Classify and Label the Content from the Chinese Text Snippets of the LLM-Generated Text Detection

Kongqiang Wang[1*], Qingli Tan[2], and Peng Zhang[1]
[1]School of Information Science and Engineering, Yunnan University, 650500, Kunming, Yunnan, China
[2]College of Ecology and Environment, Yunnan University, 650500, Kunming, Yunnan, China
*Corresponding author: wangkongqiang60@gmail.com

Abstract

The ability to understand Chinese text snippets content is an essential component of human-like artificial intelligence, as text snippets content greatly influence human cognition, decision making, and social interactions. In addition to intention recognition in Chinese text snippets, the task of identifying the potential categories behind an individual's Chinese text state in LLM-Generated text snippets is of great importance in many application scenarios. The main content of our research is content classification and labeling from Chinese text snippets of LLM-Generated text detection, which aims at assign a content classification label to Chinese text snippets taken from LLM-Generated text snippets. Each snippet contains many sentences and corresponds to one of the predefined categories based on the LLM-Generated text snippets. We used a context-based text prediction method and combine with a pre-trained model or machine learning (ML) model. During this process, we repeatedly tested different pre-trained models and machine learning (ML) models in an effort to achieve the best results. The best result on the testp1 set was a macro-averaged F1-Score of 0.5746 and on the testp2 set was a macro-averaged F1-Score of 0.5869 by using *distilbert/distilbert-base-uncased*. We have reached the most advanced level in this field compared with other participant models. The project code is available from `https://github.com/WangKongQiang/NLPCC2026_Task6`.

Index Terms— Text Content Analysis, Text Classification, Multi-category Labeling, Pre-trained Model, Machine Learning.

1 Introduction

Understanding text snippets content is crucial to achieve human-like artificial intelligence, as Chinese text snippets are intrinsic to humans life and significantly influence our cognition, decision-making, and social interactions. Chinese LLM-generated text is an important form of human communication and contains a large amount of information. Furthermore, given that Chinese LLM-generated text snippet in its natural form is textual modality, many studies have explored textual modality intention recognition in Chinese LLM-generated text using language modalities [1].

The rapid development of large language models (LLMs) has given rise to a series of challenges, including the generation of disinformation, the spread of harmful content, and various forms of misuse [2]. Against this backdrop, the efficient discrimination between LLM-generated text and human-written text has become an urgent and critical research issue in the field of natural language processing (NLP). While remarkable progress has been made, relevant research has largely focused on English, systematic and technical exploration for the Chinese remain scarce [3]. NLPCC 2026 Shared Task 6 aims to fill this gap, build more robust Chinese LLM-generated text detectors, and advance research and real-world applications in this field within the Chinese.

Following the success of the 1st Shared Task on LLM-Generated Text Detection (NLPCC 2025) [4], the 2nd Shared Task on LLM-Generated Text Detection in 2026 features significant upgrades: the task formulation has been expanded from binary to ternary classification. Specifically, in addition to distinguishing between human-written text and LLM-generated text, a new category for identifying LLM-refined text has been introduced, which better aligns with real-world application scenarios of LLMs. Participating teams are required to design and implement text detection algorithms based on the training data provided to achieve accurate classification and will undergo rigorous stress testing. Phase 1 is an open-test leaderboard. Phase 2 hidden-test evaluation will be announced separately.

To address these challenges, this study draws upon principles from information theory and causal reasoning to construct a novel computational model tailored to Chinese LLM-Generated texts. By incorporating both statistical properties of language and causal relationships among Chinese text snippets, the proposed model framework aims to enhance the interpretability and analytical accuracy of automated systems. Ultimately, this work seeks to lay a solid theoretical and methodological foundation for the automated analysis of Chinese LLM-generated texts, contributing to more reliable and scalable evaluation of Chinese LLM-generated texts performance.

We proposing numerous well-designed pipeline systems. Moreover, we applied advanced pre-trained language models for Chinese LLM-generated texts analysis

and achieved promising results. These pre-trained models include *models–distilbert–distilbert-base-uncased*, and *models–microsoft–deberta-v3-base* and machine learning (ML) models include *XGBoost*, *SVM*, *Random_Forest*, *Logistic_Regression*.

2 Related Works

2.1 The Evolution of Deep Learning in Chinese Text Classification

Text classification, as a core task in natural language processing (NLP), has undergone a significant transformation from traditional machine learning (ML) to deep learning (DL). Early studies mainly relied on Bag-of-Words models and TF-IDF feature engineering methods [5], which performed well in short text classification but had difficulty capturing complex semantic relations. The introduction of convolutional neural networks (CNNs) has brought new breakthroughs to text classification. Yoon Kim et al. [6] proved that multi-scale convolutional kernels can effectively extract local semantic features. Subsequently, recurrent neural networks, especially long short-term memory networks (LSTM), have demonstrated advantages in processing long sequential texts [7]. The emergence of the Transformer architecture has completely transformed the technical landscape of text classification. BERT [8] has achieved significant performance improvements on multiple text classification tasks through bidirectional encoder and mask language model pre-training. Subsequent improved models such as RoBERTa [9] and DeBERTa [10] have further promoted the development in this field. However, these general pre-trained models still have domain adaptability issues in professional domain text processing, especially when dealing with LLM-Generated text with special language features [11].

2.2 The Application of Multi-task Learning in Text Processing

Multi-task learning enhances the performance of related tasks by sharing underlying representations and has been widely applied in the field of text processing. The MT-DNN framework proposed by Xiaodong Liu et al. [12] demonstrates the effectiveness of multi-task pre-training in natural language understanding tasks. The pioneering work of Ronan Collobert and Jason Weston [13] demonstrated that the joint training of part-of-speech tagging, named entity recognition, and semantic role tagging can enhance the performance of each sub-task. In the field of document analysis, Zichao Yang et al. [14] proposed a hierarchical attention network to handle both document classification and sentence importance assessment tasks simultaneously. However, the existing multi-task learning methods mainly focus on the combination of tasks at the grammatical and semantic levels, and rarely involve the joint modeling of content understanding and LLM-Generated text evaluation, which provides an innovative space for this study.

2.3 Text Analysis of LLM-Generated Text Detection

The rapid advancement of large language models (LLMs) has significantly improved the fluency and coherence of machine-generated text [15], making it increasingly difficult to distinguish from human-written content. Consequently, text analysis has become a fundamental research direction for LLM-generated text detection [16]. Early studies primarily focused on handcrafted linguistic features, including lexical richness, syntactic complexity, readability metrics, and stylometric characteristics. These methods exploit statistical differences between human and machine writing but often struggle to generalize across different domains and newly emerging language models.

With the success of deep learning (DL), representation-based approaches have gradually replaced manually designed features. Pre-trained language models such as BERT [8] , RoBERTa [9], and DeBERTa [10] encode contextual semantic information into dense representations, enabling classifiers to capture subtle semantic and syntactic patterns indicative of machine-generated text. Recent studies [17] have further enhanced these representations through contrastive learning, multi-task learning, and domain adaptation, thereby improving robustness under distribution shifts. In addition, some methods [18] leverage sentence-level and document-level embeddings to model global textual consistency rather than relying solely on local linguistic cues.

Beyond semantic representations, recent research [19] has investigated intrinsic statistical properties of LLM-generated text. Perplexity-based methods measure the likelihood assigned by language models, while burstiness and entropy analyses characterize the reduced variability and smoother token distributions commonly observed in generated text. Other studies exploit token probability distributions, hidden-state representations, and generation-specific artifacts to identify subtle inconsistencies introduced during autoregressive decoding. These approaches often demonstrate strong performance when the source model is known but tend to suffer from limited generalization across unseen LLMs.

Despite these advances, current text analysis methods still face significant challenges. As state-of-the-art LLMs produce increasingly human-like text, superficial lexical or statistical differences become less discriminative, and detectors trained on specific generators often exhibit substantial performance degradation under cross-domain or cross-model settings [20]. Therefore, developing robust text analysis techniques capable of capturing model-invariant semantic and structural characteristics remains an important direction for future LLM-generated text detection research.

3 Task and Dataset

3.1 Task Description

This LLM-Generated Text Detection Task is based on NLPCC 2026 Share Task 6: The Second Shared Task on LLM-

Figure 1: This is an illustrative example for our task objective. Training Set contains data from 4 types of LLMs and 2 domains. Specifically, data sources include news and academic writing, and generation models include GPT-4, Qwen, ChatGLM, and Baichuan. The training set contains a total of 19,634 pairs of samples. Each pair contains four fields: "ID", "HWT", "LGT", and "HLT". "ID" identifies the sample number, while "HWT", "LGT", and "HLT" represent "Human-written text", "LLM-generated text", and "LLM-refined text" respectively.

Generated Text Detection. This is a ternary text classification task, which has certain similarities with other multi-category classification tasks [21].

LLM-Generated Text Detection Task: Content Classification. Participants are tasked with assigning a content class to text snippets from Chineses LLM-Generated text datasets. The Chinese text snippets are sampled from different sections of news and academic writing. Each snippet corresponds to one of the predefined category criteria in LLM-Generated field, and the goal is to classify the snippet according to its corresponding criterion section. The official evaluation metric for this task is the macro-averaged F1-Score. There are 3 classes for this LLM-Generated text detection task [22]. Their specific situations are shown in Table 1.

Our team mainly participated in LLM-Generated Text Detection Task: Content Classification and achieved significant development results by using pre-trained model technology.

3.2 Task Dataset

The training data will consist of Chinese text snippets from LLM-Generated Text Detection dataset, which contains the 19,634 pairs entire paragraph content, an illustrative example of the task objective is shown in Figure 1. The training set for this task is primarily sampled and adapted from the CUDRT (TIST 2026) dataset (Chinese subset: Complete 25 ratio). The Phase 1 test set (open evaluation) is now available; the Phase 2 test set (closed evaluation) will be released in the later stages of the competition. The evaluation dataset is extended and constructed based on the DetectRL benchmark (NeurIPS 2024) framework, containing multiple generation models and domain data to ensure the authenticity and challenge of the evaluation scenarios [23].

4 Methodology

In this task, the main model training framework we use is a supervised training framework based on text modality. In this framework, the entire paragraph content corresponding to the HWT, LGT, and HLT labels is part of the training text context, for *models–microsoft–deberta-v3-base*, we will demonstrate the usage of the multi-backend capabilities of Keras-Core and KerasNLP for the detect LLM-Generated text infernece. For the fine-tuning of the *models–distilbert–distilbert-base-uncased* and the construction of feature columns for machine learning (ML) methods including *XGBoost*, *SVM*, *Random_Forest*, *Logistic_Regression*. As shown in the Figure 2.

4.1 Data Preprocessing

The training dataset is first loaded from the CSV file. Missing text entries are replaced with empty strings to avoid invalid samples during feature extraction. All text instances are converted into a unified string format to ensure compatibility with different feature extraction methods. Finally, the dataset is organized into text–label pairs for subsequent feature extraction and model training.

4.2 Dataset Division

The preprocessed dataset is randomly divided into a training set and a test set using an 80:20 ratio. A fixed random seed (random_state = 42) is adopted to ensure reproducibility of the experimental results. The training set is used for model optimization, while the test set is employed to evaluate the model performance.

4.3 Metrics Equations

The official evaluation metric for this task is the macro-averaged F1-score (Macro-F1), which equally weights the performance of each class regardless of its frequency. For each class i, the precision and recall are defined as

$$\text{Precision}_i = \frac{TP_i}{TP_i + FP_i}, \tag{1}$$

$$\text{Recall}_i = \frac{TP_i}{TP_i + FN_i}, \tag{2}$$

where TP_i, FP_i, and FN_i denote the numbers of true positives, false positives, and false negatives for class i, respectively.

The F1-score for class i is computed as the harmonic mean of precision and recall:

$$F1_i = \frac{2 \times \text{Precision}_i \times \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i}. \tag{3}$$

Finally, the Macro-F1 score is obtained by averaging the F1-scores over all C classes:

$$\text{Macro-F1} = \frac{1}{C} \sum_{i=1}^{C} F1_i, \tag{4}$$

where C denotes the total number of classes. Since each class contributes equally to the final score, Macro-F1 is particularly suitable for evaluating classification performance on imbalanced datasets.

ID	Text	Classification	Label
Unique identifier of the sample	Text content of the sample	Human-written text	0
		LLM-refined text	1
		LLM-generated text	2

Table 1: The specific details of the 3 categories in the content classification of LLM-Generated Text Detection Task: Ternary Text Classification.

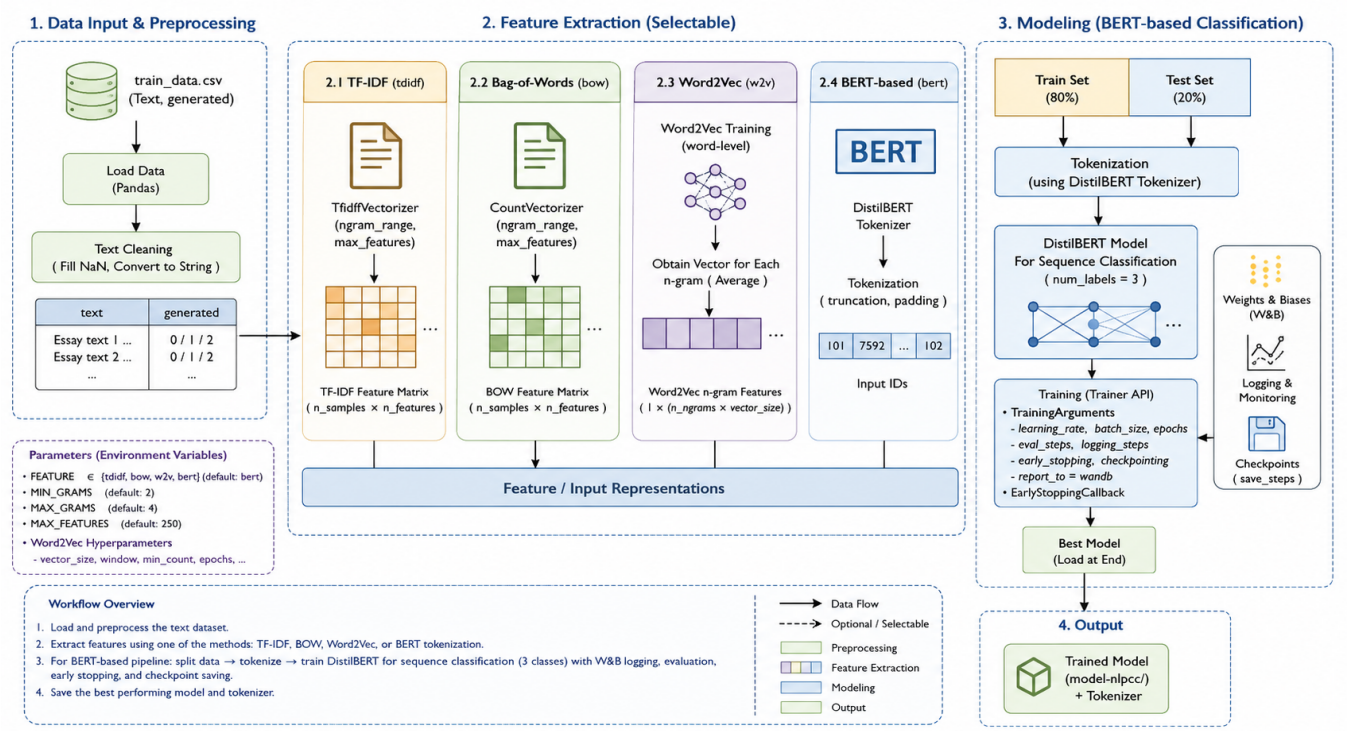

Figure 2: The $left$ subfigure is the data input & preprocessing flow, the $middle$ subfigure is feature extraction, The input representations can be selected from TF-IDF (tdidf), Bag-of-Words (bow), Word2Vec (w2v), and Bert-based (bert). And the $right$ subfigure is the text fine-tuning training process of DistilBERT pre-trained model in total.

4.4 Fine-tuning Pre-trained Models

First, we choose the pre-trained model as microsoft/deberta-v3-base[1]. DeBERTa improves the BERT [8] and RoBERTa [9] models using disentangled attention and enhanced mask decoder. With those two improvements, DeBERTa outperform RoBERTa on a majority of NLU tasks with 80GB training data.

In DeBERTa V3, they further improved the efficiency of DeBERTa using ELECTRA-Style pre-training with Gradient Disentangled Embedding Sharing. Compared to DeBERTa, their V3 version significantly improves the model performance on downstream tasks. You can find more technique details about the new model from their paper [24]. The hyperparameters we provided to the model are shown in Table 2. The macro-averaged F1-Score of this pre-trained language model on our testp1 and testp2 datasets are 0.3718 and 0.3772. The overall model training framework is shown in Figure 3.

Second, we choose the pre-trained model as distilbert/distilbert-base-uncased[2]. DistilBERT is a transformers model, smaller and faster than BERT [8], which was pretrained on the same corpus in a self-supervised fashion, using the BERT base model as a teacher. This means it was pretrained on the raw texts only, with no humans labelling them in any way (which is why it can use lots of publicly available data) with an automatic process to generate inputs and labels from those texts using the BERT base model. More precisely, it was pretrained with three objectives:

1. Distillation loss: the model was trained to return the same probabilities as the BERT base model.

2. Masked language modeling (MLM): this is part of the original training loss of the BERT base model. When

Table 2: Model hyperparameter Settings for $microsoft/deberta-v3-base$

Hyperparameter	Value
seed	42
num_folds	5
epochs	3
batch_size	6
sequence_length	200
learning_rate	5×10^{-6}
dropout	0.2

[1]https://huggingface.co/microsoft/deberta-v3-base
[2]https://huggingface.co/distilbert/distilbert-base-uncased

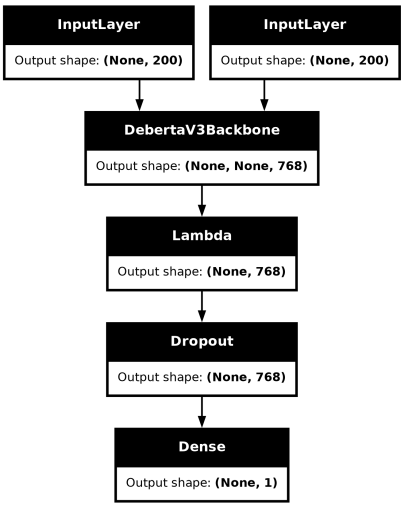

Figure 3: The overall model training framework figure is plotted by Keras.utils, and the figure is showed the text fine-tuning training process of *models–microsoft–deberta-v3-base* pre-trained models in total.

taking a sentence, the model randomly masks 15% of the words in the input then run the entire masked sentence through the model and has to predict the masked words. This is different from traditional recurrent neural networks (RNNs) that usually see the words one after the other, or from autoregressive models like GPT which internally mask the future tokens. It allows the model to learn a bidirectional representation of the sentence.

3. Cosine embedding loss: the model was also trained to generate hidden states as close as possible as the BERT base model.

This way, the model learns the same inner representation of the English language than its teacher model, while being faster for inference or downstream tasks. The hyperparameters we provided to the model are shown in Table 3. The macro-averaged F1-Score of this pre-trained language model on our testp1 and testp2 dataset are 0.5746 and 0.5869.

Table 3: Model hyperparameter Settings for $distilbert/distilbert-base-uncased$

Hyperparameter	Value
save_steps	500
eval_steps	500
num_train_epochs	10
per_device_train_batch_size	16
per_device_eval_batch_size	16
learning_rate	3×10^{-5}
weight_decay	0.01

5 Result

We also utilized the Character_count, Word_count, Average_word_length, Sentence_count, Average_sentence_length_split, Average_sentence_length_tokenize, Mean_word_length, Unique_word_count, Stopword_count, Polarity, and Subjectivity features of Chinese text snippets for statistical machine learning (ML), in order to perform the LLM-Generated Text Detection Task. We used different machine learning (ML) models include XGBoost, SVM, Random_Forest, Logistic_Regression.

The macro-averaged F1-Score of these different pre-trained models and machine learning (ML) models on our official testp1 dataset are shown in Table 4. The macro-averaged F1-Score indicators were measured through the evaluation script score.py and the gold standard data testp1_testing_label.json used for scoring.

Table 4: The macro-averaged F1-Score of these different pre-trained models and machine learning (ML) models on our official testp1 dataset.

Pre-trained or Machine Learning Model	Macro F1-Score
distilbert/distilbert-base-uncased	**0.5746**
microsoft/deberta-v3-base	0.3718
xgboost	0.3533
svm	0.3329
random forest	0.3463
logistic regression	0.3333

The macro-averaged F1-Score of these different pre-trained models and machine learning (ML) models on our official testp2 dataset are shown in Table 5. the macro-averaged F1-Score indicators were measured through the evaluation script score.py and the gold standard data testp2_testing_label.json used for scoring.

Table 5: The macro-averaged F1-Score of these different pre-trained models and machine learning (ML) models on our official testp2 dataset.

Pre-trained or Machine Learning Model	Macro F1-Score
distilbert/distilbert-base-uncased	**0.5869**
microsoft/deberta-v3-base	0.3772
xgboost	0.3369
svm	0.3436
random forest	0.3216
logistic regression	0.3491

Below are the final rankings for the LLM-Generated text

detection task, see Table 6. In the official ranking. Our team ranked 6th in the LLM-Generated text Detection Task-Content Classification with a macro-averaged F1-Score index of 0.5869.

Table 6: LLM-Generated Text Detection Task - Chinese Text Snippets Content Classification.

Rank	Team Name	Macro F1-Score
1	evildetect	0.8888
2	linglings	0.8393
3	xiwangdoudui	0.7745
4	zzunlp_zhao	0.7475
5	team_from_bit	0.7299
6	**wangkongqiang**	**0.5869**
7	xiaoeeen	0.4753
8	yiyiyi	0.4553
9	jianglifeng	0.4430
10	zutnlp	0.3979
11	test	0.3334
12	wking1688	0.2975
13	huishingcheung	0.2383

6 Conclusion

The macro-averaged F1-Score of these different pre-trained models and machine learning (ML) models on our official testp1 and testp2 dataset is not particularly outstanding. A major reason for this is that the pre-trained models cannot handle text features well. Since we provide the context of the text to the pre-trained models by concatenating the context content, it may result in the data content not being well utilized. Another point is that there are fewer pre-trained models in Chinese than in English, and the models used are relatively limited [25]. If large language models (LLMs) [26] such as GPT 4 and Llama 3 are used for fine-tuning, and then prompt word prediction is used, it is believed that the prediction can be more accurate. Better text feature extraction is also a future direction for improving the performance and macro-averaged F1-Score indicators of the different prediction models.

References

[1] Junchao Wu, Runzhe Zhan, Derek F Wong, Shu Yang, Xinyi Yang, Yulin Yuan, Lidia S Chao, "Detectrl: Benchmarking llm-generated text detection in real-world scenarios," Advances in Neural Information Processing Systems (37), 100369–100401, 2024.

[2] Uladzimir Sidarenka, "Potts: the potsdam twitter sentiment corpus," In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), 1133–1141, 2016.

[3] Hang Yan, Junqi Dai, Xipeng Qiu, Zheng Zhang, "A unified generative framework for aspect-based sentiment analysis," arXiv preprint arXiv:2106.04300, 2021.

[4] Junchao Wu, Runzhe Zhan, Qianli Wang, Yulin Yuan, Lidia S Chao, Derek F Wong, "Overview of the NLPCC 2025 Shared Task 1: LLM-Generated Text Detection," CCF International Conference on Natural Language Processing and Chinese Computing, Springer, 263–274, 2025.

[5] Gerard Salton, Michael J McGill, "Introduction to modern information retrieval," McGraw-Hill, 1986.

[6] Yoon Kim, "Convolutional neural networks for sentence classification," In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1746–1751, 2014.

[7] Sepp Hochreiter, Jürgen Schmidhube, "Long short-term memory," Neural computation, 9(8):1735–1780, 1997.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805, 2018.

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

[10] Pengcheng He, Xiaodong Liu, Jianfeng Gao, Weizhu Chen, "Deberta: Decoding-enhanced bert with disentangled attention," In International Conference on Learning Representations, 2021.

[11] Zhen Tao, Yanfang Chen, Dinghao Xi, Zhiyu Li, Wei Xu, "Toward Reliable Detection of LLM-Generated Texts: A Comprehensive Evaluation Framework with CUDRT," ACM Transactions on Intelligent Systems and Technology 2(17):1-35, 2026.

[12] Xiaodong Liu, Pengcheng He, Weizhu Chen, Jianfeng Gao, "Multi-task deep neural networks for natural language understanding," In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 4487–4496, 2019.

[13] Ronan Collobert, Jason Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," In Proceedings of the 25th international conference on Machine learning, 160–167, 2008.

[14] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy, "Hierarchical attention networks for document classification," In Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies, 1480–1489. 2016.

[15] Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Lidia Sam Chao, Derek Fai Wong, "A Survey on LLM-Generated Text Detection: Necessity, Methods, and Future Directions," Computational Linguistics 51, no. 1: 275-338, 2025.

[16] Junchao Wu, Runzhe Zhan, Derek F Wong, Shu Yang, Xuebo Liu, Lidia S. Chao, Min Zhang, "Who wrote this? the key to zero-shot llm-generated text detection is gecscore," In Proceedings of the 31st International Conference on Computational Linguistics, pp. 10275-10292. 2025.

[17] Julian Risch, Ralf Krestel, "Toxic comment detection in german," In Proceedings of the Conference on Natural Language Processing (KONVENS), 91–101, 2020.

[18] Raphael Scheible, Fabian Thomczyk, Patric Tippmann, Victor Jaravine, Martin Boeker, "Gottbert: a pure german language model," arXiv preprint arXiv:2012.02110, 2020.

[19] Xin Chen, Junchao Wu, Shu Yang, Runzhe Zhan, Zeyu Wu, Ziyang Luo, Di Wang, Min Yang, Lidia S. Chao, Derek F. Wong, "Repreguard: Detecting llm-generated text by revealing hidden representation patterns," Transactions of the Association for Computational Linguistics 13: 1812-1831, 2025.

[20] Gregor Wiedemann, Steffen Remus, Arpan Chawla, Chris Biemann, "Transfer learning for affective computing: A case study on valence-arousal prediction," In Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, 9–15, 2018.

[21] Jakob Prange, Charlott Jakob, Patrick Göttfert, Raphael Huber, Pia Wenzel, Annemarie Friedrich, "Overview of the SustainEval 2025 Shared Task," In Proceedings of the 21st Conference on Natural Language Processing (KONVENS 2025): Workshops, Hildesheim, Germany. HsH Applied Academics, 2025.

[22] Grigorios Tsoumakas, Ioannis Katakis, "Multilabel classification: An overview," International Journal of Data Warehousing and Mining (IJDWM), 3(3):1–13, 2007.

[23] Yichun Yin, Yangqiu Song, Ming Zhang, "Document-level multi-aspect sentiment classification as machine comprehension," In Proceedings of the 2017 conference on empirical methods in natural language processing, 2044–2054, 2017.

[24] Pengcheng He, Jianfeng Gao, Weizhu Chen, "DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing," https://arxiv.org/abs/2111.09543, 2023.

[25] Telmo Pires, Eva Schlinger, Dan Garrette, "How multilingual is multilingual bert?" In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy. Association for Computational Linguistics, 4996–5001, 2019.

[26] Wenxuan Zhang, Yue Deng, Bing Liu, Sinno Jialin Pan, Lidong Bing, "Sentiment analysis in the era of large language models: A reality check," In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, 1–15, 2023.