

Syntax-based Contrastive Learning for Automated Text Identification

Hongyan Wu¹, Nankai Lin^{2,*} and Shengyi Jiang^{1,*}

¹*School of Information Science and Technology, Guangdong University of Foreign Studies, Guangzhou, Guangdong, PR China*

²*School of Computer Science and Technology, Guangdong University of Technology, Guangzhou, Guangdong, PR China*

Abstract

Distinguishing automated text from human-written text is increasingly challenging due to the amazing potential of Natural Language Generation models. Existing automated text identification models primarily employ feature-based methods and neural network-based methods. However, the feature-based approach is effective in capturing syntactic features yet relies heavily on linguistic knowledge, inducing poor linguistic transferability. Concerning neural network-based methods, they excel in text representation but struggle with capturing syntactic features. Thus, this study aims to enhance the neural network model's syntax representation capability, which is conducive to identifying automated texts. Based on the IberLEF 2023 AuTextification task, we propose a novel syntax-based contrastive learning (SCL) method that explicitly incorporates syntactic features into the neural network model. In the evaluation phase, our SCL method achieved third and fifth place in Subtask 2 for English and Spanish, respectively.

Keywords

Syntax-based contrastive learning, automated text identification, AuTextification

1. Introduction

Automated text is increasingly difficult to distinguish from human-written text. Powerful open-source models are being made freely available. The enormous potential of state-of-the-art Natural Language Generation (NLG) systems is being eroded by multiple avenues of abuse. The analysis of threat models suggests that identification is a crucial strategy to reduce the damage caused by NLG model abuse [1].

Identifying automated text involves two tasks: identifying whether the text is generated by the machine and identifying which model the generative text is generated by. Essentially, the two tasks can be considered as a binary classification task and a multi-classification task. Previous automated text identification models primarily employ feature-based methods and neural network-based methods. Nevertheless, although the feature-based approach can capture the syntactic features of texts, it heavily relies on linguistic knowledge to a certain extent, resulting in poor linguistic transferability. Meanwhile, the neural network-based approach exhibits strong text representation capabilities but is not very adept at capturing syntactic

IberLEF 2023, September 2023, Jaén, Spain


*Corresponding author.

✉ 2754976781@qq.com (H. Wu); neakail@outlook.com (N. Lin); 200511402@oamail.gdufs.edu.cn (S. Jiang)

ORCID 0000-0003-2838-8273 (N. Lin)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

features. Thus, this paper focuses on enhancing the syntax representation capability of the neural network model, equipped with outstanding capabilities to identify automated texts.

Based on the task IberLEF 2023 [2] AuTextification: Automated Text Identification [3], we present an innovative syntax-based contrastive learning (SCL) method, which enables the neural network model to explicitly capture syntactic features. The proposed technique demonstrates impressive performance in identifying whether the text is machine-generated (Subtask 1) and identifying a specific generative model for the automated text (Subtask 2). In the evaluation phase, our SCL method secured third and fifth place on subtask 2 for English and Spanish, respectively.

2. Related Work

Existing automated text identification models mainly exploit feature-based methods and neural network-based methods.

Feature-based methods for identifying machine-generated text have matured and continued to show significant value against contemporary NLG models. While these methods have advantages in providing diverse features that may complicate [4] or improve the efficiency of adversarial attacks [5, 6], they also have weaknesses. Specifically, some feature architectures and sampling methods may lack portability and require more samples to make general statistical trends clear. Previous research has revealed that statistical methods are most effective when applied to larger text sets.

Neural network-based identification methods, especially those that leverage features extracted from the Transformer neural language model (NLM), demonstrate high efficiency in detecting machine-generated text. It is consistent with the trend of natural language processing, where the Transformer model has achieved state-of-the-art performance on various natural language tasks.

NLM-based methods fall into two categories: zero-shot classification based on existing models and fine-tuning based on pre-trained language models, which represent the vast majority of NLM-based automated text identification. The baseline methods for identifying automated text involve leveraging the generative models to perform text classification, for instance, GPT-2 or Grover [7, 8, 9]. The generative models can be utilized without fine-tuning to identify their output or output from similar generative models. Autoregressive generative models, such as GPT-2, GPT-3 and Grover, are unidirectional with each token having an embedding that depends on the past tokens' embeddings, which generates the words of the input sentence one by one and exploits them as input for the next word to obtain a feature vector of the sentence. The feature vectors derived from the labeled datasets containing automated texts as well as human-written texts can be exploited to train the linear layer of neurons, determining whether the input sequence is generated by machines or humans.

The most advanced methods for identifying automated text using neural networks involve fine-tuning large bidirectional language models. In terms of the original GPT-2 text evaluation methods, RoBERTa [10], a BERT-based masked language model, is fine-tuned to distinguish whether a given sentence has been generated by a machine or a human. More precisely, appending a classification token [CLS] at the beginning of the input sequence generates an

embedding of the input sequence, and the embedding of the token [CLS] is utilized as a feature vector of the entire sentence to perform classification. Kushnareva et al. [11] adopted attention graph information from the Transformer model to perform topological data analysis as a feature to identify automated text. Bakhtin et al. [12] found that the method based on a bidirectional encoder can achieve the strongest performance.

3. Our Method

3.1. Text Representation

To capture the rich semantic information in the text, we leverage a non-autoregressive pre-trained model renowned for its commendable performance in text semantic representation to encode sentences. The non-autoregressive pre-trained model provides an extensive array of linguistic, syntactic, and lexical knowledge for downstream tasks through unsupervised training on a substantial corpus during the pre-training phase. The underlying structure of the non-autoregressive pre-trained model involves a multi-layer bidirectional Transformer encoder. Specifically, we opt for XLM-RoBERTa as our preferred text encoding backbone. Given an input sentence S which is composed of a sequence of tokens $\{w_1, w_2, w_3, \dots, w_n\}$, the semantic representation h_i corresponding to w_i encoded by the XLM-RoBERTa pre-trained model is:

$$h_i = \text{Encoder}(w_i) \quad (1)$$

where $h_i \in R^m$ and m denotes the dimension of semantic representation.

3.2. Syntax-based Contrastive Learning

The dependency syntactic tree consists of the syntactic features of a sentence. There are significant differences in syntactic features between automated texts and human-written texts. More precisely, automated texts generated based on linguistic rules tend to be syntactically more reasonable, and its corresponding dependency syntactic tree is clearer than that of human-written texts. Unlike traditional methods based on feature extraction, we do not extract syntactic features directly, while exploiting dependency syntactic information to implicitly change the semantic space distribution of the model. We propose syntax-based contrastive learning to reduce the distance between each token and its related tokens in the semantic space by considering dependencies between tokens over the dependency syntactic tree, which ensures that the distribution of samples in the semantic space tends to be consistent with the shape of the dependency syntactic tree. The strategy makes the semantic space of automated text more distinguishable from that of human-written text to overcome the difficulty of automated text identification. We adopt the spacy tool [13] to extract the dependency syntactic information of all sentences and generate the dependency syntactic tree. Note that the dependency syntactic tree could be simplified to an undirected graph. For a token w_i , we treat the edge-connected tokens as its associated tokens in an undirected graph, subsequently defined as positive samples set P of w_i in syntax-based contrastive learning. The contrastive loss of w_i is formulated as follows:

$$L_{sbc_i} = -\frac{1}{|P|} \sum_{p \in P} \log \frac{\exp(\frac{\text{sim}(h_i, h_p)}{\tau})}{\sum_{k \in I \setminus \{i\}} \exp(\frac{\text{sim}(h_i, h_k)}{\tau})} \quad (2)$$

where I is the subscript list of tokens in the sentence sequence. $\text{sim}(\cdot)$ indicates the cosine similarity function and τ denotes a scalar temperature parameter. The overall contrastive loss for a sentence sequence is:

$$L_{sbc} = \frac{1}{n} \sum_{i=1}^n L_{sbc_i} \quad (3)$$

3.3. Text Identification

We proceed with text classification by leveraging the semantic representation associated with the token “[CLS]” within the given sentence. The representation is utilized as the sentence’s overall feature representation, which is subsequently fed into a linear classifier with a softmax function. To address subtask 1, it is imperative to construct a classifier with a designated output dimension of 2. For subtask 1, the predicted probabilities are:

$$y^1 = \text{softmax}(W_1^T \cdot h_{[\text{CLS}]} + b_1) \quad (4)$$

where W_1 and b_1 are learnable parameters, and y^1 is the predicted probability of subtask 1. The cross-entropy loss is employed to calculate the loss that penalizes the predicted class probability based on how far it is from the actual expected value. The cross-entropy loss function of subtask 1 L_{ce}^1 is defined as:

$$L_{ce}^1 = - \sum_{j=1}^2 e_j^1 \log y_j^1 \quad (5)$$

where e^1 is the one-hot encoding of the sample’s actual expected value of subtask 1. Likewise, we design a classifier with an output dimension of 6 in subtask 2. The predicted probabilities and cross-entropy loss of the subtask 2 are formulated as follows:

$$y^2 = \text{softmax}(W_2^T \cdot h_{[\text{CLS}]} + b_2) \quad (6)$$

$$L_{ce}^2 = - \sum_{j=1}^6 e_j^2 \log y_j^2 \quad (7)$$

where W_2 and b_2 are learnable parameters, and y^2 is the predicted probability of subtask 2.

3.4. Loss

Cross-entropy loss function and the optimized contrastive loss function are combined together by a weighting coefficient α :

$$L = \alpha \cdot L_{ce} + (1 - \alpha) \cdot L_{sbc}, \quad (8)$$

where L_{ce} and L_{sbc} represent the cross-entropy loss function and the contrastive loss function respectively, and $L_{ce} \in \{L_{ce}^1, L_{ce}^2\}$. Our training target is to minimize the loss L .

4. Experiments

4.1. Experimental Setup

All experiments are conducted based on the NVIDIA A30 24-GB GPU. We utilize pytorch [14] and transformers [15] to build our models. The feed-forward layer is initialized using weights drawn from a truncated normal distribution, characterized by a standard deviation of $2e-2$, while the bias is initialized to zero. A fixed initial learning rate of $2e-5$ is consistently applied throughout the experiments. The maximum sequence length is set to 128, indicating the prescribed limit on the number of tokens in a sentence. To facilitate training, a warmup proportion of $1e-3$ is employed, denoting the fraction of the total training time during which the learning rate remains low and gradually increases. This technique has been observed to yield advantageous outcomes in the training process. The training episodes are executed over the course of 10 epochs with a batch size of 8. For the syntax-based model, we select the small-scale English model (en_core_web_sm-3.5.0) and the Spanish model (es_core_news_sm-3.5.0) respectively.

In our experiments, we employ a 5-fold cross validation to ensure a fair assessment of the effectiveness of strategies, which entails dividing the datasets into five subsets to construct an ensemble model with enhanced generalization capabilities. Specifically, four subsets are designated for training, while the remaining subset is utilized for verification. The evaluation result of strategy effectiveness is derived from averaging the results obtained from the five cross models. The weighted coefficient α has great influence on the performance of the model. We searched for hyper-parameters in the range $\{\alpha | 0.005 \leq \alpha \leq 0.1\}$. The optimal weights of English subtask 1, Spanish subtask 1, English subtask 2 and Spanish subtask 2 are 0.1, 0.1, 0.01 and 0.005 respectively.

4.2. Experimental results

Table 1

Main results.

Subtask	Language	Model	Five-fold Cross Validation	Test
1	English	XLM-RoBERTa	91.30	55.21
		SCL	92.48	57.84
		Merge	-	56.34
	Spanish	XLM-RoBERTa	91.13	59.58
		SCL	91.74	58.83
		Merge	-	59.07
2	English	XLM-RoBERTa	54.90	59.27
		SCL	57.61	59.70
		Merge	-	59.86
	Spanish	XLM-RoBERTa	58.44	60.39
		SCL	58.90	60.61
		Merge	-	60.93

We conduct experiments in two subtasks of English and Spanish respectively, and the experimental results are shown in Table 1. Overall, our proposed method SCL achieve a significant

improvement over the XLM-RoBERTa model.

In terms of 5-fold cross validation, our proposed method consistently outperforms XLM-RoBERTa across four distinct tasks. Notably, in subtask 2 of English, the leverage of SCL yields the most substantial enhancement, resulting in a remarkable improvement of 2.71 in accuracy compared to the XLM-RoBERTa model.

In the comprehensive evaluation conducted on the test set, the SCL method achieves consistent gains over the XLM-RoBERTa model across three tasks, except for subtask 1 of Spanish, which suggests that our approach is effective. Specifically, in subtask 1 of English, the SCL method achieves an accuracy of 57.84 and XLM-RoBERTa demonstrates an accuracy of 59.58 in subtask 1 of Spanish. Subsequently, we further explore the potential of probabilistic fusion by combining the SCL models with the XLM-RoBERTa models. However, the merging strategy only proved effective in subtask 2, resulting in an accuracy of 59.86 and 60.93 for English and Spanish, respectively.

5. Conclusion

Our study aims to enhance the syntax representation capability of the neural network model, which exhibits remarkable performance in identifying automated texts. In the IberLEF 2023 AuTextification task, which focuses on Automated Text Identification, we introduce a novel syntax-based contrastive learning method, which empowers the neural network model to explicitly capture syntactic features, thus enhancing performance in identifying automated texts. In the evaluation phase, our SCL method demonstrates outstanding performance by securing the third and fifth place in Task 2 for English and Spanish, respectively.

In subsequent investigations, we will explore more advanced dependency syntactic models and superior pre-trained models to optimize the performance of automated text identification models, striving for enhanced outcomes and heightened efficiency.

Acknowledgments

This work was supported by the Guangdong Philosophy and Social Science Foundation (No. GD20CWY10), the National Social Science Fund of China (No. 22BTQ045), and the Science and Technology Program of Guangzhou (No.202002030227).

References

- [1] E. Crothers, N. Japkowicz, H. Viktor, Machine generated text: A comprehensive survey of threat models and detection methods, 2023. [arXiv:2210.07321](https://arxiv.org/abs/2210.07321).
- [2] S. M. Jiménez-Zafra, F. Rangel, M. Montes-y Gómez, Overview of IberLEF 2023: Natural Language Processing Challenges for Spanish and other Iberian Languages, *Procesamiento del Lenguaje Natural* 71 (2023).
- [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, in: *Procesamiento del Lenguaje Natural*, Jaén, Spain, 2023.

- [4] E. Crothers, N. Japkowicz, H. Viktor, P. Branco, Adversarial robustness of neural-statistical features in detection of generative transformers, in: 2022 International Joint Conference on Neural Networks (IJCNN), IEEE, 2022. URL: <https://doi.org/10.1109%2Fijcnn55064.2022.9892269>. doi:10.1109/ijcnn55064.2022.9892269.
- [5] L. Fröhling, A. Zubiaga, Feature-based detection of automated language models: tackling gpt-2, gpt-3 and grover, *PeerJ Computer Science* 7 (2021) e443.
- [6] A. Holtzman, J. Buys, L. Du, M. Forbes, Y. Choi, The curious case of neural text degeneration, 2020. [arXiv:1904.09751](https://arxiv.org/abs/1904.09751).
- [7] I. Solaiman, M. Brundage, J. Clark, A. Askell, A. Herbert-Voss, J. Wu, A. Radford, G. Krueger, J. W. Kim, S. Kreps, M. McCain, A. Newhouse, J. Blazakis, K. McGuffie, J. Wang, Release strategies and the social impacts of language models, 2019. [arXiv:1908.09203](https://arxiv.org/abs/1908.09203).
- [8] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., Language models are unsupervised multitask learners, *OpenAI blog* 1 (2019) 9.
- [9] R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, Y. Choi, Defending against Neural Fake News, Curran Associates Inc., Red Hook, NY, USA, 2019.
- [10] L. Zhuang, L. Wayne, S. Ya, Z. Jun, A robustly optimized BERT pre-training approach with post-training, in: Proceedings of the 20th Chinese National Conference on Computational Linguistics, Chinese Information Processing Society of China, Huhhot, China, 2021, pp. 1218–1227. URL: <https://aclanthology.org/2021.ccl-1.108>.
- [11] L. Kushnareva, D. Cherniavskii, V. Mikhailov, E. Artemova, S. Barannikov, A. Bernstein, I. Piontkovskaya, D. Piontkovski, E. Burnaev, Artificial text detection via examining the topology of attention maps, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021, pp. 635–649. URL: <https://aclanthology.org/2021.emnlp-main.50>. doi:10.18653/v1/2021.emnlp-main.50.
- [12] A. Bakhtin, S. Gross, M. Ott, Y. Deng, M. Ranzato, A. Szlam, Real or fake? learning to discriminate machine from human generated text, *CoRR abs/1906.03351* (2019). URL: <http://arxiv.org/abs/1906.03351>. [arXiv:1906.03351](https://arxiv.org/abs/1906.03351).
- [13] M. Honnibal, I. Montani, *spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing*, *To appear* 7 (2017) 411–420.
- [14] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, *Advances in neural information processing systems* 32 (2019).
- [15] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al., Huggingface’s transformers: State-of-the-art natural language processing, *arXiv preprint arXiv:1910.03771* (2019).