The iimasNLP team at IberAuTexTification 2024: Integrating Graph Neural Networks, Multilingual LLMs, and Stylometry for Automatic Text Identification

Andric Valdez-Valenzuela[1,*,†], Ricardo Loth Zavala-Reyes[1,*,†], Victor Giovanni Morales-Murillo[2,†] and Helena Goméz-Adorno[3,†]

[1]Posgrado en Ciencia e Ingeniería de la Computación, Universidad Nacional Autónoma de México, CDMX, México.
[2]Benemérita Universidad Autónoma de Puebla, Facultad de Ciencias de la Computación, Puebla, México
[3]Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, Universidad Nacional Autónoma de México, CDMX, México.

Abstract

The emergence of large language models (LLMs) opens up many opportunities, such as text generation, language translation, and human-like question-answering. While these advances are impressive, there is concern that LLMs could also be used for malicious purposes, such as generating fake or misleading content. For this reason, it is urgent to build systems that help distinguish between text written by humans and text generated by LLMs. In this work, as a part of the Autextification 2024 shared task, We proposed a novel architecture to accurately classify text as human-written or machine-generated (Subtask 1) and distinguish between various machine-generated texts (Subtask 2). The system architecture incorporates Graph Neural Networks, Multilingual Large Language Models, and stylometric features to improve the accuracy and robustness of text classification. Our system performed better than the baselines and obtained competitive results on the leaderboard (4th place for Subtask 1 and 2nd place for Subtask 2).

Keywords

AI-Generated Text, Large Language Models, Graph Neural Networks, Stylometric Features, Natural Language Processing,

1. Introduction

This paper describes the iimasNLP team's participation in the Automated Text Identification on Languages of the Iberian Peninsula (IberAuTexTification 2024) shared tasks [1] at the 6th Workshop on Iberian Languages Evaluation Forum (IberLEF 2024) [2] during the 40th International Conference of the Spanish Society for Natural Language Processing (SEPLN 2024). Nowadays, Large Language Models (LLMs) have a high capability to generate human-like texts, which have been integrated into individual and company workflows for many different tasks. Besides, LLMs have demonstrated high performance in several natural language processing (NLP) tasks, such as machine translation, summarization, dialogue systems, question answering, and information retrieval. However, ensuring the authenticity of machine-generated text is a complex challenge due to the potential for these technologies to be used for malicious purposes [3] such as generating academic essays, polarized opinions, fake news, phishing campaigns, malicious code, fake customer profiles, and other text for criminal activities.

On the other hand, the wide availability of LLMs due to the continuous development of these architectures increases the number of potential malicious users. Furthermore, malicious users can generate attacks in different languages, domains, models, or strategies. This shared task aims to create new mechanisms based on NLP, such as content moderation strategies, to deal with machine-generated

IberLEF 2024, September 2024, Valladolid, Spain
*Corresponding author.
†These authors contributed equally.
andric.valdez@gmail.com (A. Valdez-Valenzuela); zricardoloth@gmail.com (R. L. Zavala-Reyes); vg055@hotmail.com (V. G. Morales-Murillo); helena.gomez@iimas.unam.mx (H. Goméz-Adorno)
https://helenagomez-adorno.github.io/ (H. Goméz-Adorno)
0000-0002-0877-7063 (A. Valdez-Valenzuela); 0000-0002-0877-7063 (R. L. Zavala-Reyes); 0000-0002-6786-9232 (V. G. Morales-Murillo); 0000-0002-6966-9912 (H. Goméz-Adorno)

text [1]. AI organizations such as companies, research groups, and academic institutions are greatly interested in addressing these tasks [4] [5]. For example, companies are highly interested in detecting automatically generated content to protect or enhance the reputation of their products and brands and verify the authenticity of news or statements.

For this reason, IberAuTexTification 2024 introduces two sub-tasks. The first sub-task is a binary classification task with two classes, human and generated, where the goal is to detect whether a text has been generated by an LLM, i.e., given a text, the participants should determine whether the text has been automatically generated. The second sub-task is a multi-class classification task, which aims to identify the model that generated a machine-generated text for further forensic purposes, i.e., given an automatically generated text, the participants should determine what model generated it. In addition, five domains were used for training in both sub-tasks, and two domains were utilized by testing to encourage models to learn features that generalize to new writing styles.

The iimasNLP team tackles both sub-tasks of IberAuTexTification 2024 using two main approaches: (1) Graph Neural Networks (GNNs) and (2) Multilingual Large Language Models. The first approach aims to obtain graph-based representations and combine them with stylometric features to generate embeddings for training different classifiers such as stochastic gradient descent (SGD) and support vector classifier (SVC). For the second approach, LLM embeddings are concatenated with the first approach, and the same classifiers are used.

This document is organized as follows: section 2 introduces the background related to this shared task, section 3 describes the system overview describing the architecture proposed and the data stratification, section 4 analyzes the results obtained in our experiments and the final submission, and section 5 presents our conclusions.

2. Background

Different fields, such as computer vision, natural language processing, and speech recognition have been revolutionized by deep learning models and have succeeded in many applications. However, a single deep learning model may have limitations regarding generalization, robustness, and performance [6]. Similarly, traditional machine learning methods may fail to perform satisfactorily when dealing with complex data, such as imbalanced, high-dimensional, or noisy data. Capturing the data's multiple characteristics and underlying structures is a challenge for these methods [7]. Therefore, word representation as embeddings in a continuous vector space is used in multiple NLP tasks, such as text classification, where pre-trained embeddings serve as powerful word representations. Furthermore, better word representations can be obtained by concatenating different types of embeddings, although the proper selection of embedding types for specific tasks remains a challenge [8].

Some traditional Machine Learning methods are used for text classification tasks such as Support Vector Machine (SVM), K-Nearest Neighbor (KNN), and Random Forest [9] [10]. Some Deep Learning methods used are Convolutional Neural Networks (CNN) or Recurrent Neural Networks (RNN), CNN focuses on extracting local feature information of text compared to RNN, which focuses on extracting global feature information of text, which has the risk of gradient disappearance

On the other hand, text graphs have emerged as an innovative solution to the limitations of traditional methods in addressing various NLP tasks, such as text classification. These graph-based techniques emphasize representing text documents as graphs to effectively model the relationships and structure within the data. Utilizing text graph structures in NLP can lead to improved performance and more accurate results [11]. These graph representations are valuable for numerous text operations, including topological, relational, and numerical analyses. For instance, they can be used to extract centrality measures (such as paths, distances, degrees, and clustering) to determine the relative importance of a text node within the network. Another application is employing graph representations to solve text classification tasks using Graph Neural Networks (GNNs). GNNs, which are deep learning-based methods designed to operate on graph-structured data, use text graphs as inputs, leveraging the set of nodes and their relationships (edges) to learn relevant information patterns from these complex

structures [12]. GNNs are particularly effective due to their capacity to directly process graph-structured data, enabling them to capture intricate relationships and dependencies naturally represented as graphs. This capability allows GNNs to learn representations (embeddings) for nodes and edges, capturing both local and global structural information [13].

Finally, Multilingual LLMs leverage powerful language models to handle and respond to queries in multiple languages, achieving remarkable success in multilingual natural language processing tasks [14]. Moreover, Stylometric features are statistical-based text representations, including sentence length, complexity, frequent words, spelling errors, etc. These features have been used to detect writing styles in authorship analysis [15], and related works have utilized these features to identify machine-generated text [16][17].

For this reason, this work combines Graph embeddings, Finetuned Multilingual LLMs embeddings, and Stylometric features to address the tasks of classifying human-written versus machine-generated text (Subtask 1) and distinguishing between different machine-generated texts (Subtask 2).

3. System Overview

This section describes the system overview of our approach: Model Architecture, Data Stratification, and Graph Representation. Model Architecture lays out the structure and detailed workings that drive our method. Data Stratification shows the partition process for the data. Graph Representation explains the text-to-graph representation process.

3.1. Data Stratification

The corpus provided for this second version of the AuTexTification shared task brings a variety of innovations, including more LLMs, new domains, and languages. Therefore, appropriately partitioning the training and validation sets is essential to prevent them from overfitting these characteristics. One effective way to achieve this is by performing a random shuffle before dividing the corpus and then proceeding with the stratified division according to the classes of each subtask, getting well-balanced partitions: 70 % in the training set and 30 % in the validation set. This random shuffle results in a training and test set that is not stratified in terms of the model used to generate the machine-written texts, the domain, or the language, but only in terms of the classes of our tasks.

Table 1 shows each subtask's total number of instances and distribution classes. For Subtask 1, the training comprises 35,636 human-written instances and 41,128 machine-generated instances, leading to 76,764 text documents. For the validation set, there are 15,273 human-written examples and 17,626 machine-generated examples, summing up 32,899 instances. Subtask 2, on the other hand, is divided into six distinct classes labeled A through F. Partitions are partially well-balanced throughout all classes. For the training set, class A has 5,181 examples (the lowest), and class D has 8,679 examples (the highest), making a total number of 41,127 instances; and, for the validation set, the total number of instances sums up to 17,627, having 2,221 examples in class A and 3,720 examples in class D.

Table 1
Total number of problems for Train and Validation sets for Subtask 1 and 2

	Subtask 1			Subtask 2						
Partition	Human	Machine	Total	A	B	C	D	E	F	Total
Train	35,636	41,128	76,764	5,181	5,828	6,255	8,679	6,692	8,492	41,127
Validation	15,273	17,626	32,899	2,221	2,498	2,680	3,720	2,868	3,640	17,627

3.2. Model Architecture

Figure 1 shows a comprehensive architecture designed to address the tasks of classifying human-written versus machine-generated text (Subtask 1) and distinguishing between different machine-generated texts (Subtask 2). The architecture integrates two main approaches: GNNs and Multilingual LLMs.

- **Graph Neural Networks**. The process begins by inputting the train set documents (described above). These documents are transformed into a co-occurrence graph using the text2graphAPI [18]. In this graph, nodes represent words, and edges represent co-occurrence relationships between these words. This transformation captures the relational structure of the text, which is crucial for further processing. The features of the nodes in the co-occurrence graph are initialized using word embeddings extracted from the finetuned BERT Base Multilingual LLM. These embeddings capture the semantic meaning of words, providing a rich feature set for each node in the graph. These graphs are then processed by a Graph Neural Network with TransformerConv Layers [19]. This network processes the graph to generate document embeddings encapsulating the text's structural and relational information. The output of this GNN is a set of document embeddings, referred to as Graph Docs Embeddings, which provide a graph-based representation of the text documents. Finally, these embeddings (in combination with stylometric features) are used to train a final classifier using traditional machine learning algorithms such as Stochastic Gradient Descent (SGD) or Support Vector Classifier (SVC).
- **Multilingual LLMs** In this approach, three LLMs are employed to extract deep semantic embeddings from the text documents. The models used are BERT-Base-Multilingual [20], Multilingual-E5-Large [21], and XLM-Roberta-Base [22]. Each model is fine-tuned to generate document-level embeddings, capturing context-aware features from the text. Additionally, stylometric features are extracted to capture each document's linguistic and stylistic properties. The embeddings extracted from the LLMs and the stylometric features are concatenated to form a contextualized representation for each document. Finally, these embeddings are fed into a machine learning classifier (such as SGD and SVC) to perform the final classification task for both subtasks.

On the other hand, although LLMs can capture contextual information from a document and greatly contribute to model performance, different fields can be explored to characterize a given text. One of these is Stylometry, which analyzes the linguistic style of the text and is frequently used in the Authorship Analysis area. This was the motivation behind incorporating stylometric features into our architecture, as our primary goal was to find a representation of documents that encapsulates condensed context information and a characterization of their writing style and composition for each one.

The incorporated features are valid for any language and are defined as follows [23]:

- Lexical diversity: Provides an idea of the author's vocabulary richness. A higher ratio indicates a more varied vocabulary and reflects tendencies toward word repetition.
- Average word length: Indicates the average number of characters per word in the text. The use of longer words is generally associated with more pedantic and formal writing styles, whereas shorter words are typical in informal spoken language.
- Average sentence length: This represents the average number of words per sentence in the text. Longer sentences often indicate carefully planned writing, while shorter sentences are more characteristic of spoken language.
- Standard deviation of sentence length: Indicates variation in sentence length.
- Average paragraph length: Average number of sentences per paragraph in the text. It takes into account that paragraph length is influenced by dialogue presence.
- Chapter length: This measure provides insights into how the author structures and organizes content. Longer chapters may indicate a more detailed exploration of themes, while shorter chapters may suggest a more concise style.
- Number of commas per thousand tokens: Provides information on the continuous flow of ideas within a sentence.

- Number of semicolons per thousand tokens: Indicates the author's tendency to use semicolons to connect ideas within a sentence rather than ending it and starting anew.
- Number of quotation marks per thousand tokens: Determines the frequency of quotations. Frequent use of quotations is considered a typical feature of engagement.
- Number of exclamation marks per thousand tokens: Displays the frequency of expressing intense emotions or emphasis in the analyzed text.
- Number of colons per thousand tokens: Helps structure and organize ideas, providing clarity and emphasis on relationships between different parts of the text.
- Number of dashes per thousand tokens: Some authors use hyphenated words more than others.
- Number of long dashes per thousand tokens: Used to separate clauses or phrases within a sentence, add additional information or emphasize certain elements.
- Number of digits: The number of digits can influence the author's style perception, especially in academic, scientific, or technical texts where numerical data is relevant to the content.
- Number of spaces: Contribute to text readability and organization.

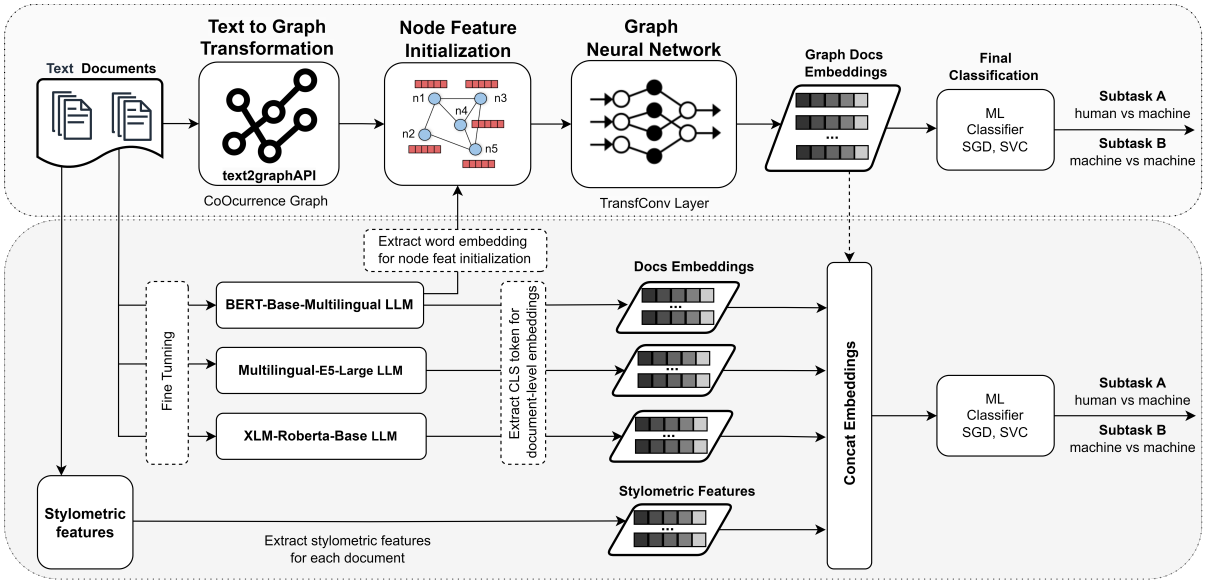

Figure 1: Model Architecture.

4. Results

This section presents the results obtained by running several experiments (evaluating the validation set) using the architecture described above and shows the final submission scores released by the organizers.

Table 2 shows the results using different approaches for Subtask1 and Subtask2. It compares various combinations of GNNs, LLMs, and stylometric features. The best performance for Subtask 1 is achieved using the combination of LLMs, GNNs, and Stylometry features, with a Macro F1 score of 0.9746. For Subtask 2, the best performance is achieved using LLMs + GNNs, with a Macro F1 score of 0.8828.

Based on these results, We decided to submit three runs per subtask to cover the main approaches described before; the final runs and approaches were named as follows:

- Run 1 -> GNNs + LLMs + StylometryFeat
- Run 2 -> GNNs + StylometryFeat
- Run 3 -> LLMs + StylometryFeat

Table 2
Results in the validation set using different approaches for Subtask1 and Subtask2

Approach	Subtask1		Subtask2	
	Clf Model	Marco F1	Clf Model	Marco F1
GNNs	SGD	0.9352	SVC	0.7614
GNNs + StylometryFeat	SGD	0.9358	SVC	0.7707
LLMs	SGD	0.9743	SVC	0.8814
LLMs + StylometryFeat	SGD	0.9743	SVC	0.8814
LLMs + GNNs	SGD	0.9745	SVC	**0.8828**
LLMs + GNNs + StylometryFeat	SGD	**0.9746**	SVC	0.8813

Table 3 shows the results for the final submission across all systems. Our team called **iimasNLP** performed notably well in both subtasks, securing top positions and high Macro-F1 scores. In Subtask 1, our system's best performance was in Run 2, achieving a Macro-F1 score of 0.7188, getting the 4th position out of 54 submissions. In Subtask 2, our best performance was in Run 3, with a Macro-F1 score of 0.5173, obtaining the 2nd position out of 14 submissions.

Table 3
Final submission leaderboard (test set) for Subtask1 and Subtask2; our team is called iimasNLP.

Subtask1				Subtask2			
Position	Team	Run	Macro-F1	Position	Team	Run	Macro-F1
1	jor_isa_uc3m	1	0.8050	1	gmc_fosunlp	1	0.5231
2	gmc_fosunlp	1	0.7663	**2**	**iimasNLP**	**3**	**0.5173**
3	telescope_team	2	0.7579	3	Drocks	2	0.5075
4	**iimasNLP**	**2**	**0.7188**	4	Drocks	1	0.5030
5	gmc_fosunlp	2	0.7155	**5**	**iimasNLP**	**1**	**0.4958**
8	**iimasNLP**	**3**	**0.7051**	6	KaramiTeam	2	0.4930
9	telescope_team	1	0.6965	7	Drocks	3	0.4827
10	**iimasNLP**	**1**	**0.6793**	8	KaramiTeam	1	0.4806
22	paporomerol	2	0.6418	9	Aberdeen	3	0.4006
28	KaramiTeam	2	0.6315	10	Aberdeen	1	0.4002
34	Joavpa	2	0.6083	11	Achraf	1	0.3905
52	olgasolana	2	0.5684	12	Yano	1	0.3043
53	paporomerol	1	0.5629	13	Aberdeen	2	0.3034
54	Yano	3	0.5608	14	**iimasNLP**	**2**	**0.1582**

5. Conclusions

This research presents a novel system developed for the AuTexTification 2024 shared task that demonstrates a robust integration of Graph Neural Networks, Multilingual Large Language Models, and stylometric features to classify human-written versus machine-generated texts and differentiate among various types of machine-generated texts. The data stratification approach ensured a balanced distribution of training and validation datasets, which is crucial for preventing model overfitting and enhancing generalization across new domains and languages.

On the other hand, our results show robust performance and great potential; we obtained better results compared to the baselines and competitive performance compared to the first places (especially for subtask 2). Moreover, for future work, it could be interesting to implement refinements to the architecture proposed: try different graph representations such as Heterogeneous Graphs, use LLMs that extract syntactic and semantic information from the text, and use more advanced stylometric features.

6. Acknowledgments

This paper has been supported by PAPIIT projects IN104424, TA101722, and CONAHCYT CF-2023-G-64. We also want to thank Ricardo Villareal and Rita Rodríguez for their help with the computing resources and Eng. Roman Osorio for assisting with the project's student administration.

References

[1] A. M. Sarvazyan, J. Á. González, F. Rangel, P. Rosso, M. Franco-Salvador, Overview of iberautextification at iberlef 2024: Detection and attribution of machine-generated text on languages of the iberian peninsula, Procesamiento del Lenguaje Natural 73 (2024).

[2] L. Chiruzzo, S. M. Jiménez-Zafra, F. Rangel, Overview of IberLEF 2024: Natural Language Processing Challenges for Spanish and other Iberian Languages, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2024), co-located with the 40th Conference of the Spanish Society for Natural Language Processing (SEPLN 2024), CEUR-WS.org, 2024.

[3] M. Nitu, M. Dascalu, Beyond lexical boundaries: Llm-generated text detection for romanian digital libraries, Future Internet 16 (2024). URL: https://www.mdpi.com/1999-5903/16/2/41. doi:10.3390/fi16020041.

[4] R. Deng, F. Duzhin, Topological data analysis helps to improve accuracy of deep learning models for fake news detection trained on very small training sets, Big Data and Cognitive Computing 6 (2022). URL: https://www.mdpi.com/2504-2289/6/3/74. doi:10.3390/bdcc6030074.

[5] J. Tourille, B. Sow, A. Popescu, Automatic Detection of Bot-generated Tweets, in: 1st ACM International Workshop on Multimedia AI against Disinformation, Proceedings of the 1st International Workshop on Multimedia AI against Disinformation (MAD '22), Newark, United States, 2022, pp. 44–51. URL: https://cea.hal.science/cea-03788573. doi:10.1145/3512732.3533584.

[6] S. Abimannan, E.-S. M. El-Alfy, Y.-S. Chang, S. Hussain, S. Shukla, D. Satheesh, Ensemble multifeatured deep learning models and applications: A survey, IEEE Access 11 (2023) 107194–107217. doi:10.1109/ACCESS.2023.3320042.

[7] X. Dong, Z. Yu, W. Cao, Y. Shi, Q. Ma, A survey on ensemble learning, FRONTIERS OF COMPUTER SCIENCE 14 (2020) 241–258. doi:10.1007/s11704-019-8208-z.

[8] X. Wang, Y. Jiang, N. Bach, T. Wang, Z. Huang, F. Huang, K. Tu, Automated concatenation of embeddings for structured prediction, 2021. arXiv:2010.05006.

[9] S. Merugu, M. C. S. Reddy, E. Goyal, L. Piplani, Text message classification using supervised machine learning algorithms, in: A. Kumar, S. Mozar (Eds.), ICCCE 2018, volume 500 of *Lecture Notes in Electrical Engineering*, 2019, pp. 141–150. doi:10.1007/978-981-13-0212-1_15, international Conference on Communications and Cyber Physical Engineering (ICCCE), Hyderabad, INDIA, 2018.

[10] E. D. Madyatmadja, C. P. M. Sianipar, C. Wijaya, D. J. M. Sembiring, Classifying crowdsourced citizen complaints through data mining: Accuracy testing of k-nearest neighbors, random forest, support vector machine, and adaboost, INFORMATICS-BASEL 10 (2023). doi:10.3390/informatics10040084.

[11] A. H. Osman, O. M. Barukub, Graph-based text representation and matching: A review of the state of the art and future challenges, IEEE Access 8 (2020) 87562–87583. doi:10.1109/ACCESS.2020.2993191.

[12] L. Wu, Y. Chen, K. Shen, X. Guo, H. Gao, S. Li, J. Pei, B. Long, et al., Graph neural networks for natural language processing: A survey, Foundations and Trends® in Machine Learning 16 (2023) 119–328.

[13] B. Jin, G. Liu, C. Han, M. Jiang, H. Ji, J. Han, Large language models on graphs: A comprehensive survey, arXiv preprint arXiv:2312.02783 (2023).

[14] L. Qin, Q. Chen, Y. Zhou, Z. Chen, Y. Li, L. Liao, M. Li, W. Che, P. S. Yu, Multilingual large language model: A survey of resources, taxonomy and frontiers, 2024. arXiv:2404.04925.

[15] H. Gomez Adorno, G. Rios, J. Posadas Durán, G. Sidorov, G. Sierra, Stylometry-based approach for detecting writing style changes in literary texts, Computación y Sistemas 22 (2018). doi:10.13053/cys-22-1-2882.
[16] T. Kumarage, J. Garland, A. Bhattacharjee, K. Trapeznikov, S. Ruston, H. Liu, Stylometric detection of ai-generated text in twitter timelines, 2023. arXiv:2303.03697.
[17] G. K. Mikros, A. Koursaris, D. Bilianos, G. Markopoulos, Ai-writing detection using an ensemble of transformers and stylometric features, in: IberLEF@SEPLN, 2023. URL: https://api.semanticscholar.org/CorpusID:264586529.
[18] A. Valdez, H. Gómez Adorno, Text2graphapi a library to transform text documents into different graph representations, Available at SSRN 4763799 (????).
[19] Y. Shi, Z. Huang, S. Feng, H. Zhong, W. Wang, Y. Sun, Masked label prediction: Unified message passing model for semi-supervised classification, 2021. arXiv:2009.03509.
[20] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).
[21] L. Wang, N. Yang, X. Huang, L. Yang, R. Majumder, F. Wei, Multilingual e5 text embeddings: A technical report, arXiv preprint arXiv:2402.05672 (2024).
[22] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, arXiv preprint arXiv:1911.02116 (2019).
[23] GitHub - jpotts18/stylometry: A Stylometry Library for Python — github.com, https://github.com/jpotts18/stylometry, ???? [Accessed 12-07-2024].