yields robust performance across the different genres.

2 Related Work

In this section, we will describe some of the most successful modern authorship attribution methods. Furthermore, a brief overview of the subfield of cross-domain authorship attribution will be provided.

Generally, there is a distinction between the use of profile-based and instance-based approaches in solving authorship identification problems [20]. In profile-based approaches, all available training texts per author are concatenated to create a cumulative representation of the author's writing style. In contrast, instance-based approaches treat each training text as an individual representation of the author's writing style. When these approaches are compared, it is shown that profile-based approach may be advantageous when only training documents of limited size are available [20]. As a result, the concatenation of the training documents may lead to a more reliable representation of the author's writing style. In contrast, the implementation of an instance-based approach ensures that interactions between several stylometric features can be captured, even when the distributions of these features differ between documents that are written by the same author. As an extension to both the profile-based and instance-based approaches, hybrid approaches are proposed that integrates aspects of both the profile-based and instance-based approaches [4]. Then, a single vector per author will be produced by averaging the sum of the individually represented training texts. We argue that these hybrid approaches might be superior to the profile-based and instance-based approaches, since they could capture more reliable characteristics of writing style across multiple documents of the same author.

As can be observed from the results of the PAN-CLEF 2018 shared task on cross-domain authorship attribution, we examine that the best performance was obtained by the implementation of character-level and word-level n-grams as textual features [7]. These were normalized by a tf-idf weighting scheme and used in combination with support vector machines. Previous research on cross-domain authorship attribution endorses the effectiveness of a support vector machines applied on character-level n-grams [18, 19, 21]. This suggests that the use of these methodologies can still be determined as a valuable strategy in solving cross-domain authorship attribution problems.

In previous research, thirty-nine different types of textual features that are often used in modern authorship attribution studies were compared [3]. In this study, a token normalized version of the punctuation frequency was the most successful feature used to discriminate between the different authors. Moreover, character-level bi-grams and tri-grams were also among the most promising textual features presented, and this result is substantiated by the results of numerous other research findings [2, 6, 7, 13, 22]. Following this, we decided to include these features into our own approach by creating such a feature representation that is tailored to the data set we had available.

Furthermore, research on cross-domain authorship attribution showed that topic-dependent information can be discarded from documents by carrying out several pre-processing steps [9]. It was suggested to replace all digits by zero, separate punctuation marks from their adjacent words, and replace named entities with a dummy symbol. The latter pre-processing step is effective, since named entities are often strongly related to the topic of a document. After these steps character-level n-grams were extracted from the documents, and an increase in performance was reported when these pre-processed character-level n-grams were used. Moreover, attribution performance can be improved by applying a frequency threshold to the extracted character-level n-gram representations [9]. As such, the least frequent occurring n-grams associated with topic-specific information should be removed from the model. In our work, we decided to extend on their work by implementing an adaptation of their suggested frequency threshold. Moreover, we attempt to improve our model performance by applying these pre-processing steps not solely on character-level n-grams, but on multiple textual feature levels inside their training documents.

3 Data and Resources

A development data set is provided by the organizers of the authorship attribution task. The goal of the shared task is not to train a model on known training data and to test it on unknown test data, but rather to design a model that can be trained on unknown training data and then be tested on unknown test data. The development data set is not called a training data set, because there is no overlap in candidate authors in the development data and the undisclosed data that the models are trained and evaluated on for the shared task. Instead, the development data contains twenty separate problem sets with each a training part and a test part. The final evaluation will be performed on a similar set of problem sets.

The development data set contains twenty problem sets in four different languages, resulting in five problem sets per language. The set of languages consists of English, French, Italian and Spanish. Each problem set contains nine candidate authors with seven known documents each. The task is to assign each document in a set of unknown documents to a candidate author within the problem set, if the author unknown document is actually in the candidate set. There is also the possibility that the actual author is unknown and in that case the unknown document should be given an *unknown* label. For evaluation a separate corpus is held back with similar characteristics as the development corpus. The evaluation data contains problem sets in the same languages as the development data, but there is no overlap in authors in the development and test sets. Therefore, no features can be learned for specific authors before testing.

The documents that are to be classified consist of a set of fanfiction stories with a length of 500 to 1000 tokens each. The stories were scraped from an online fanfiction website. Candidate authors write stories in different fandoms, so it is important that the model will learn author-specific textual features and not features that are inherent to the fandom or the universe that the story takes place in. Because of the large content differences between fandoms, the fandoms are considered to be different domains.

The amount of unknown documents per problem set is highly variable, since it ranges between 46 and 561 documents per problem set. Moreover, occurrences of candidate authors in the unknown documents are not uniformly distributed either. This is not inherently important for development and methodology design choices. However, very rare occurrences of certain candidate authors may have large effects on evaluation scores during development, since scores are macro-averaged per candidate within the problem sets. The fraction of unknown documents that are not written by any known author is also variable. Although, within the development set overall a third of all documents are written by an unknown author.

The imbalance of the testing parts of the problem sets in the development data does not influence model training, since the distribution is unknown at training time. During the development of our methodology, the average result may however be strongly influenced by unbalanced problem sets. The goal is to let our model be able to work with problem sets with unknown distributions, so the development data should be balanced when we want to trust the results. Our algorithmic and hyper-parameter choices are therefore not based on the given configuration of the development data set. Instead, we merged the problem sets per language and evaluated on random permutations of candidate authors. Our new shuffled problem sets contained nine random candidate authors in the same language with seven random known documents for each author. It can be possible that these documents originate from documents that were originally meant for model testing. The test data for our new shuffled problem sets contained up to 100 documents per candidate author that were not used for training. The set of test documents is extended with documents that are not written by the candidate authors. These documents served as the documents with an unknown author, and a third of each generated problem set consists of these documents with an unknown author. This method enabled us to generate a large amount of unique permutations of problem sets. For the development of our methodology, we evaluated our choices by using a fixed set of twenty shuffled problem sets per language, as opposed to the original five problem sets per language. The original development problem sets are used for validation and the results in this paper are evaluated on the original problem sets.

4 Methods

The code that is used for our authorship attribution approach is fully open-source and available at https://github.com/wietsedv/pan19-cross-domain-authorship-attribution.

4.1 Classification approach

Because of their success in previous authorship attribution approaches, we choose to use a support vector machine with document level features. We extract different types of textual features from the known documents which are used to train a support vector machine (SVM) model for each problem set. We used the SVM implementation that is available in the Scikit-learn Python package [12]. Hyper-parameters are tuned globally using a grid search method with our randomly permuted problem sets. Therefore, hyper-parameters are equal for all languages. The specific values of the hyper-parameters will be discussed in Section 4.3. Hyper-parameters that are specific for feature types are tuned separately from each other using separate grid searches within intuitively plausible parameter ranges. This constrained grid search is chosen because of computational limitations, but also to prevent overfitting on the hyper-parameters.

The classifier that is used is a support vector machine classifier with a linear kernel. Multiple classes are handled using the one-vs-rest scheme. We also tried using other SVM kernels as well as the using a random forest classifier, but preliminary results indicated that these options are unlikely to lead to better classification accuracy results in this task.

The support vector machine classifier has to be reasonably certain in its candidate author choice, since there are also test documents that have an unknown author. This is achieved by setting a probability threshold for the support vector machine classifications. Probabilities are calculated in the SVM model by scaling the distance of the sample to each hyper-plane between zero and one. SVM predictions and probability values can be heavily influenced by single training documents because of the small amount of documents for each author and the high risk of learning fandom specific features. Additionally, SVM outputs may not be reliable estimators for probability. Therefore, a good solution in finding more reliable probabilities would be to use probability calibration [10]. In this process, five-fold cross validation is applied on the training data to train five separate classifiers. In each of these classifiers, probability estimations are calibrated using Platt scaling [16]. Probability estimations of the five classifiers are averaged to get the final probabilities.

Each test document is attributed to the most probable author if and only if the difference between the maximum probability and the second highest probability score is at least 0.1. As opposed to an absolute minimum probability, this minimum difference threshold is less sensitive to different probability distributions. Contrasting distributions in different languages or problem sets may result in very differing maximum probabilities. However, we are only interested in cases where the most probable choice is more likely distinguishable than the second most probable choice. The choice of a minimum probability difference of 0.1 is arrived at by using a grid search with values between 0.01 and 0.3 with intervals of 0.05.

The features that are used with the SVM consist of an union of six different feature types that will be described in Section 4.3. The different types of features rely on different representations of the documents for which pre-processing is required. In the next section (Section 4.2), the pre-processing steps are described that are needed to extract these linguistic features.

4.2 Pre-processing

The first preprocessing step is to tokenize the documents. Tokenization is done using the UDPipe [23] tokenizer to get tokens in the format that can be used by both the part-of-speech tagger and the dependency parser. For part-of-speech tagging, Structbilty [15], a Bi-LSTM sequence tagger, is trained on Universal Dependencies data sets [11]. Validation set accuracy scores after training are all between between 0.95 and 0.98 for the four different languages. The part-of-speech tagger is trained on Universal Dependencies data sets. More precisely, we used `UD_English-EWT`, `UD_French-GSD`, `UD_Italian-ISDT` and `UD_Spanish-GSD`.

Dependency parses are provided by the UUParser [8]. UUParser was trained using the same data sets as were used by the part-of-speech tagger. When using the document tokens as input, the parser achieves validation LAS scores between 0.83 and 0.89 on the Universal Dependencies development treebanks. To improve performance, we also trained the parser by using ELMo embeddings [14] of the documents instead of the tokens. The ELMo embeddings were extracted using pre-trained ELMo representations [1]. As a result, the validation scores after training the parser ranged between 0.86 and 0.90, which is a considerable improvement over the original model. Calculating ELMo representations is however an expensive process and because of hardware limitations in the shared task setup, we decided to use the token based dependency parses instead of the ELMo based parses. This compromise does not not have large negative effects on the classifier performance as will be discussed in section 5.

4.3 Features

Different textual feature types are extracted from the documents independently from each other. Each feature type yields numeric features that are linearly scaled between zero and one. Subsequently, the dimensionality of each feature type is reduced to 150 by applying truncated singular value decomposition. After this dimensionality reduction for each feature type, all features are combined into a single feature set. The dimensionality is reduced before combining the features to make sure that all feature types are fairly represented in the feature set. As discussed before, the hyper-parameters for feature extraction are tuned per feature using a grid search approach. Fine tuned hyper-parameters for features include the n-gram range, the use of tf-idf, maximum document frequencies and minimum group document frequencies. The minimum group document frequency is a threshold that we have created that ensures that any feature must at least be present in n documents with the same target label during training. This created hyper-parameter eliminates features that are only present in few documents written by an author, which suggests that the feature is domain related instead of author related.

For each of the feature types, we explored n-gram ranges between one and five. Subsequently, the following features were extracted from the documents:

Character n-grams The first feature type that is included in our model is based on the tf-idf scores of character n-grams in the raw document text. The value of n after tuning ranges between two and four.

Punctuation n-grams This feature type consists of n-grams of consecutive punctuation tokens where non-punctuation tokens are skipped. For example: this feature type contains the bi-gram ",." if a sentence contains a comma and ends with a dot. Occurrences of uni- and bi-grams are counted and used as features.

Token n-grams This feature type consists of the counts of token n-grams in the tokenized text. Only bi-grams are counted for this feature type, and only bi-grams that occur in at least five documents are included.

Part-of-speech n-grams This feature type consists of the counts of n-grams of the part-of-speech tags corresponding to each of the tokens in the document text. The value of n after tuning ranges between one and four.

Dependency relations syntactic n-grams This feature type consists of sequences of dependency relations. These sequences are created by chaining the syntactic relations between words. For instance, a bi-gram consists of the relation between a word and its head, and the relation between the head and its head. Note that this chaining procedure is different from the positional ordering of the words. For dependency relation syntactic n-grams, only uni-grams and bi-grams are included that occur at least thrice in the training document of a candidate author. The dependency relation syntactic bi-grams for instance include $nsubj - ROOT$, if a sentence contains a nominal subject that is connected to the root of the sentence.

Token syntactic n-grams This feature type consists of actual words in syntactic n-gram relations. The same chaining procedure is used but actual tokens are chained instead of relation labels. Token syntactic n-grams also have a minimum group document frequency of three. The n-gram range for this feature type is two to three.

5 Results

Following the description of the PAN-CLEF 2019 shared task, we aimed to determine the author of a fanfiction text among a list of candidate authors. Following this, we have created a support vector machine approach with multiple features derived from different textual levels inside the documents.

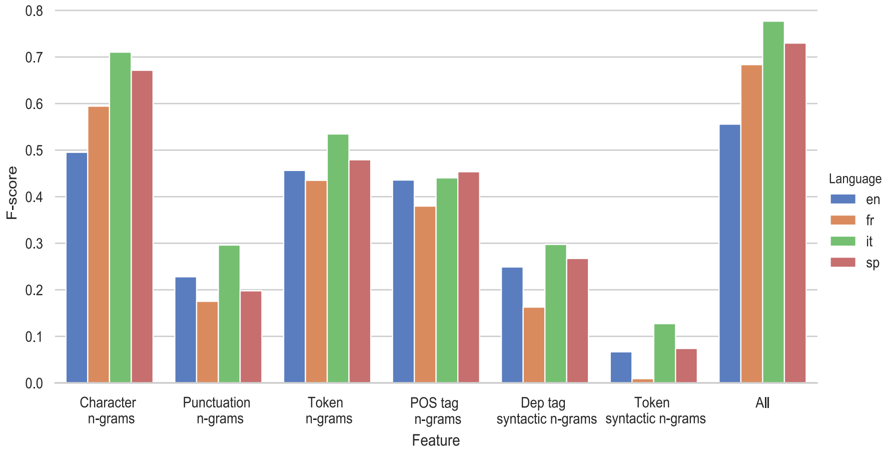

Figure 1. F-scores per feature type per language

5.1 Individual feature types

In Figure 1, both the performance of the individual feature types and the overall performance of our approach are visualised. Just like the evaluation of the shared task, the performance was evaluated by calculating macro-averaged F1 scores that were calculated when the *unknown* target labels were excluded [5]. From these results, we observe that the character-level n-grams yielded the best performance among the tested feature types. In contrast, the lowest performance for all languages was obtained with the textual representation that counted occurrences of syntactic token n-grams. These observations apply to all the languages that were included in our data set. Examining the individual languages in particular, we note that our model performed well for the Italian language. This observation applied to both the performance of the individual feature types as well as the performance when the all feature types are combined. Furthermore, the performance per feature type is the lowest when we look at the the English and French languages, suggesting that the variation of features might have less predictive power for these languages as compared to the other languages present. However, the difference may also be an artifact of the data set.

When we compare the performance of our character-level n-gram feature representation to the performance of the baseline approach that was provided by PAN-CLEF 2019, we observe that we outperformed the baseline approach by 6.7%. More specifically, the PAN-CLEF 2019 baseline approach obtained a macro averaged F1 score of 0.579 across all languages. This approach consisted of a character-level tri-gram representation in combination with a linear support vector machine, and a simple probability threshold rejection option was included to assign an unknown document to the *unknown* class. Our performance gain was calculated across all four languages that were included in the data set, and an even larger performance gain with respect to the baseline is re-

Table 1. F-scores per language from the ablation study

Feature	**English**	**French**	**Italian**	**Spanish**	**Average**
Not character n-grams	0.547	0.538	0.651	0.637	0.593
Not punctuation n-grams	0.550	0.678	0.764	0.729	0.680
Not token n-grams	0.545	0.676	0.756	0.729	0.676
Not POS tag n-grams	0.532	0.626	0.744	0.700	0.651
Not dependency tag syntactic n-grams	0.547	0.684	0.766	0.729	0.682
Not token syntactic n-grams	0.538	0.665	0.774	**0.736**	0.678
All	**0.556**	**0.684**	**0.777**	0.730	**0.687**

ported when we compare the performance of our system with all features included. Then, we outperformed the baseline approach by 18.7%.

In order to clarify the contributions of the individual features to the overall performance of our approach, we performed an ablation study. Initially, we started with the complete feature set, after which we eliminated the individual features in the feature set, respectively. As shown in Table 1, we examine that the largest decrease in performance (13.7%) was obtained when the character-level n-gram feature was omitted from the feature set. These findings correspond well to the observed effect that was previously described and visualised in Figure 1. Also, the same method of reasoning can be applied when we compare the remaining results of the ablation study with their corresponding counterparts that can be found in Figure 1. We observed higher scores for the individual performance of the token-level n-grams as compared to the individual performance of the part-of-speech-level n-grams. When we compare this observation to the outcomes of the ablation study in Table 1, we observe the opposite effect. This suggests that solely using token-level n-grams achieves better performance than solely using part-of-speech based n-grams. However, the information that is captured by token-level n-grams seems also to be captured by other feature types whereas part-of-speech based n-grams provide additional information. This observation confirms the power of combining different feature types that may not be good predictors individually.

5.2 ELMo embeddings

As illustrated in Table 2, we compared the performance of the dependency relation syntactic n-grams and contrasted these results with the performance of the complete feature set. With this comparison, we wanted to examine whether the use of ELMo embeddings improved the general performance of our approach, and we wanted to observe the effect of ELMo embeddings on the results produced by the dependency parser. As illustrated in Table 2, the performance can be observed per language, and we distinguished between F1 scores that were obtained when we trained the parser by using ELMo embeddings, and F1 scores that were obtained when we trained the dependency parser using regular tokens. Following this, we observe that the inclusion of ELMo embeddings had the largest advantageous effect on the textual problems related to the English language for both the dependency tag syntactic n-gram feature type and the complete feature set. An additional increase in performance was observed for the dependency tag syntactic n-gram feature type when looking at the French language. In all

Table 2. F-scores per language with optional ELMo embeddings

Feature	ELMo	English	French	Italian	Spanish	Average
Dependency tag syntactic n-grams	Yes	0.249	0.163	0.298	0.267	**0.244**
Dependency tag syntactic n-grams	No	0.199	0.121	0.298	0.267	0.221
All	Yes	0.556	0.684	0.777	0.730	**0.687**
All	No	0.548	0.685	0.777	0.730	0.685

other cases the use of ELMo embeddings did not have any effect or even resulted in a decrease in performance. Given the fact that the calculation of ELMo embeddings was an expensive process, we argue that including ELMo embeddings for the derivation of syntactic information is not beneficial for this task.

5.3 Development results

In conclusion, with the best feature settings included in our approach we obtained an average macro F1 score of 0.687 based on the development data. More specifically, this score was obtained by calculating the average of the F1 scores of the four individual languages. In more detail, when we observe the F1 scores per language, we are able to conclude that our model performed best for the Italian problem sets (0.777), and this score is followed by the Spanish (0.730), French (0.684) and English (0.556) problem sets, respectively.

5.4 Test results

Based on the previously described results, we submitted the full model to the TIRA submission platform with all feature types for the shared task testing phase [17]. The overall macro F1 score of our model is 0.644, which is slightly lower than our macro F1 score on the development data. The English and French testing scores are 0.558 and 0.687, respectively. These two scores are marginally higher than the scores on the development data, which indicates that our methodology appears to be robust for these languages. The Italian and Spanish scores were highest during development, but these scores have dropped to 0.700 and 0.629, respectively. The Italian and Spanish testing scores are more similar to the English and French results, which indicates that our model may perform more consistently across languages than what seemed during development. Therefore, the decrease in performance for Italian and Spanish may be a positive result.

6 Conclusion

In this paper, we presented a support vector machine approach with multiple features derived from different textual levels inside the documents. The implemented support vector machine made use of a linear kernel function, and the multiple classes that were presented to the classifier were handled using the one-vs-rest scheme. In order to be able to deal with the open-set attribution conditions, we implemented a probability threshold that was taken into account when computing the support vector machine classifications. The textual features that were used in this task consisted of a union of six different feature types that each correspond to a unique representational textual level. As such, we included character-level n-grams, punctuation-level n-grams, tokens-level n-grams, part-of-speech n-grams, dependency relations syntactic n-grams, and token syntactic n-grams. After the hyper-parameter tuning for these features, we obtained an average macro F1 score of 0.687 on the development data, and an average macro F1 score of 0.644 on the test data.

Even though we outperformed the baseline by 18.7%, we still note that cross-domain authorship attribution studies are challenging. We have demonstrated that more sophisticated features, like the inclusion of dependency tag syntactic n-grams, are capable of capturing those stylometric elements represented in texts, but without help of other feature types their predictive power is not great. However, in combination with other more simple and straightforward lexical n-grams, they do improve model performance. Further research should aim to determine whether these more elaborate textual features are able to provide an accurate and reliable basis that can be used to capture valuable elements of an author's writing style.

References

1. Che, W., Liu, Y., Wang, Y., Zheng, B., Liu, T.: Towards better UD parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation. In: Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. pp. 55–64. Association for Computational Linguistics, Brussels, Belgium (October 2018), http://www.aclweb.org/anthology/K18-2005
2. Escalante, H.J., Solorio, T., Montes-y Gómez, M.: Local histograms of character n-grams for authorship attribution. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. pp. 288–298. Association for Computational Linguistics (2011)
3. Grieve, J.: Quantitative authorship attribution: An evaluation of techniques. Literary and linguistic computing 22(3), 251–270 (2007)
4. Halteren, H.V.: Author verification by linguistic profiling: An exploration of the parameter space. ACM Transactions on Speech and Language Processing (TSLP) 4(1), 1 (2007)
5. Júnior, P.R.M., de Souza, R.M., Werneck, R.d.O., Stein, B.V., Pazinato, D.V., de Almeida, W.R., Penatti, O.A., Torres, R.d.S., Rocha, A.: Nearest neighbors distance ratio open-set classifier. Machine Learning 106(3), 359–386 (2017)
6. Kešelj, V., Peng, F., Cercone, N., Thomas, C.: N-gram-based author profiles for authorship attribution. In: Proceedings of the conference pacific association for computational linguistics, PACLING. vol. 3, pp. 255–264. sn (2003)
7. Kestemont, M., Tschuggnall, M., Stamatatos, E., Daelemans, W., Specht, G., Stein, B., Potthast, M.: Overview of the author identification task at pan-2018: cross-domain authorship attribution and style change detection. In: Working Notes Papers of the CLEF 2018 Evaluation Labs. Avignon, France, September 10-14, 2018/Cappellato, Linda [edit.]; et al. pp. 1–25 (2018)
8. de Lhoneux, M., Stymne, S., Nivre, J.: Arc-hybrid non-projective dependency parsing with a static-dynamic oracle. In: Proceedings of the The 15th International Conference on Parsing Technologies (IWPT). Pisa, Italy (2017)

9. Markov, I., Stamatatos, E., Sidorov, G.: Improving cross-topic authorship attribution: The role of pre-processing. In: International Conference on Computational Linguistics and Intelligent Text Processing. pp. 289–302. Springer (2017)
10. Niculescu-Mizil, A., Caruana, R.: Predicting good probabilities with supervised learning. In: Proceedings of the 22nd international conference on Machine learning. pp. 625–632. ACM (2005)
11. Nivre, J., Abrams, M., Agić, Ž., et al.: Universal dependencies 2.3 (2018), http://hdl.handle.net/11234/1-2895, LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University
12. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12, 2825–2830 (2011)
13. Peng, F., Schuurmans, D., Wang, S., Keselj, V.: Language independent authorship attribution using character level language models. In: Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 1. pp. 267–274. Association for Computational Linguistics (2003)
14. Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). pp. 2227–2237 (2018)
15. Plank, B., Agić, Ž.: Distant supervision from disparate sources for low-resource part-of-speech tagging. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 614–620. Association for Computational Linguistics (2018), http://aclweb.org/anthology/D18-1061
16. Platt, J., et al.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. Advances in large margin classifiers 10(3), 61–74 (1999)
17. Potthast, M., Gollub, T., Wiegmann, M., Stein, B.: TIRA Integrated Research Architecture. In: Ferro, N., Peters, C. (eds.) Information Retrieval Evaluation in a Changing World - Lessons Learned from 20 Years of CLEF. Springer (2019)
18. Sapkota, U., Bethard, S., Montes, M., Solorio, T.: Not all character n-grams are created equal: A study in authorship attribution. In: Proceedings of the 2015 conference of the North American chapter of the association for computational linguistics: Human language technologies. pp. 93–102 (2015)
19. Sapkota, U., Solorio, T., Montes, M., Bethard, S., Rosso, P.: Cross-topic authorship attribution: Will out-of-topic data help? In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers. pp. 1228–1237 (2014)
20. Stamatatos, E.: A survey of modern authorship attribution methods. Journal of the American Society for information Science and Technology 60(3), 538–556 (2009)
21. Stamatatos, E.: Authorship attribution using text distortion. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers. vol. 1, pp. 1138–1149 (2017)
22. Stamatatos, E., et al.: Ensemble-based author identification using character n-grams. In: Proceedings of the 3rd International Workshop on Text-based Information Retrieval. vol. 36, pp. 41–46 (2006)
23. Straka, M., Straková, J.: Tokenizing, pos tagging, lemmatizing and parsing ud 2.0 with udpipe. In: Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. pp. 88–99. Association for Computational Linguistics, Vancouver, Canada (August 2017), http://www.aclweb.org/anthology/K/K17/K17-3009.pdf