Multidimensional Text Feature Analysis: Unveiling the Veil of Automatically Generated Text

Mingcan Guo[1], Zhongyuan Han[1,*], Xintian Wang[2] and Jiangao Peng[1]

[1]*Foshan University, Foshan, China*
[2]*Minzu University of China, Beijing, China*

Abstract

With the rapid development of artificial intelligence technology, the application of large language models (LLMs) in generating text has become increasingly widespread across various domains. However, distinguishing automatically generated text from human-authored text and determining the attribution of generated text to specific LLMs remain challenging problems. Our proposed method is developed to address these challenges by participating in the shared task "IberAuTexTification: Automated Text Identification on Languages of the Iberian Peninsula." Our method primarily involves incorporating different text features, including word-level perplexity features and sentence-level features, such as count, readability, lexical richness, and punctuation, into the pre-trained language model(PLM). These features improve the performance of our text detector and model attributor in identifying automatically generated text. Finally, our model achieved a Macro-F1 score of 0.7663 in subtask 1 and 0.5231 in subtask 2 on the official test set.

Keywords

LLMs, automatically generated text, text features, text detector, model attributor

1. Introduction

As artificial intelligence (AI) continues to advance, the proliferation of automatically generated text presents remarkable opportunities and significant challenges. With the advent of sophisticated AI language models like GPT-4 and its successors, the creation of high-quality text has become increasingly accessible. However, this technological advancement has led to a pressing need for robust methods to distinguish between human-authored and automatically generated content. Furthermore, the ability to attribute specific texts to their respective LLM sources has emerged as a critical area of research, given the diverse applications and potential misuse of these models.

To address the challenges of automatically generated text, IberLEF 2024 [1] has launched the IberAuTexTification: Automated Text Identification on Languages of the Iberian Peninsula shared task [2], which is an upgraded version of the 2023 AuTexTification task [3]. This task expands to include more models, domains, and languages from the Iberian Peninsula, aiming to develop a more versatile text detection and attribution system.

Participants are required to develop models that utilize linguistic and semantic cues to identify automatically generated texts from different models, domains, and languages. Subtask 1 requires participants to determine whether a given text is automatically generated, which is a binary classification task. On the other hand, subtask 2 requires participants to identify which LLMs generated a given text, making it a multi-class classification task.

This paper employs a comprehensive set of text features to explore innovative method for text detection and attribution. We combine perplexity features extracted from the modeling process with traditional sentence-level features. We then fine-tune PLM and incorporate features to generate the final results. Our method aims to identify automatically generated text and determine its source accurately. This Multidimensional feature analysis not only enhances detection accuracy but also provides deeper

IberLEF 2024, September 2024, Valladolid, Spain

*Corresponding author.

gmc9812@163.com (M. Guo); hanzhongyuan@gmail.com (Z. Han); wxintian2022@126.com (X. Wang); wyd1n910@gmail.com (J. Peng)
0000-0002-4977-2138 (M. Guo); 0000-0001-8960-9872 (Z. Han); 0009-0006-3780-5023 (J. Peng)

insights into the unique characteristics of different LLMs. It contributes to the responsible and ethical use of AI in text generation.

2. Related Work

The analysis of automatically generated content focuses on detecting the difference between human and machine-generated text and attributing the different generated text to specific models.

For generation detection, Tang et al.[4] proposed that the technical routes for automatically generated text detection methods are Black-box Detection and White-box Detection. White-box Detection requires the developer to be an insider of a LLM and then insert a watermark for detection during or after the generation of the text[5]. Black-box detection is mainly based on the text features of automatically generated text and human-generated text, and it mainly utilizes feature-based methods and neural network-based methods. The former has been developed more maturely but is limited by its reliance on large data to extract features[6]. At the same time, the latter can obtain good results on small sample datasets [7] by fine-tuning PLMs such as Bert[8] and RoBERTa[9].

For model attributing, Wu et al.[10] introduce LLMDet, a text detection tool that accurately identifies the sources of generated text, such as Human, LLaMA, OPT, or others. LLMDet calculates the proxy perplexity of each language model by analyzing the probabilities of significant n-grams' following words, enabling precise attribution of the generated text to specific language models.

Using PLM to incorporate different text features has been empirically proven effective[11, 12]. Some researchers have noted that incorporating features such as Perplexity [13], Semantic [14], and stylometric features [15] can yield text encodings with high-quality semantic information. These findings have provided us with inspiration for our work.

3. Method

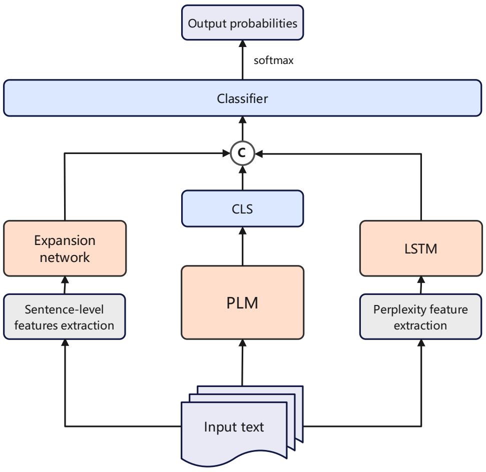

Figure 1: Our model architecture for text detection and attribution

In this section, we will describe the design and implementation of our solution for the task of automatically generated text. We aim to identify whether a given text is automatically generated (subtask 1) or determine its source LLM using a combination of multiple techniques (subtask 2). We primarily utilize the PLM and incorporate multi-dimensional textual feature information to achieve text detection and model attribution.

Due to the similar optimization objectives of subtask 1 and subtask 2, we only need to determine whether the task belongs to binary or multi-class classification during implementation. Therefore, we use a consistent model framework for both the text detector and model attributor and the target class quantity can be specified in the classifier for different subtasks. Figure 1 illustrates our model architecture, which consists of three main modules: extraction of sentence-level features, extraction of perplexity features, and PLM. The modules in our model are designed to be functionally independent, but they are integrated into a single framework and jointly trained in an end-to-end fashion. During training, they are optimized together to achieve the overall objective of the model.

3.1. Perplexity Feature

When evaluating the prediction accuracy of a system on sample data, we often refer to the perplexity metric, which quantifies the ambiguity or unpredictability of data interpretation. According to the research[16], LLMs tend to exhibit lower perplexity scores when generating text due to their overemphasis on regular patterns in the training data. In stark contrast, human authors possess infinite flexibility and diversity in expression, which poses a challenge for language models regarding prediction. As a result, texts created by human authors often exhibit higher perplexity scores. Correspondingly, texts generated by different LLMs also tend to differ in perplexity performance.

The perplexity can be determined by the information entropy, where for a given token sequence of a specific length, the information entropy $H(t)$ is defined as Equation 1.

$$H(t) = -\sum_{t \in \mathcal{T}} p(t) \log_2 p(t) \tag{1}$$

In Equation 1, $\mathcal{T}$ represents the sample space of token t, and p represents the corresponding probability distribution of t. For a discrete random variable t with a probability distribution p, the perplexity can be further expressed as Equation 2.

$$PPL(t) = 2^{H(t)} \tag{2}$$

Although the task involves generating text using various LLMs, they are all based on the same structure, namely, the transformer. Therefore, we select the latest and representative Llama3-8b [17] as the scoring model to generate the probability distribution of text tokens.

In addition, for a given sequence where t_i represents the i-th token in the sequence and t_{i+1} represents the next token immediately following t_i, t_i' represents the Llama3-8b predicted token for t_i (the token with the highest probability in the vocabulary). Therefore, when considering the token probabilities across the entire vocabulary in Llama3-8b, we also take into account the logarithmic probabilities of the occurrence of the succeeding token t_{i+1} and the predicted token t_i in the vocabulary. As shown in Equation 3 and Equation 4, these measures assess the probability of the context token occurring and the model's confidence in predicting the token, respectively.

$$Prob(t_{i+1}) = \log p(t_{i+1}) \tag{3}$$

$$Conf(t_i') = \log p(t_i') \tag{4}$$

Finally, the three feature values are concatenated together and fed into an LSTM network for encoding processing, resulting in a 128-dimensional vector representation as the output.

3.2. Sentence-level Feature

The research conducted by Mindner et al. [18] indicates that text generated by LLMs exhibits noticeable differences from human-authored text regarding traditional stylometric features. Therefore, besides the perplexity feature, we consider incorporating the following four categories of stylometric features, consisting of 32 indicator dimensions, to enhance semantic representation in our model.

- **Count:** This category includes four indicators: word count, sentence count, average number of words per sentence, and the square root of the variance of the number of words per sentence.

- **Readability:** This category uses the Python readability library to assess the text's readability. It includes three indicators: flesch-kincaid grade level, flesch reading ease score, and automated readability Index.
- **Lexical richness:** This category measures the vocabulary richness of the text using the Python lexical richness library. It includes two indicators: mean average type-token ratio (MATTR) and textual lexical diversity (MTLD) measure.
- **Punctuation:** This category calculates the frequency of punctuation marks within the set of symbols ("!", "’", ",", "-", ":", ";", "?", "@", """, "=", "#"). It comprises 23 indicators, including the total count of punctuation marks, the percentage of each punctuation mark in the document, and the average number of punctuation marks per sentence.

Based on the abovementioned features, we have designed an Expansion Network, a sequence container consisting of two linear layers. Its purpose is to elevate the dimensionality of sentence-level features to 128 dimensions.

3.3. Fine-tuning Model and Feature Fusion

Given the widespread use of transformer-based pre-trained models in various downstream tasks and the availability of multilingual datasets in official task data, XLM-RoBERTa[19] is a multilingual version of Roberta that has been pre-trained on a filtered CommonCrawl dataset comprising 2.5TB of data from 100 languages. It has demonstrated high effectiveness in multilingual classification tasks. XLM-Align[20] is a pre-trained cross-lingual language model that supports 94 languages. We attempted to use XLM-RoBERTa-Large and XLM-Align-Base as PLM in our frameworks.

Next, we will use PLM to obtain the vector representation $[CLS]$ and concatenate the perplexity vector representation from the LSTM output, the sentence-level feature vector representation from the expansion network output, and $[CLS]$. This concatenated representation will be fed into a classifier of two linear layers. Finally, we fine-tune the entire model network, and The output will then pass through a softmax function to obtain the predicted probabilities for each class.

4. Experiments and Results

4.1. Datasets

The datasets provided by the IberAuTexTification shared task at IberLEF 2024 are designed with a focus on multilingualism (languages from the Iberian Peninsula such as Spanish, English, Catalan, Galician, Basque, and Portuguese), multidomain (news, comments, emails, essays, dialogues, Wikipedia, wikiHow, tweets, emails, etc.), and multimodal (GPT[21], LLaMA[22], Mistral[23], Anthropic[24], MPT[25], Falcon[26], etc.) settings.

For the training set, subtask 1 provides $109,663$ texts, while subtask 2 provides $58,754$ texts. These ample amounts of data are sufficient for fine-tuning a model with solid feature extraction. Regarding the test set, subtask 1 provides $43,365$ texts, while subtask 2 provides $23,935$ texts.

For each subtask, to encourage models to learn features that generalize to new writing styles, five domains will be used for training and two different domains for testing.

4.2. Experimental Setting

We split data on the officially labeled training set into a 70% training set and a 30% test set for training and evaluation purposes. Furthermore, within the training set, we further split it into a new training set and a validation set using an $8:2$ ratio for model fine-tuning and validation.

For the framework and parameter selection, we utilized the pytorch-lightning framework to develop our model. We set the batch size to 48 and the maximum sequence length to 512. The initial learning rate was set to 2e-5, and during the training process, we employed the ReduceLROnPlateau algorithm[27] to adjust the learning rate dynamically. Finally, we fine-tune model on a machine with A800 using the AdamW[28] optimizer for an average of 25 epochs.

In addition, we set hyperparameters is_feature and num_classes to specify whether to include text features and to indicate the number of target classes, respectively. This allows us to generate different runs for different subtasks. We freeze the model's weights during inference to output the final results on the official test set. For subtask 1, we submitted a run incorporating text features and another run that solely fine-tunes the PLM. For subtask 2, we submitted a run that incorporates text features.

Table 1
The internal evaluation results of our model during the verification phase using the Macro-F1 score. The best values in each column are highlighted in bold.

Variant	Subtask 1	Subtask 2
XLM-Align-Base	0.9697	0.8428
XLM-Align-Base+Feature	0.9701	0.8453
XLM-RoBERTa-Large	0.9716	0.8586
XLM-RoBERTa-Large+Feature	**0.9780**	**0.8837**

Table 2
Top 10 submissions in the final evaluation and calculated Macro-F1 scores

	Subtask 1		Subtask 2	
No	Name	Macro-F1	Name	Macro-F1
1	jor_isa_uc3m-run1	0.8050	**PLM+Feature-run1**	0.5231
2	**PLM+Feature-run1**	0.7663	iimasNLP-run3	0.5173
3	telescope_team-run2	0.7579	Drocks-run2	0.5075
4	iimasNLP-run2	0.7188	Drocks-run1	0.5030
5	**PLM-run2**	0.7155	iimasNLP-run1	0.4958
6	telescope_team-run3	0.7118	KaramiTeam-run2	0.4930
7	jor_isa_uc3m-run2	0.7069	Drocks-run3	0.4827
8	iimasNLP-run3	0.7051	KaramiTeam-run1	0.4806
9	llmixtic_llama-baseline	0.6984	mdeberta-v3-base-baseline	0.4650
10	telescope_team-run1	0.6965	llmixtic_llama-baseline	0.4555
	(55 more)		(15 more)	

4.3. Results

Table 1 presents the internal validation results of our experimental process, evaluating different variations of pre-training models with and without text features. Macro-F1 is used as the evaluation metric to quantify the performance changes resulting from feature fusion. The results indicate that adding text features enhances the models' classification and generalization abilities, improving performance in both subtasks to varying degrees.

Table 2 presents the final evaluation results obtained from different configurations we submitted for subtask 1 and subtask 2, with the Macro-F1 metric used as the evaluation measure, here we use XLM-RoBERTa-Large as PLM.

For subtask 1, our best performance ranked 2nd out of 65 runs, achieving a Macro-F1 score of 0.7663. In subtask 2, our method achieved the top rank among 25 runs, with a Macro-F1 score of 0.5231. Our method consistently outperformed the majority of runs, including all advanced baselines, validating the effectiveness of our research approach and experimental design. Furthermore, the results in subtask 1 indicate that runs incorporating text features with the PLM outperformed those fine-tuning only the PLM. This also shows the benefit of incorporating text features to improve automatically generated text classification performance.

Nevertheless, the model exhibits a significant disparity between the final evaluation results and the internal validation results. This discrepancy may primarily be attributed to utilizing a test set that encompasses samples from domains with different styles compared to the training set. Regarding generalization capabilities, there remains substantial room for improvement in the model.

5. Conclusion

This paper describes the proposed method for the IberAuTexTification: Automated Text Identification on Languages of the Iberian Peninsula task. We employed fine-tuning a PLM and incorporating multidimensional text features to construct detectors and attributors. Specifically, we introduced a method that combines LLM to measure the perplexity feature of text sequences and incorporates four categories of sentence-level features: Count, Readability, Lexical richness, and Punctuation. These enhancements aim to improve the performance of the PLM. Through evaluations conducted on the task, our proposed method demonstrates good performance.

We aim to contribute to the overall understanding of LLMs' text generation drive advancements in this research field and provide valuable references for future related tasks. Future work should continue to explore the effectiveness of different features in identifying automatically generated text across languages, domains, and LLM variants. Additionally, investigating the fusion method of feature vectors and PLMs can help improve the model's ability to capture useful information. Finally, there is ample room for improvement and progress in text generation detection and model attribution tasks, which require further exploration.

Acknowledgments

This work is supported by the Social Science Foundation of Guangdong Province, China (No.GD24CZY02)

References

[1] 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.

[2] 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).

[3] A. M. Sarvazyan, J. Á. González, M. Franco-Salvador, F. Rangel, B. Chulvi, P. Rosso, Overview of autextification at iberlef 2023: Detection and attribution of machine-generated text in multiple domains, Procesamiento del Lenguaje Natural 71 (2023) 275–288.

[4] R. Tang, Y.-N. Chuang, X. Hu, The science of detecting llm-generated texts, arXiv preprint arXiv:2303.07205 (2023).

[5] J. Kirchenbauer, J. Geiping, Y. Wen, J. Katz, I. Miers, T. Goldstein, A watermark for large language models, in: International Conference on Machine Learning, PMLR, 2023, pp. 17061–17084.

[6] K. Krishna, Y. Song, M. Karpinska, J. Wieting, M. Iyyer, Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense, Advances in Neural Information Processing Systems 36 (2024).

[7] R. Koike, M. Kaneko, N. Okazaki, Outfox: Llm-generated essay detection through in-context learning with adversarially generated examples, in: Proceedings of the AAAI Conference on Artificial Intelligence, volume 38, 2024, pp. 21258–21266.

[8] 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).

[9] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized BERT pretraining approach, CoRR abs/1907.11692 (2019). URL: http://arxiv.org/abs/1907.11692. arXiv:1907.11692.

[10] K. Wu, L. Pang, H. Shen, X. Cheng, T.-S. Chua, Llmdet: A third party large language models generated text detection tool, in: Findings of the Association for Computational Linguistics: EMNLP 2023, 2023, pp. 2113–2133.

[11] A. Palmer, N. Schneider, N. Schluter, G. Emerson, A. Herbelot, X. Zhu, Proceedings of the 15th international workshop on semantic evaluation (semeval-2021), in: Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), 2021.

[12] P. Przybyła, N. Duran-Silva, S. Egea-Gómez, I've seen things you machines wouldn't believe: Measuring content predictability to identify automatically-generated text, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2023). CEUR Workshop Proceedings, CEUR-WS, Jaén, Spain, 2023.

[13] S. Gehrmann, H. Strobelt, A. M. Rush, Gltr: Statistical detection and visualization of generated text, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2019, pp. 111–116.

[14] B. Guo, X. Zhang, Z. Wang, M. Jiang, J. Nie, Y. Ding, J. Yue, Y. Wu, How close is chatgpt to human experts? comparison corpus, evaluation, and detection, differences 14 (2023) 18.

[15] T. Kumarage, J. Garland, A. Bhattacharjee, K. Trapeznikov, S. Ruston, H. Liu, Stylometric detection of ai-generated text in twitter timelines, arXiv preprint arXiv:2303.03697 (2023).

[16] R. Tang, Y.-N. Chuang, X. Hu, The science of detecting llm-generated text, Communications of the ACM 67 (2024) 50–59.

[17] AI@Meta, Llama 3 model card (2024). URL: https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

[18] L. Mindner, T. Schlippe, K. Schaaff, Classification of human-and ai-generated texts: Investigating features for chatgpt, in: International Conference on Artificial Intelligence in Education Technology, Springer, 2023, pp. 152–170.

[19] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, É. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised cross-lingual representation learning at scale, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 8440–8451.

[20] Z. Chi, L. Dong, B. Zheng, S. Huang, X.-L. Mao, H. Huang, F. Wei, Improving pretrained cross-lingual language models via self-labeled word alignment, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Online, 2021, pp. 3418–3430. URL: https://aclanthology.org/2021.acl-long.265. doi:10.18653/v1/2021.acl-long.265.

[21] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Advances in neural information processing systems 33 (2020) 1877–1901.

[22] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al., Llama: Open and efficient foundation language models, arXiv preprint arXiv:2302.13971 (2023).

[23] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, et al., Mistral 7b, arXiv preprint arXiv:2310.06825 (2023).

[24] A. Askell, Y. Bai, A. Chen, D. Drain, D. Ganguli, T. Henighan, A. Jones, N. Joseph, B. Mann, N. DasSarma, et al., A general language assistant as a laboratory for alignment, arXiv preprint arXiv:2112.00861 (2021).

[25] M. N. Team, Introducing mpt-7b: A new standard for open-source, commercially usable llms, 2023. URL: www.mosaicml.com/blog/mpt-7b, accessed: 2023-05-05.

[26] E. Almazrouei, H. Alobeidli, A. Alshamsi, A. Cappelli, R. Cojocaru, M. Debbah, É. Goffinet, D. Hesslow, J. Launay, Q. Malartic, et al., The falcon series of open language models, arXiv preprint

arXiv:2311.16867 (2023).

[27] I. Loshchilov, F. Hutter, Decoupled weight decay regularization, in: International Conference on Learning Representations, 2017.

[28] N. Shazeer, M. Stern, Adafactor: Adaptive learning rates with sublinear memory cost, in: International Conference on Machine Learning, PMLR, 2018, pp. 4596–4604.