Authorship Attribution in 19th-century Philippine Literature Using A Deep Learning Multi-label Classifier

Paolo Espiritu Jason Jabanes Charibeth Cheng
De La Salle University - Manila, Manila, Philippines
{paolo_edni_espiritu_a, jason_jan_c_jabanes, charibeth.cheng}@dlsu.edu.ph

Abstract

Authorship attribution (AA) is an essential task in Natural Language Processing (NLP) that plays a crucial role in historical literary analysis, intellectual property protection, digital forensics, document identification, and plagiarism detection. Despite recent advancements for high-resource languages, AA for low-resource languages remains underexplored due to the lack of annotated datasets. This study aims to address this gap by focusing on 19th-century Filipino literary texts. To facilitate this, we introduce Panitikan, a publicly available, pre-processed dataset of Filipino literary texts. Given the complex morphological structure of the Filipino language, we discuss various preprocessing techniques designed to enhance model performance. We employed a closed-set multi-label classification approach using Long Short-Term Memory (LSTM), Convolutional Neural Networks (CNN), and fine-tuned RoBERTa-TL models (Base and Large) tailored for Tagalog. The models were evaluated using accuracy, precision, recall, and F1 score metrics. Our results demonstrate that on a 10-author dataset, the RoBERTa-TL-Large model achieved the highest F1 score (96.45%), outperforming LSTM (82.40%), CNN (74.95%), and RoBERTa-TL-Base (95.78%). On a more extensive 34-author dataset, RoBERTa-TL-Large maintained superior performance with an F1 score of 92.81%, followed by RoBERTa-TL-Base (85.87%), LSTM (55.23%), and CNN (48.30%).

1 Introduction

Authorship attribution (AA) is a classification task aimed at identifying the true author of a given text from a set of potential candidates. This task has gained significant attention due to its practical applications in areas such as historical literature analysis, digital forensics, document identification, plagiarism detection, and more (Reisi and Mahboob Farimani, 2020; Fabien et al., 2020; Theophilo et al., 2022). However, most research in this field has focused on high-resource languages, largely due to the availability of expertly annotated datasets that facilitate model development and validation. In contrast, there remains a significant need for developing datasets and methodologies tailored to low-resource languages (Nitu and Dascalu, 2024). Recent advancements in Natural Language Processing (NLP) offer various methodologies that can be adapted to address the unique challenges associated with AA in these languages (He et al., 2024).

For instance, a study by Fedotova et al. (2022) explored authorship attribution for Russian texts, including social media and literary works, using a variety of machine learning models, neural networks, and hybrid approaches such as Support Vector Machines (SVM), fastText, Convolutional Neural Networks (CNN), Long Short-Term Memory networks (LSTM), and the Bidirectional Encoder Representations from Transformers (BERT). These models were trained in a closed-set scenario, meaning they could only classify texts authored by a limited, predefined set of individuals. The study found that deep neural networks achieved the highest average accuracy of 82.3%, followed closely by fastText at 82.1% and SVM with a genetic algorithm at 80.4% (Fedotova et al., 2022).

The success of such models often hinges on the availability of high-quality, expertly annotated corpora, which are frequently lacking in low-resource settings. For example, a study on the Romanian language created a corpus from Romanian stories comprising of 1,263 texts and 12,516 paragraphs written by 19 authors (Nitu and Dascalu, 2024). They employed preprocessing techniques specific to the Romanian language to enhance model training. They utilized a hybrid model combining top predictive linguistic features (selected using the Kruskal-Wallis mean rank) with a fine-tuned Romanian BERT model, achieving state-of-the-art F1 scores of 0.87 on full texts and 0.77 on paragraphs (Nitu and Dascalu, 2024).

In this study, we explore authorship attribution for 19th-century Filipino literary works. To our knowledge, this is the first study to investigate AA in Filipino historical texts. The complexity of the Filipino language, characterized by its intricate morphology and syntax, necessitated the implementation of unique preprocessing techniques. The models were trained in a closed-set configuration to limit predictions to a specific set of authors. Although this study focuses on the Filipino language, the methodologies discussed can be adapted for other low-resource languages.

This paper aims to contribute the following:

1. Publicly available Filipino literature *Panitikan* dataset with 2 versions, which contain 10 authors with 19 written works and 34 authors with 47 written works.

2. Trained LSTM and CNN models and the fine-tuned RoBERTa-Tagalog (TL) models (Cruz and Cheng, 2022) to identify 19th-century Filipino authors based on the given literary text.

2 Related Works

Traditionally, AA has mostly relied on a manual method to extract elements pertaining to an author's style or substance. However, in recent times, deep learning methods have been employed for AA tasks as these are expected to automatically capture stylometric features of the text (Chowdhury et al., 2019). These different approaches have been used to conduct AA for languages such as English, Russian, and Bengali. However, the same cannot be said about the advancements made in Philippine Literature. This section examines novel studies that employ deep learning methods, such as neural networks and transformers, to perform AA. Additionally, current state of AA in Philippine literature will also be explored.

2.1 Deep Learning-based Approaches

Chowdhury et al. (2019) used fastText's word embedding model with Convolutional Neural Networks (CNN) to investigate AA in Bengali literature. The study was able to demonstrate that CNN models could accurately capture stylistic subtleties in Bengali text, achieving an accuracy of 92% on their dataset. Kapočiūtė-Dzikienė et al. (2015) focused on age and gender characteristics in author profiling of Lithuanian literary texts. They achieved a 89.2% accuracy with the Naive Bayes Multinomial method and character tri-grams. The study by Fabien et al. (2020) is one of the first efforts to perform author classification by fine-tuning a pre-trained BERT model. Their approach outperformed traditional machine learning models by 2.7% and set a new benchmark for the IMDB dataset. The study was able to show that Transformer models was able to reach competitive results across three different benchmark datasets, even with large amounts of authors.

2.2 Authorship Attribution in the Philippines

Dumalus and Fernandez (2011) explores the use of writer's rhythm as a stylometric feature, achieving a 50% accuracy using a Naive Bayesian classifier. The study considers this result significant enough to suggest that rhythm can be considered as a viable style marker. It is worth noting that, while the study was conducted in the Philippines, the corpora used does not contain any Filipino text data. Marvin Imperial (2021) examined the stylistic writing of potential pedophiles and child sex traffickers in the Philippines using Twitter as their main source of data. The findings demonstrate that child traffickers and peddlers often employ the same terminologies. Furthermore, the study used these co-occurring terminologies to build four different online personas that characterize a pedophile.

3 System Design and Architecture

3.1 Overview of the System

Figure 1 shows the model training pipeline used for LSTM, CNN, RoBERTa-TL-Base, and RoBERTa-TL-Large. It shows the step-by-step process of creating a multi-label classification model using the aforementioned deep learning architectures. As observed in Figure 1, the trained models often shared the same processes and only diverged after tokenizing the dataset.

3.1.1 Panitikan Corpus

The pre-processed dataset, which contains the features and labels, was loaded to train the models for the multi-label classification task.

3.1.2 Extract labels and input columns

The necessary features and labels were selected in preparation for the training process.

3.1.3 Split into train/test/validation sets

The dataset was split into 80:10:10, respectively, using the *datasets* library from Hugging Face.

3.1.4 Encode with tokenizer

A tokenizer was used to encode the dataset into a numerical format for computational efficiency.

3.1.5 Fine-tuning (RoBERTa Tagalog Models)

Since RoBERTa-TL-Base and RoBERTa-TL-Large models were already pre-trained on the Tagalog language, it was only necessary to perform fine-tuning using the *Panitikan* dataset.

3.1.6 Train Word2Vec Model (LSTM & CNN)

A skip-gram word2vec was trained using the train set that will serve as the embedding layer for LSTMs and CNNs.

3.1.7 Hyperparameter Tuning (LSTM & CNN)

Hyperband tuning was used in selecting optimal hyperparameter configurations for the LSTM and CNN models.

3.1.8 Model Training (LSTM & CNN)

The LSTM and CNN models were trained with a batch size of 32 on 10 epochs. The models were then saved for evaluation and inference.

3.1.9 Multi-label Classification Model

After training or fine-tuning, the best model is saved into a local directory. This step is important to prevent restarting the entire pipeline when evaluating or inferencing.

3.1.10 Evaluation and Inference

The model is evaluated in terms of accuracy, precision, recall, and F1 score. It may now also be used to test custom inputs.

3.2 Convolutional Neural Network (CNN)

CNN is a deep learning architecture popularized due to its numerous practical applications such as recommendation systems, facial recognition, speech and text processing, and more (Alzubaidi et al., 2021). It consists of multiple layers, including the input, convolution, pooling, fully connected, and output. As the input sequence goes through each layer, a series of matrix multiplications and subsampling operations are performed before evaluating the features to generate an output (Alzubaidi et al., 2021).

3.3 Long Short-Term Memory (LSTM)

Another known neural network in NLP is LSTM which was created to handle vanishing gradient issues experienced by traditional recurrent neural RNNs. It excels in a variety of tasks due to its capability to learn when to retain and forget information. To achieve this, it implements three gates: (1) forget, (2) input, and (3) output. The forget gate is responsible for discarding the information from the previous state by assigning the previous and current input to a rounded value between 0 (discard) and 1 (save). Furthermore, the input gate chooses which new information to store in the current state using the same algorithm as forget gates. Finally, the output gates determine which information to output from the current state (Fedotova et al., 2022).

3.4 Robustly Optimized BERT Approach (RoBERTa)

To achieve state-of-the-art performance, BERT models are often used due to their self-attention mechanism to process sequences of text and produce contextualized word embeddings. Its superior results may also be attributed to its bi-directional capability, allowing it to capture a wider context and better understand the semantic meaning of the token (Fedotova et al., 2022).

Given the strengths of the BERT model, its variant, RoBERTa, has demonstrated better performance (Naseer et al., 2021; Rajapaksha et al., 2021; Adoma et al., 2020). RoBERTa was created by optimizing BERT in terms of its training pipeline and data (Liu et al., 2019).

In this study, two Filipino pre-trained transformers, RoBERTa-TL-Base and RoBERTa-TL-Large, will be used. The models will be fine-tuned on the constructed dataset to classify the true author with the corresponding text.

3.5 Implementation Details

For the LSTM and CNN models, training was performed using the Tesla V100-PCIE-32GB. On the other hand, NVIDIA RTX 6000 Ada Generation was utilized to train RoBERTa-TL models by renting a GPU from vast.ai. This ensured that model training would not be prematurely terminated due to memory limitations.

The models were written in a Jupyter notebook to sufficiently document each step for replicability. The software libraries used to train and evaluate the models are illustrated in Table 1.

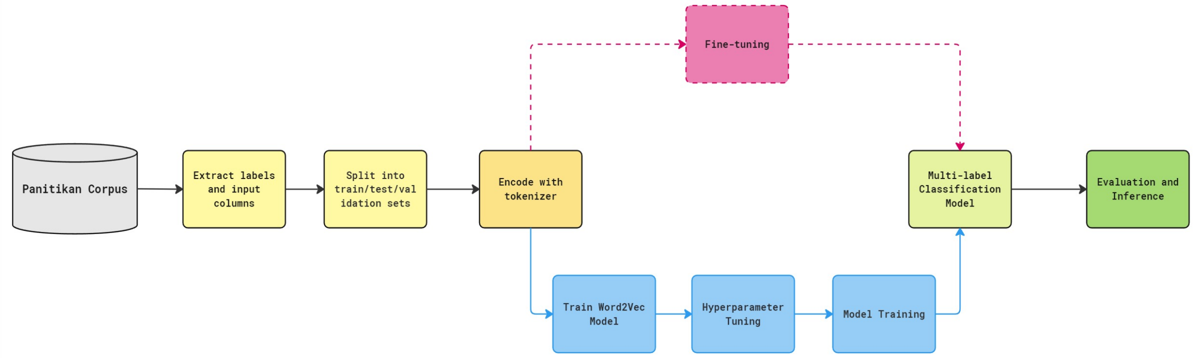


Figure 1: Model training pipeline for RoBERTa-TL , LSTM, and CNN

Table 1: Software Libraries

Transformers	NLTK	Tensorflow
Datasets	Scikit-learn	Keras
Torch		

4 Methodology

4.1 Data Collection

A web scraper was constructed to extract various 19th-century and early 20th-century Filipino literary works, representing novels, poems, and short stories. The scraper was developed with Python using the `scrapy` library. The data originated from Project Gutenberg, which provides more than 60,000 free eBooks, many of which are literary and historical works (Lebert, 2008). The web scraper was successful in obtaining 60 literary works written by 45 distinct authors from Project Gutenberg.

4.2 Data Preprocessing

4.2.1 Initial Filtering

The dataset was first filtered to only include original literary works written exclusively in the Filipino language. Dictionaries, thesauruses, and works translated from other languages were excluded, leaving only 47 literary works from the original 60.

4.2.2 Data Cleaning

The first steps in data cleaning were standardizing text format and eliminating unnecessary information. The literary works were stripped of Project Gutenberg information, such as initial descriptions and transcriber remarks. Textual normalization converted UTF-8 characters (á, é, ï, ó, ë, ü, ñ) into their corresponding ASCII characters. Additionally, elements such as bracketed text ([]{}), punctuations (excluding sentence delimiters; more on this later), Roman numerals, numbers, numbers with periods and commas, and capitalized words (which are almost always titles, headings, or dialogue indicators in literature) were removed. To further standardize the text, all text were converted to lowercase and extra whitespaces were eliminated.

4.2.3 Sentence Tokenization

To correctly tokenize text into sentences, the first step is to detect abbreviations that terminate in periods. This is necessary since periods often indicate sentence boundaries. Words with periods that appear more than once in a text and were less than six characters were filtered to identify possible abbreviations. Following their identification, these abbreviations were not included in the sentence splitting procedure. The text was then segmented into separate sentences based on standard punctuations using NLTK's (Bird et al., 2009) sentence tokenization tool, with the previously noted abbreviations being treated as exceptions. Afterwards, punctuations that were used to segment into sentences were removed. Duplicate sentences from the same author were also identified and removed. Lastly, sentences with less than 10 characters were removed as they tend to be more or less meaningless.

4.3 Document Representation

The corpus, henceforth referred to as the *Panitikan* corpus, is presented in two versions: (a) the entire corpus from 34 authors and comprising 47 literary works; (b) a subset with 10 authors and consisting of 19 literary works. The ten authors in the subset were chosen for having the most token counts in the corpus. This was created to evaluate a dataset with a more balanced data distribution. Specifications for the *Panitikan* corpus are presented in Table 2.

Table 2: Specifications of the *Panitikan* corpus

Corpus	**Items**	**Count**
Corpus Size (34 Authors)	No. of tokens	724,133
	Vocabulary Size	60,354
	No. of literary works	47
	No. of authors	34
Corpus Size (10 Authors)	No. of tokens	458,254
	Vocabulary Size	41,210
	No. of literary works	19
	No. of authors	10

The table illustrates that the entire corpus is composed of 724,133 tokens having a vocabulary size of 60,354 (unique tokens). The subset of 10 authors is composed of 458,254 tokens having a vocabulary size of 41,210.

For this study, two document representations were used to examine the effect of contextual information on AA. In the first approach, individual sentences are treated as a single document. This technique evaluates how well an author's distinctive style indicators can be recognized in the context of a single phrase. On the other hand, the second method defines documents as text chunks that are about 1000 characters long, or about equivalent to a paragraph. The objective is to analyze these varied document lengths in order to assess if a more comprehensive context is required in order to correctly distinguish between writers according to their styles of writing. Document counts for the different document representations across the different corpus are shown in Table 3.

Table 3: Document Count of different representations

Corpus	**Representation**	**Doc. Count**
Corpus Size (34 Authors)	Sentence	38,340
	1000-character chunks	4,965
Corpus Size (10 Authors)	Sentence	25,026
	1000-character chunks	3,114

4.4 Experimental Setup

In this study, the AA is treated as a classification problem. To accommodate the distinct requirements of different models, two text preprocessing pipelines were employed. Wherein each author in the dataset represents a class, the goal of these deep learning models is to predict the class of a test document.

Deep learning models will be trained on preprocessed text that had been lowercased and punctuations removed. Conversely, the BERT model, which benefits from preserving linguistic subtleties, was fed with the cleaned text data containing punctuations and word casing. For text encoding, we used One-Hot Encoding to turn category data into binary vectors. Using this technique, a vector of zeros is created, with a single one at the index that represents the presence of a specific category.

The input texts were also encoded using two distinct libraries. The TensorFlow Keras Tokenizer was employed for the LSTM and CNN models, while the RoBERTa-TL models utilized a `tokenizer` from Hugging Face. This encoding process assigns unique numerical identifiers to each token, a crucial step that optimizes the models' ability to analyze and comprehend human language more effectively.

4.5 Training and Hyperparameters

In all experiments, we adopted an 80/10/10 train/validation/test split. For the word embeddings, we proceeded with skip-gram word level embeddings by word2vec. To generate the word vectors, a vector dimension of length 300 and context window of 5 were used.

The pre-trained Word2Vec embeddings created will be used as an embedding layer when training the deep learning models. This was only applied for both LSTM and CNN. The models' output layer employed a Softmax activation function for multi-class classification, with categorical cross-entropy as the loss function. Model optimization was achieved using the Adam optimizer, and accuracy was the primary evaluation metric.

Hyperband tuning was used in selecting the optimal hyperparameter configurations for the LSTM and CNN models, while standard hyperparameter values were used for RoBERTa-TL models. Hyperparameter configurations for each model is presented in Table 4.

LSTM and CNN models were trained on a Tesla V100-PCIE-32GB GPU. On the other hand, the RoBERTa-TL models were trained on a NVIDIA RTX 6000 Ada Generation GPU.

After training the models, the test data will be used to measure the models' performance in pre-

Table 4: Hyperparameter configurations for each model

Model	**Parameter**	**Value**
LSTM	LSTM units	50
	Batch size	32
	Epochs	10
CNN	Conv1D Filters	256
	Conv1D Kernel Size	5
	MaxPooling1D Pool Size	5
	Dense Layer Units	128
	Dropout rate	0.2
	Learning rate	0.001
	Batch size	32
	Epochs	10
RoBERTa-TL	Weight decay	0.01
	Learning rate	0.00002
	Batch size	8
	Epochs	10

Table 5: Results of AA on Deep Learning techniques for *Panitikan* corpus

		10 Authors		34 Authors	
Model	**Measure**	**SEN**	**PARA**	**SEN**	**PARA**
LSTM	Accuracy	0.799	0.865	0.656	0.672
	Precision	0.793	0.840	0.542	0.519
	Recall	0.787	0.827	0.587	0.549
	F1 score	0.786	0.824	0.552	0.509
CNN	Accuracy	0.714	0.828	0.591	0.643
	Precision	0.751	0.769	0.496	0.508
	Recall	0.692	0.755	0.460	0.510
	F1 score	0.699	0.749	0.461	0.483
RoBERTa-TL-Base	Accuracy	0.846	0.949	0.761	0.795
	Precision	**0.860**	0.967	**0.823**	0.934
	Recall	0.850	0.949	0.766	0.795
	F1 score	**0.855**	0.958	**0.793**	0.858
RoBERTa-TL-Large	Accuracy	**0.848**	**0.961**	**0.764**	**0.895**
	Precision	0.858	**0.968**	0.817	**0.963**
	Recall	**0.851**	**0.961**	**0.768**	**0.895**
	F1 score	0.854	**0.965**	0.791	**0.928**

[1]SEN = Sentence-level features, PARA = 1000-character chunk features

dicting the author of the text. Measures such as accuracy, precision, recall, and F1-score were used.

5 Results and Analysis

In this section, the results of the deep learning techniques on the task of author identification are discussed. Table 5 presents the results of all the deep learning techniques for each document representation according to the corpus size. In addition, Figures 2a, 2b, 2c, and 2d visualize the data presented in Table 5 using a grouped bar chart.

It can be observed that the RoBERTa-TL models significantly outperform the LSTM and CNN models in our experiments, with a 10~17% increase across all metrics for the different features. This superior performance is likely due to the transformer-based architecture, which uses self-attention mechanisms to better extract contextual information and intricate patterns from the text. Additionally, RoBERTa-TL is pre-trained on a larger Filipino dataset. Despite the *Panitikan* corpus's use of older Filipino forms and spellings (e.g. '*huag*' instead of '*huwag*'), RoBERTa-TL successfully catches the text's subtleties, proving its strong ability to manage variances in language form and style.

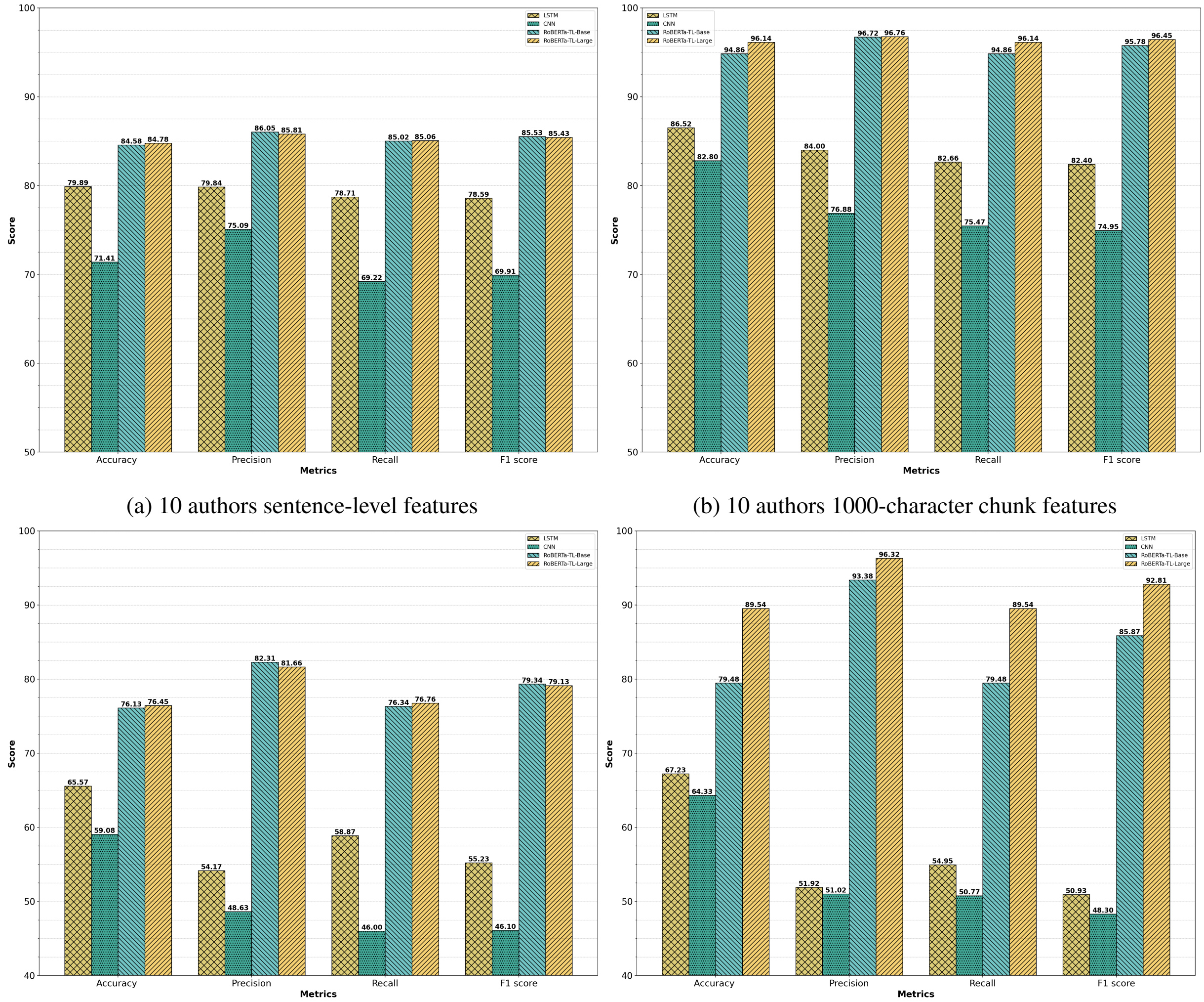
It is also worth mentioning that RoBERTa-TL-Large showed the best results for all experiments, as seen in Figure 2. Despite being trained on 34 labels, the model managed to achieve an F1 score of 92.81% on paragraph features. This is a 6.94% F1 score difference compared to RoBERTa-TL-Base despite having similar scores on sentence-level features. With this, it can be stated that the strengths of RoBERTa-TL-Large are fully utilized when using paragraph features, as it showed significantly better performance than other models.

Based on Table 5, we evaluate the performance of the specialized neural networks on word2vec, specifically LSTM and CNN. CNNs are mainly utilized for image processing because of their pattern detection capabilities (Ruder et al., 2016). Since sentences also have a sequential dimension, CNNs are able to effectively capture the context and stylistic elements of different authors. Despite this, CNN only achieves an accuracy of 59.1% at the sentence level on the test dataset for 34 authors.

In contrast, the LSTM can retain memory by using its prior output as one of its inputs (Zaremba et al., 2014). Additionally, the gating mechanisms in LSTM assist in filtering out less significant information, enabling the model to extract relevant features that identify an author's style (Zaremba et al., 2014).

With this, the LSTM model significantly outperformed the CNN model on all evaluation metrics on the test set. Specifically, for the test dataset with 34 authors at the sentence level, the LSTM achieves an accuracy of 65.6%. While this is higher than the CNN, both models underperform when trained and tested on the full corpus of 34 authors. This

1



(a) 10 authors sentence-level features

(b) 10 authors 1000-character chunk features

(c) 34 authors sentence-level features

(d) 34 authors 1000-character chunk features

Figure 2: Authorship attribution performance in terms of Accuracy, Precision, Recall, and F1 score

underperformance may be due to the difficulty in distinguishing between authors who have very similar writing styles, which might have confused the models. Additionally, it is worth noting that the data was somewhat imbalanced, which might have caused the models to be biased toward the authors with the most entries.

Based on the comparison of the two corpora (10 authors vs. 34 authors), it can be observed that the F1 scores achieved by the LSTM, CNN, and the RoBERTa-TL models for the subset of 10 authors are substantially higher than the full corpus of 34 authors. Since there are fewer authors, there is less intricacy and writing style overlap, which makes it simpler to discern between the subtleties in their vocabulary and writing style. Additionally, the models may benefit from a deeper understanding of the language nuances present in the smaller set, allowing for more effective differentiation between the authors' writing styles. As the number of authors increases, the task becomes more challenging due to the increased variability and similarity in writing.

When comparing the sentence-level features and the paragraph features, it is shown that all deep learning models produced the highest F1 score using paragraph features. This suggests that longer contexts might provide more information to differentiate the author's writing styles. However, an exception is observed for LSTM when classifying 34 authors, where the sentence feature outperformed the paragraph feature. This might be due to the paragraph feature with 34 authors producing more noise than clarity, making it more challenging for the LSTM model to classify the authors. This implies that while longer contexts often provide more information, the model's capacity to use it will rely on its architecture and the specific task.

6 Conclusion

This study contributes to the field of authorship attribution (AA) by focusing on the Filipino language. We developed the Panitikan corpus, a Philippine literature dataset representing 19th-century to early 20th-century works. The corpus includes 724,133 tokens across 47 literary works attributed to 34 different authors.

For feature selection, we explored both sentence-level and paragraph-level features, and compared the performance of models trained on a subset of 10 authors against those trained on the full 34-author dataset. One of the study's key contributions is the use of fine-tuned RoBERTa-Tagalog models, which were benchmarked against deep learning models such as Long Short-Term Memory (LSTM) networks and Convolutional Neural Networks (CNN). The RoBERTa-Large model achieved the highest performance, with an accuracy of 0.961 and an F1-score of 0.965 on the 10-author dataset when using paragraphs as input. For the 34-author dataset, the RoBERTa-Large model reached an accuracy of 0.895 and an F1-score of 0.928, outperforming CNN and LSTM models by 10-17

Our findings suggest that reducing the number of authors from 34 to 10 improves model accuracy and F1-score, likely due to the more balanced data distribution with the top 10 authors having the most tokens. Additionally, using paragraph-level inputs with 1000-character chunks resulted in better performance than sentence-level inputs, possibly because longer contexts provide more information to distinguish authors' writing styles and reduce input size variability.

This study represents the first attempt to implement AA specifically for Filipino literary texts. While our focus was on applying deep learning models to this context, our findings have broader implications. They contribute to the understanding of text analysis in the Filipino language, aid in the historical analysis of documents to verify authorship, and support literary studies by identifying authorial style.

7 Recommendations

To further enhance AA research in Filipino literary works, several recommendations can be made:

1. ***Expand the Dataset***. Increasing the dataset size by including more works from the same authors could help models better capture an author's entire range of writing styles, rather than being limited to individual pieces.

2. ***Incorporate Contemporary Works***. Including more recent literary works could allow for a comparative analysis between classical and modern writing styles, providing deeper insights into evolving authorship patterns.

3. ***Improve Data Balancing Techniques***. As the dataset grows, developing more efficient data balancing techniques will be crucial to minimize biases and ensure that models learn from a diverse set of texts.

4. ***Explore Paragraph-Level Features***. Further research into paragraph-level features is recommended. Testing different chunk sizes (both longer and shorter) could yield better results in distinguishing writing styles.

5. ***Experiment with Word Embeddings and Model Architectures***. Investigating different word embeddings, such as FastText or GLoVe, might improve model performance. Additionally, combining CNN and LSTM networks could potentially enhance results by leveraging the strengths of both architectures.

6. ***Explore Advanced Models and Attention Mechanisms***. Future research should consider experimenting with other Transformer-based models or advanced attention mechanisms. These models might achieve comparable or even superior performance to our current best metrics, thereby improving AA in Filipino literary texts.

References

Acheampong Francisca Adoma, Nunoo-Mensah Henry, and Wenyu Chen. 2020. Comparative analyses of bert, roberta, distilbert, and xlnet for text-based emotion recognition. In *2020 17th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, pages 117–121.

Laith Alzubaidi, Jinglan Zhang, Amjad J Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, José Santamaría, Mohammed A Fadhel, Muthana Al-Amidie, and Laith Farhan. 2021. Review of deep learning: concepts, cnn architectures, challenges, applications, future directions. *Journal of big Data*, 8:1–74.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.".

Hemayet Ahmed Chowdhury, Md Azizul Haque Imon, Syed Md Hasnayeen, and Md Saiful Islam. 2019. Authorship attribution in bengali literature using convolutional neural networks with fasttext's word embedding model. In *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, pages 1–5. IEEE.

Jan Christian Blaise Cruz and Charibeth Cheng. 2022. Improving large-scale language models and resources for Filipino. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6548–6555, Marseille, France. European Language Resources Association.

A Dumalus and P Fernandez. 2011. Authorship attribution using writer's rhythm based on lexical stress. In *11th Philippine Computing Science Congress, Naga City, Philippines*.

Maël Fabien, Esaú Villatoro-Tello, Petr Motlicek, and Shantipriya Parida. 2020. Bertaa: Bert fine-tuning for authorship attribution. In *Proceedings of the 17th International Conference on Natural Language Processing (ICON)*, pages 127–137.

Anastasia Fedotova, Aleksandr Romanov, Anna Kurtukova, and Alexander Shelupanov. 2022. Authorship attribution of social media and literary russian-language texts using machine learning methods and feature selection. *Future Internet*, 14(1).

Xie He, Arash Habibi Lashkari, Nikhill Vombatkere, and Dilli Prasad Sharma. 2024. Authorship attribution methods, challenges, and future research directions: A comprehensive survey. *Information*, 15(3).

Jurgita Kapočiūtė-Dzikienė, Andrius Utka, and Ligita Šarkutė. 2015. Authorship attribution and author profiling of lithuanian literary texts. In *The 5th Workshop on Balto-Slavic Natural Language Processing*, pages 96–105.

Marie Lebert. 2008. Project gutenberg (1971-2008).

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Joseph Marvin Imperial. 2021. How do pedophiles tweet? investigating the writing styles and online personas of child cybersex traffickers in the philippines. *arXiv e-prints*, pages arXiv–2107.

Muchammad Naseer, Muhamad Asvial, and Riri Fitri Sari. 2021. An empirical comparison of bert, roberta, and electra for fact verification. In *2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pages 241–246.

Melania Nitu and Mihai Dascalu. 2024. Authorship attribution in less-resourced languages: A hybrid transformer approach for romanian. *Applied Sciences*, 14(7).

Praboda Rajapaksha, Reza Farahbakhsh, and Noel Crespi. 2021. Bert, xlnet or roberta: The best transfer learning model to detect clickbaits. *IEEE Access*, 9:154704–154716.

Ehsan Reisi and Hassan Mahboob Farimani. 2020. Authorship attribution in historical and literary texts by a deep learning classifier. *Journal of Applied Intelligent Systems and Information Sciences*, 1(2):118–127.

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*.

Antonio Theophilo, Rafael Padilha, Fernanda A Andaló, and Anderson Rocha. 2022. Explainable artificial intelligence for authorship attribution on social media. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2909–2913. IEEE.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.