Should Language Inform Model Design for Authorship Attribution?

Abstract

Historical literature is a window of culture and knowledge, from which we derive much of our understanding of the past. Yet, it is often very difficult to determine the authorship of classical texts. Authorship attribution (AA) is time-intensive and tedious, requiring careful analysis to identify patterns that characterize the work of particular authors. Deep Learning models have shown to excel on pattern recognition tasks, and have been used for AA (Bogdanova and Lazaridou, 2014) (Ramezani, 2021) (Murauer and Specht, 2021)(Hedegaard and Simonsen, 2011). However, it is uncommon to find AA models developed for classical texts, and even more uncommon to find them applied to ancient languages. In this paper, we investigate the impact of language on the performance of various published models for AA. We reproduce the work of (Uchendu et al., 2021) and (Ai et al., 2022), but on a corpus of ancient text in Latin that has been professionally translated to English (Grosenthal, 2023). We employ back-translation to augment the data-set and train various models with varying pre-processing procedures. We find that a Multilingual BERT (Devlin et al., 2018) performs the greatest overall with a 93.06% accuracy over both languages, with a character-level CNN performing the best on Latin text (98.64% accuracy), an SVM performing the best on English text (90.56% accuracy), and RNNs performing the poorest over both languages (42.78% accuracy), supporting our hypothesis that optimal model design depends on language. Find our models at NLP-Authorship-Latin-Eng.

1 Introduction

Authorship attribution (AA) of classical texts holds profound significance in our understanding of historical and cultural narratives. These texts are foundational to our modern-day societies and encompass a wide range of genres, including philosophy, literature, and religious scripture. The accurate identification of authors in such texts not only sheds light on the historical context and the intended message but also influences our interpretation of current societal norms and beliefs. For instance, in religious scriptures, where the author's identity can be a matter of theological importance, AA can impact religious practices and doctrines. Understanding who wrote these texts can unravel the intentions and beliefs of historical societies, offering insights that resonate with contemporary cultural and ethical discussions.

While important, manual AA of these texts is a challenging endeavour, often mired in subjectivity and requiring extensive scholarly expertise. The intricate analysis necessary to discern stylistic signatures is both time-consuming and prone to ambiguity. With the advent of machine learning, we have the opportunity to automate and refine this process, leveraging computational models to detect patterns and stylistic markers that might elude human analysis.

Due to a scarcity of data, most applications of machine learning for AA are limited to English text. This is not sufficient for the application of classical literature AA, as classical literature is often in ancient languages.

The differences between languages are not merely different labels for the same words with the same grammatical rules. For instance, Latin declines nouns differently depending on whether they are subjects or objects while English does not-this allows for greater freedom in word order, significantly impacting syntax. This is only one of the many ways that Latin differs from English; these differences are why manual authorship attribution is difficult and time-consuming.

In this paper, we build on previous works that employ language models on English text for authorship attribution, to see how they compare against text in Latin (Bogdanova and Lazaridou, 2014) (Ramezani, 2021) (Murauer and Specht, 2021)(Hedegaard and Simonsen, 2011). We use the Latin-English Translation dataset (Gil Rosenthal, 2023), which consists of passages in Latin and corresponding passages in English. We employ various models used in (Uchendu et al., 2021), including a Multilingual BERT (Devlin et al., 2018) and character-level CNNs (Zhang et al., 2015) and RNNs (Gupta et al., 2019), to discern patterns that are indicative of authorship in these linguistically diverse texts. We employ 3 pre-processing techniques and train a model on each language and each pre-processing technique, for a total of 6 models. We evaluate the models on their accuracy. Our findings reveal intriguing variations in model performance across the two languages, offering insights into the nuanced relationship between language structure and AA.

2 Related Work

Various authors have studied the related task of "Cross-Language Authorship Attribution" (CLAA), which (Bogdanova and Lazaridou, 2014) defines as AA where the model is trained on texts in one language and must make predictions on texts in another language. This differs from our task, because we instead train a language-specific model for each language and pre-processing decision as we aim to determine if state-of-the-art AA models can still perform as good if trained on another language (thus if models should be developed specific to the language at hand or if optimal model structure does not differ across diffrent languages).

Recently R. Ramezani looked at AA in Persian and English datasets as well, comparing how different languages and document lengths contribute to AA (Ramezani, 2021). More similarly to our experiments, Murauer et al. tackle CLAA through shorter social media comments and also test both multi-lingual BERT models and SVMs(Murauer and Specht, 2021).

The works of (Uchendu et al., 2021) and (Ai et al., 2022) explore various models on English AA. These papers focus on enhancing the general performance of models for authorship attribution, with (Uchendu et al., 2021) consisting of a benchmark of common model's and their performances, and (Ai et al., 2022) further examining the effectiveness of the multilingual- BERT proving that it outperfomres all of the other models on AA.

3 Method

3.1 Data Preprocessing and Augmentation Methods

The Latin-English dataset consists of 101,000 professionally translated text pairs from 57 classical books, with a 99%/1%/1% train/validation/test split. We found the distribution of texts to the books they are sourced from to be heavily skewed, as you can see in 1. Hence, we selected the 5 most common books and equalized their numbers. This reduced the dataset size to approximately 4,400 (Table 1), but allowed for us to fairly train the models. We also examined back-translation as a way to augment the text. In this, we increased the size of the dataset to approximately 17,000 entries. To do this, we first translated the texts to Italian, and then back to their respective language, which yielded approximately 8,800 texts. We then further back-translated these texts to French, for a total of 17,352 texts.

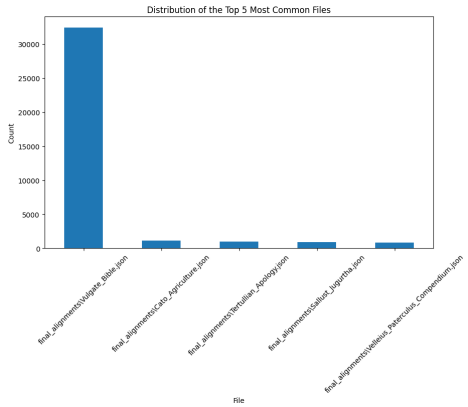

Figure 1: Distribution of the top 5 most frequent files in the training data set before processing.

We noticed many key errors in the Latin-English dataset, so we used the Lingua Language Detector to detect text that was not of the correct language (Stahl, 2023). We filtered the erroneous entries accordingly, in all instances of the data. Additionally, we also expanded all of the contractions in the text. We used the word's movers distance measure to accurately expand multi-semantic contractions in the English text (Beaver, 2023), and the rules listed by (Carey, 2023) to expand the contractions in the Latin text.

For specific pre-processing, we used the NLTK libraries (Loper and Bird, 2002) to lemmatize and remove stop words from the English data, and the CLTK libraries (Johnson et al., 2021) to perform the same on the Latin text .

The pre-processing steps for each data version is summarized on 1.

3.2 Models

In line with the experiments of (Ai et al., 2022), all models- apart from the SVM-used a learning rate of $2e^{-5}$ and were trained for 8 epochs with a batch size of 24. To prevent over-fitting, the mBERT, RNN, and CNN employed early stopping, in which if the validation loss increased for 2 consecutive epochs, the model would stop training. For fairness, we did not reset the models to the weights at their lowest validation loss, but instead kept the weights at what was returned after two consecutive increases of validation loss. Further, the mBERT, RNN, and CNN employed Elastic Net Regularization(Zou and Hastie, 2005), as they would over-fit if this was not done.

For fairness of comparison, no model-specific tuning was done between the models of the same

Data set Name	Contractions	Cased/Uncased	Filtered Erroneous Data	Lemmatized	Stop Words Removed	Size
Unprocessed Unaugmented	Expanded	Depends on model	Filtered	No	No	4,368
Unprocessed Augmented	Expanded	Depends on model	Filtered	No	No	17,252
Lemmatization + SW removed	Expanded	Depends on model	Filtered	Yes	Yes	17,352

Table 1: Dataset Types Employed

nature.

3.2.1 mBERT

The mBERT (Multi-lingual Bidirectional Encoder Representations from Transformers) was originally presented by (Devlin et al., 2018). We base our implementation on the non-contrastive mBERT created by (Ai et al., 2022). We used the 'bert-base-multilingual-uncased' pre-trained model from Hugging Face's Transformers library, which treats capitalized data the same as uncapitalized data. We appended a classifier that consists of a fully connected layer with a ReLU activation and dropout layer for regularization. The drop-out was set to 0.35, and the length of the classifier was 768- as done by(Ai et al., 2022). The loss function used was Cross Entropy loss, with the AdamW optimizer (Loshchilov and Hutter, 2017), a weight decay of 0.0001, and a LASSO coefficient of 0.01. Early stopping was available, but was never needed to be used.

3.2.2 N-gram SVMs with varied N

The support vector machine (SVM) model uses the scikit-learn implementation of the support vector machine classification technique seen in class, specifically `LinearSVC` from `sklearn.svm` (Pedregosa et al., 2011). We used bag-of-n-gram features with a tokenizer that splits strings at spaces. For each model, the n-gram range was optimized using cross-validation, with possible values of (1, 1), (1, 2), and (1, 3). This is a reasonable domain over which to optimize as n-gram ranges starting higher than 1 may cause the model to overlook an important word because it is not in the usual context and n-grams longer than 3 are sparsely distributed in the training data.

3.2.3 CNN

The CNN was implemented via TensorFlow's Keras package which consists of an embedding layer with an input dimension of 128 ASCII characters and an output dimension of 16 (Abadi et al., 2015). This is followed by a 1-D convolutional layer with 128 filters and a kernel size of 5, using the ReLU activation function. Max pooling is used to extract the most significant features, reducing the spatial dimensions of the data. Two dense layers follow, with 320 and 5 neurons, respectively, using the soft-max activation function. The uses the Adam optimizer, and it uses cross-entropy loss as the loss function, with accuracy as the evaluation metric. We set the L2 regularization coefficient to 0.0001 and the LASSO coefficient to 0.001.

3.2.4 RNN

Using TensorFlow, we implemented an RNN consisting of an embedding layer, a bidirectional LSTM layer with 64 units, a dense layer with 64 units with ReLU activation, and a final dense layer with 5 units (corresponding to the number of authors) with softmax activation. We used an L1 regularization coefficient of 0.0001 and a LASSO coefficient of 0.01.

4 Results

4.1 Overall Performance Accuracy Trends

As evident in 2, the best-performing model overall was the mBERT, with a highest average accuracy across all pre-processing steps and both languages. It proved to be the most reliable, albeit it is the most computationally expensive model.

The worst performing model overall was the RNN. It's performance varied significantly based on the pre-processing technique used on each language.

The models generally performed worse on English text than they did on Latin text. The highest accuracy was lower on English data than on Latin data, along with every average accuracy. Further, the lowest accuracy was lower on English text.

Model	Pre-processing	Latin	English	Average Over Both Languages
mBERT	Unprocessed Unaugmented	96.67%	89.44%	**93.06%**
mBERT	Unprocessed Augmented	96.94	89.17%	**93.06%**
mBERT	Augmented+Lemmatization and Stop words Removed	94.75%	89.50%	92.13%
mBERT	Average	96.12%	89.37%	92.75%
RNN	Unprocessed Unaugmented	**27.78%**	57.78%	**42.78%**
RNN	Unprocessed Augmented	76.67%	54.72%	65.69%
RNN	Augmented+Lemmatization and Stop words Removed	74.31%	**24.03%**	49.17%
RNN	Average	59.58%	45.51%	52.55%
CNN	Unprocessed Unaugmented	**98.64%**	82.33%	90.49%
CNN	Unprocessed Augmented	97.44%	75.37%	86.40%
CNN	Augmented+Lemmatization and Stop words Removed	96.34%	78.93%	87.64%
CNN	Average	97.48%	78.88%	88.18%
SVM	Unprocessed Unaugmented	90.83%	**90.56%**	90.69%
SVM	Unprocessed Augmented	87.22%	88.33%	87.78%
SVM	Augmented+Lemmatization and Stop words Removed	88.40%	85.08%	86.74%
SVM	Average	88.82%	87.99%	88.40%

Table 2: Comparison of model accuracies across different pre-processing strategies. The highest accuracy per column is highlighted in blue and the lowest in red, indicating the models' performance variability with respect to language and pre-processing.

4.2 Latin Accuracy Trends

The CNN outperformed every other model on Latin text data, with the RNN performing the worst overall, but with its performance varying drastically based on the pre-processing method used- with a difference of 48.89% between pre-processing techniques.

4.3 English Accuracy Trends

The mBERT performed the best on English text, with a 10.49% increase in average accruacy from the CNN, which performed 33.37% better on average than the RNN.

5 Discussion and Conclusion

5.1 Variation of RNN Performance

The RNN's poor performance on the unprocessed unaugmented Latin data and the pre-processed augmented English data is likely because of the overly sequential nature of our RNN architecture and the over-adaptation of the data.

The RNN likely performed very poorly on the unaugmented Latin text because its sequential nature does not fare well with the relatively unconstrained word ordering of Latin. Back-translating the Latin text likely anglicized the word ordering of the passages, making it more suitable for the right-ward direction of the RNN.

The performance of the RNN on English text decreased with more processing. This is likely due to both the back translation impacting word order negatively, as back translation tends to distort grammar, and the lemmatization and stop word removal eliminating words that indicate style and context. This can also be seen for the Latin text, in that the accuracy of the lemmatized and stop word removed data was lower than that of the unprocessed unaugmented data.

5.2 Effects of Pre-processing Decisions

While the impact of lemmatization and stop word removal is unclear on English text, on Latin text, stop word removal and lemmatization consistently reduced accuracy. This is because Latin, with its rich inflectional morphology, relies heavily on word endings to convey syntactic and semantic information. Lemmatization, which reduces words to their base forms, can strip away these crucial grammatical cues. Furthermore, Latin's use of function words (many of which are considered stop words) is integral to its sentence structure and meaning. Removing these words can lead to a significant loss of contextual information, hindering the model's ability to accurately attribute authorship.

5.3 mBERT Performance

The strong performance of the mBERT is likely due to its multi-lingual encoder being pre-trained on 104 languages (Devlin et al., 2018). While it's pre-training was not suited for AA, it still familiarized the model with the structures of various languages. Hence, it's model is flexible to morphology and word orderings, making it generalize well to languages where the word order varies.

5.4 Strengths

While assessing model performance based on their ability to predict text titles from passages does not address cases in which the author is not known, it helps point towards the writing styles of the passages.

Back-translation proved to work sufficiently. Although it did not increase accuracy in all cases, it tended to increase accuracy more than it would decrease accuracy- most notably the improvement it made to the RNN on Latin text. This is important as sourcing enough data was our greatest challenge due to the lack of data-sets that exist that comprise of classic literature and is multi-lingual, so it is likely that other researchers too face the same problem. While imperfect, back-translation makes sourcing enough data easier.

Our findings obtained very high accuracies using the mBERT, highlighting its ability to be used for authorship attribution.

Further, the high accuracies achieved by the relatively computationally inexpensive SVM and CNN are promising, as they show that a very large sophisticated model is not a requirement for cross-language authorship attribution.

5.5 Limitations

While most of the models performed well across the languages, this remains a 5-class classification problem. Practical classical authorship attribution would require a model trained on a wide plethora of texts.

Additionally, the mBERT converted all of the text data to lower-case, while the other models did not. This makes the results less significant. If we were to re-do the experiment, we would examine the impact of case on accuracy.

Further, the inclusion of the Bible as one of the books may have skewed the results, as religious scripture tends to have a apparent tone, and uses certain words like the names of prophets, that make it easy to classify between literary works.

Additionally, the test data used was disproportionate, in that most passages point to one of the books. This can skew our results, as a more appropriate test dataset would be one in which the distribution of the frequency of books is more balanced. We balanced the validation set, but at the cost of reducing its size significantly.

We also noticed that many entries in the training data consisted of only roman numerals (in both the English and Latin sections), which would be associated with the same book. Data like this skews the models by making them associate the numerals with a specific text. Had we done this experiment again, we would too filter out texts that only consist of roman numerals.

5.6 Future work

We could extend our work by evaluating the use of a contrastive loss function. Contrastive loss functions have been shown to make data increasingly separable, and improve performance on AA (Liu et al., 2023) & (Ai et al., 2022).

6 Conclusion

Our study supports our hypothesis by showing that language-specific attributes play an important role in determining model design for AA. In particular, sequential-input models should not be used for languages where word-ordering is not linear. Further, flexible models that have been pre-trained on multiple languages prove to be the best performing.

7 Statement of Contributions and Repository

All members participated in extensive group discussions about the design of this experiment. All contributed to the writing. All members contributed equally to conducting experiments. Our GitHub repository can here: NLP-Authorship-Latin-Eng

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan,

Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Bo Ai, Yuchen Wang, Yugin Tan, and Samson Tan. 2022. Whodunit? learning to contrast for authorship attribution.

Ian Beaver. 2023. pycontractions. Accessed: December 19, 2023.

Dasha Bogdanova and Angeliki Lazaridou. 2014. Cross-language authorship attribution. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2015–2020, Reykjavik, Iceland. European Language Resources Association (ELRA).

William L. Carey. 2023. Latin contractions. Accessed: December 19, 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Gil Rosenthal. 2023. $\text{latin}_{e}nglish_{t}ranslation(revision93aa4f6)$.

Grosenthal. 2023. grosenthal/$\text{latin}_{e}nglish_{t}ranslation$. Accessed: December 19, 2023.

Shriya TP Gupta, Jajati Keshari Sahoo, and Rajendra Kumar Roul. 2019. Authorship identification using recurrent neural networks. In *Proceedings of the 2019 3rd International Conference on Information System and Data Mining*, ICISDM '19, page 133–137, New York, NY, USA. Association for Computing Machinery.

Steffen Hedegaard and Jakob Simonsen. 2011. Lost in translation: Authorship attribution using frame semantics. volume 2, pages 65–70.

Kyle P. Johnson, Patrick J. Burns, John Stewart, Todd Cook, Clément Besnier, and William J. B. Mattingly. 2021. The Classical Language Toolkit: An NLP framework for pre-modern languages. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 20–29, Online. Association for Computational Linguistics.

Shangqing Liu, Bozhi Wu, Xiaofei Xie, Guozhu Meng, and Yang Liu. 2023. Contrabert: Enhancing code pre-trained models via contrastive learning.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. Published as a conference paper at ICLR 2019.

Benjamin Murauer and Gunther Specht. 2021. Small-scale cross-language authorship attribution on social media comments. In *Proceedings of the 4th Workshop on Technologies for MT of Low Resource Languages (LoResMT2021)*, pages 11–19, Virtual. Association for Machine Translation in the Americas.

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.

Reza Ramezani. 2021. A language-independent authorship attribution approach for author identification of text documents. *Expert Systems with Applications*, 180:115139.

Peter M. Stahl. 2023. lingua-language-detector. Accessed: December 19, 2023.

Adaku Uchendu, Zeyu Ma, Thai Le, Rui Zhang, and Dongwon Lee. 2021. TURINGBENCH: A benchmark environment for Turing test in the age of neural text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2001–2016, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Hui Zou and Trevor Hastie. 2005. Regularization and Variable Selection Via the Elastic Net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 67(2):301–320.