YNU-HPCC at SemEval-2025 Task 6: Using BERT Model with R-drop for Promise Verification

Dehui Deng, You Zhang*, Jin Wang, Dan Xu and **Xuejie Zhang**
School of Information Science and Engineering
Yunnan University
Kunming, China
Contact: dehuideng@stu.ynu.edu.cn, yzhang0202@ynu.edu.cn

Abstract

This paper presents our participation in the SemEval-2025 task 6: multinational, multilingual, multi-industry promise verification. The SemEval-2025 Task 6 aims to extract Promise Identification, Supporting Evidence, Clarity of the Promise-Evidence Pair, and Timing for Verification from the promises made to businesses and governments. Using these data to verify whether companies and governments have fulfilled their promises. In this task, we focus on the English dataset. Our model introduces regularization dropout based on the BERT-base model to focus on the stability of non-target classes, improve the robustness of the model, and ultimately improve the indicators. Our approach obtained competitive results in task. The code of the paper is available at: https://github.com/xxkaras/SemEval-2025-Task-6.

1 Introduction

Tracking and verifying promises made by businesses and governments is essential for fostering accountability and trust. However, assessing whether these promises are upheld is often hindered by the complexities of different industries, languages, and countries. SemEval-2025 Task 6 (Chen et al., 2025) introduces a novel approach to multilingual, multi-industry promise verification to tackle this challenge. This task is designed to extract key information from promises made to businesses and governments, such as identifying supporting evidence for the promise, evaluating the clarity of the promise-evidence relationship, and determining the appropriate timing for verification. By leveraging this data, the goal is to assess the degree to which these promises have been honored.

The subtasks of SemEval-2025 Task 6 can be divided into binary classification and multi-classification tasks. Text classification is an important task in natural language processing (NLP) (Cambria and White, 2014) that aims to organize text into predefined categories automatically. According to the different types of tasks, text classification can be divided into binary, multi-class, and multi-label. In binary classification tasks, the text is divided into two mutually exclusive categories, such as spam and non-spam; multi-class classification tasks divide the text into multiple mutually exclusive categories, such as classified news; and multi-label classification tasks allow each text to belong to multiple categories at the same time, such as multi-label sentiment analysis. This process usually includes data reconstruction, feature extraction, model training, and evaluation. The text must be cleaned and segmented in the data reconstruction stage to convert it into a machine-processable format. Next, the feature extraction step converts the text into a numerical vector. Commonly used methods include the bag-of-words model, TF-IDF, Word2Vec, and BERT (Koroteev, 2021). The labeled data trains the machine learning or deep learning model in the model training phase. Common models include support vector machine (SVM) (Wang and Hu, 2005) and naive Bayes (Naive) (Webb et al., 2010). After the training is completed, the model needs to be evaluated on the test set, and the performance is evaluated using indicators such as accuracy, precision, recall, and F_1-score. Text classification (Gasparetto et al., 2022) has many applications, including sentiment analysis, spam filtering, topic classification, public opinion monitoring, etc. With the development of deep learning technology, pre-trained language models (Min et al., 2023) (such as BERT and GPT (Achiam et al., 2023)) have performed well in text classification tasks, especially for large-scale data sets (Bzdok et al., 2019) and complex tasks.

SemEval-2025 Task 6 contains the following four subtasks:

- Subtask 1 Promise Status (PS): Identify

*Corresponding author.

Proceedings of the The 19th International Workshop on Semantic Evaluation (SemEval-2025), pages 1905–1911

whether there is a promisein the sentence

- Subtask 2 Evidence Status (ES): Identify whether there is supporting evidence for the promise in the sentence
- Subtask 3 Verification Timeline (VT): Identify whether there is a verification time for the supporting evidence in the sentence
- Subtask 4 Evidence Quality (EQ): Identify the clarity of the evidence related to the promise in the sentence

Subtask 1 and subtask 2 are binary classification tasks, and subtask 3 and subtask 4 are multi-classification tasks.

SemEval-2025 Task 6: PromiseEvalMultinational, Multilingual, MultiIndustry Promise Verification competition features a novel multilingual dataset comprising English, French, Chinese, Japanese, and Korean, designed to evaluate corporate ESG promises and their implementation. Our team has participated in this competition and focus on English datasets. To improve the robustness of our model, we introduced a regularization dropout mechanism based on BERT-base. This method focuses on enhancing the stability of non-target classes, ultimately boosting the model's performance and generalizability. Our approach delivered competitive results in this task, showcasing the potential of incorporating regularization techniques into promise verification tasks. This paper discusses our methodology, results, and insights into how these advancements contribute to more effective promise verification.

The rest of this paper is organized as follows. Section 2 introduces the related work before our study for this task. Section 3 gives an overview of our system for this task. Section 4 presents the specific details of our system and discusses the experimental results. The conclusions are drawn in Section 5.

2 Ralated Work

Some companies use misleading information to create an overly positive environmental image, a practice known as greenwashing. To address the greenwashing phenomenon and the challenge of evaluating corporate promises, Seki et al. (2024) proposed ML-Promise, the first multilingual dataset for deep promise verification, including Chinese, English, French, Japanese, and Korean. The dataset provides key training samples for related technologies in the field of natural language processing (NLP) to verify corporate promises in environmental, social, and governance (ESG) reports. The dataset contains promise data from different countries and companies, with structured labels to facilitate the identification and evaluation of corporate promises, supporting evidence, supporting evidence quality, and verification time of the promise. The labels are divided into four main aspects to ensure a comprehensive assessment of corporate promises.

Hillebrand et al. (2023) proposed a recommendation system based on natural language processing (NLP) to automatically analyze the credibility and substantive content of corporate sustainability reports. The study adopted a BERT-based multi-task learning framework, combined with rule matching and attention mechanism, to extract promise statements from reports and classify their credibility, and external data was used for cross-validation. The traditional manual review process was enhanced through multimodal analysis, explainable recommendations, and cross-domain adaptation. Experiments show that the system achieves excellent results in the task of sustainability report detection. The study provides a technical reference for the automated verification of corporate promises.

To promote in-depth verification of promises, this paper aims to apply models in the field of natural language processing to promise verification, and to monitor corporate promises and their compliance with ESG promises, as well as the promises and compliance of public figures.

3 System Overview

Our system is based on the BERT-based model, and the regularization technology RDrop (Regularized Dropout) (Wu et al., 2021) has been added to implement it. The overall structure of our system consists of four modules, which are described below.

Input layer. In this layer, we build text processing tools for performing text preprocessing and word embedding (Jiao and Zhang, 2021). The input is a data frame containing text data (obtained from the train data test set). The text data is converted to the BERT input format (token IDs) using functions provided by BERT. The input is padded to ensure consistent input length within a batch. The dataset

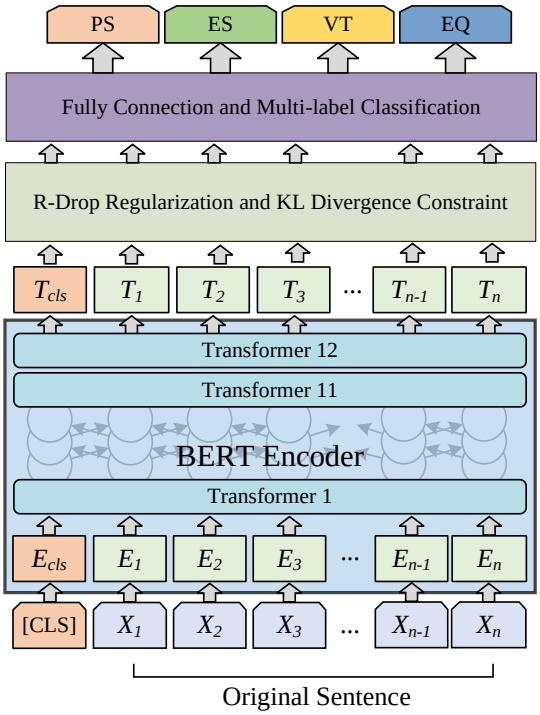

Figure 1: Multi-label classification system

is converted to the Hugging Face dataset format using API provided by Hugging Face.

Context encoder. BERT (Devlin et al., 2019) is a natural language processing (NLP) model proposed by Devlin et al. in 2018. The main novelty of BERT lies in its bidirectional encoder structure, which can simultaneously model text from left to right and from right to left, allowing it to better capture contextual information in the text. BERT uses a pre-training method and can handle various text-processing tasks through fine-tuning. BERT is based on the Transformer (Zheng et al., 2022) architecture and adopts bidirectional. This means that when learning context, BERT considers the information before and after the word and simultaneously analyzes the relationship between the left and right sides. Hence, it has a stronger ability to understand semantics. BERT first pre-trains with large-scale corpus to learn general language knowledge. On this basis, BERT can adapt to specific tasks (such as text classification, question answering, named entity recognition, etc.) through fine-tuning (Wang et al., 2024) to achieve excellent performance. In the pre-training stage, BERT uses static masking to process text. In each training cycle, BERT randomly masks some words in the input text and trains the model to predict these masked words. In this way, BERT can learn the deep relationship between words. This module mainly uses the pre-trained BERT model to complete the context encoder (Pathak et al., 2016).

Dropout Layer. Dropout (Srivastava et al., 2014) is a common regularization technique used in the training process of neural network models to prevent overfitting of the model. It was proposed by Geoffrey Hinton et al. in 2014 and is widely used in deep learning. The core idea of Dropout is to discard some neurons in the neural network randomly. In each training, randomly select some neurons and their connections to make them *invalid* or *not involved in the calculation* in the current iteration. This operation helps to reduce the complex dependencies between neurons and forces each part of the model to learn features independently, thereby improving the model's generalization ability.

Linear Classifier. In our model, the linear classifier (Bai et al., 2022) maps the context information extracted by BERT to the target label space, such as promise status, evidence status, verification time, and evidence quality. It classifies label through the fully connected layer. The classifier combines the dropout layer (regularization) to avoid overfitting and trains the model through the cross-entropy loss so that the model can predict the score of each label based on the input text.

The input size of this layer is 768 (the hidden layer size of BERT-BASE), and the output size is 4 (the number of labels)

Loss Function. In our model, the calculation of the loss function includes the following parts: Cross-Entropy Loss (Mao et al., 2023), which is used to calculate the gap between logits and labels. Kullback-Leibler (KL) divergence (Van Erven and Harremos, 2014) calculates the consistency between model outputs. As a regularization term, It encourages the model to produce similar predictions in different training cycles. Cross-entropy loss measures the difference between the probability distribution predicted by the model and the distribution of the true label. The calculation formula for Cross-Entropy Loss is as follows:

$$CrossEntropyLoss = -\sum y_i log(p_i) \quad (1)$$

Kullback-Leibler Divergence is a measure of the difference between two probability distributions. Our model uses KL divergence to calculate the difference between logits and kl_logits (logits calculated by BERT for the second time), which is added to the loss function as a regularization term. The calculation formula for KL divergence is as follows:

$$KL\,Divergence(P||Q) = \sum P(x) log(\frac{P(x)}{Q(x)}) \quad (2)$$

KL Divergence Loss (Cui et al., 2025) calculates the KL divergence between logits and kl_logits and the KL divergence between kl_logits and logits. The final KL Divergence Loss is the average of the two, encouraging the model's output to be consistent across different training steps. The total loss of the model is a weighted sum of the cross-entropy loss, the KL divergence loss, and another cross-entropy loss.

The application of our model in SemEval-2025 Task 6 is discussed below. The first part of this task is extracting semantic information from a given tweet text. We call it sentence classification. In the upstream task, we used a pre-trained BERT model for the upstream sentence information extraction task. In the downstream task, we used BERT for multi-label classification, where the model's goal is to predict multiple labels based on the input text (such as a promise statement or a business text). Specifically, the model needs to predict the following labels: promise_status, verification_timeline, evidence_status, and evidence_quality. The prediction of each label is independent, and the output features of BERT, i.e., the pooled output of [CLS] token, are mapped to the logits of each label. These scores are converted into probability values for each label through the sigmoid activation function, indicating whether the label exists.

4 Experiments

Datasets. SemEval-2025 Task 6 uses the Multilingual Dataset for Corporate Promise Verification as the dataset that needs to extract from the promise text whether there is a promise, supporting evidence, the quality of the promise and the time limit for the verification of the promise. Table 1 shows the labels of each subtask. The dataset is in JSON format and Parquet format. Because the TSV format is more convenient for data processing, we convert both the training set and the dataset to the TSV format.

Evaluation Methods. The evaluation metric for SemEval-2025 Task 6 is accuracy. Each subtask in the task is evaluated for accuracy. Leaderboard scores are aggregated scores for all subtasks.

Implementation Details. To evaluate the effectiveness of our proposed method, we conducted a series of experiments. All experiments were performed under identical conditions to ensure consistency and comparability of results. Our model was separately trained and tested for label prediction across four subtasks. We split the training data into training and validation sets at an 8:2 ratio. For label encoding, we employed LabelEncoder to convert textual category labels into numerical values (e.g., encoding "No" as 0 and "Yes" as 1 in the promise status task). We constructed a custom BERT classification model based on BertPreTrainedModel. The model architecture incorporates a dropout layer and a linear classification layer (with output dimensions corresponding to the number of label categories) on top of the pooled BERT representations. The number of labels varied across tasks (e.g., binary classification for promise status and evidence status with 2 labels, 5 labels for verification timeline, and 4 labels for evidence quality). To enhance model robustness, we combined cross-entropy loss with KL-divergence loss, computing consistency regularization through dual forward propagation. Train-

Subtask	Label
Promise Status	Yes/No
Evidence Status	Yes/No
Verification Timeline	within 2 years/2-5 years/longer than 5 years/other/nan
Evidence Quality	Clear/Not Clear/Misleading/nan

Table 1: Label in each subtask

	Learning Rate	Train Epochs	Train Batch Size	Warmup Steps	Weight Decay
Promise Status	3.6672	4	4	200	0.0881
Evidence Status	7.2486	4	8	200	0.0156
Verification Timeline	4.2965	4	4	500	0.0442
Evidence Quality	3.9092	2	4	1000	0.03

Table 2: The optimal parameters after fine-tuning

	w/ optimal params	w/o optimal params
Promise Status	0.7875	0.7625
Evidence Status	0.6753	0.625
Verification Timeline	0.4252	0.3875
Evidence Quality	0.3	0.2125

Table 3: performance comparison of model in promise verification with and without optimal parameters

ing was conducted using optimized hyperparameters. Upon completion of training, predictions were made on the test set by selecting the class with the maximum logit value. Finally, the numerical predictions were converted back to their corresponding textual labels. This approach ensures both task adaptability and improved generalization performance through our loss design.

Hyperparameters Finetuning. We train and evaluate the model with different hyperparameters in the objective function.Our hyperparameter tuning employs Bayesian optimization through Optuna, systematically exploring the learning rate (1e-5 to 5e-5), batch sizes during training (4, 8, 16), total number of training epochs (2 to 5), number of warmup steps for the learning rate scheduler (100 to 1000, in steps of 100), and strength of weight decay (0.0 to 0.1). The optimization goal is to maximize validation accuracy over 10 trials, each performing complete model training using Hugging Face's Trainer API. It determines the optimal hyperparameter combination through multiple experiments, as shown in Table 2.

Ablation Study. To evaluate the impact of regularized dropout in ESG promise verification, we conducted ablation study by systematically removing regularized dropout from parts of the model and deeply analyzed the contribution of regularized dropout to the experimental results. The Table 4 compares the performance of the BERT-BASE model combined with R-drop and the Bert-Base model after fine-tuning in each subtask. To ensure the accuracy and fairness of the experiment, both models use the parameters fine-tuned by Optuna. The results prove that R-drop is effective in improving the accuracy in ESG promise verification.

	w/ R-drop	w/o R-drop
Promise Status	0.7875	0.75
Evidence Status	0.6753	0.6125
Verification Timeline	0.4252	0.3756
Evidence Quality	0.3	0.2752

Table 4: performance comparison of BERT model in promise verification with and without R-drop

Results and Analysis. In the competition leaderboard, our system ranked 9 in the English leaderboard. As indicated, our method is effective.This is mainly because BERT is combined with R-Drop to improve generalization ability, enhance robustness, optimize training process, improve task performance and reduce overfitting.Our model has an accuracy of 0.7875 in the promise status subtask and0.6753 in the evidence status subtask. The model performs well in the binary classification task and can judge the existence of the promise and the existence of evidence in the text with high accuracy.However, the accuracy in the verification timeline and evidence quality subtasks is only 0.4252 and 0.3 respectively, which means that the model has difficulty in clearly judging the clarity of the given evidence in relation to the promise and specific deadline for promise verification completion.

5 Conclusions

This paper proposes a promise verification label classification model base BERT with R-drop for SemEval-2025 Task 6. The model successfully solves the classification problems of commitment existence status, evidence existence status, evidence quality and commitment verification timeline.In the promise verification task, R-Drop simulates different sub-networks by randomly dropping neurons, making the model more robust to input perturbations and making the final output labels of each sub-task more accurate. Although small models combined with few-shot learning can effectively solve binary classification tasks such as promise status and evidence status, the effect on verification timeline and evidence quality tasks is not ideal. Future works will apply, text data augmentation, such as using synonym replacement, random insertion or back-translation techniques, to expand the training set. Moreover, weighted loss functions or oversampling/undersampling techniques to balance the distribution of labels and avoid the model being biased towards the majority class label.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant Nos. 61966038 and 62266051. We would like to thank the anonymous reviewers for their constructive comments.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Wenqiang Bai, Jin Wang, and Xuejie Zhang. 2022. Ynu-hpcc at semeval-2022 task 4: Finetuning pretrained language models for patronizing and condescending language detection. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 454–458.

Danilo Bzdok, Thomas E Nichols, and Stephen M Smith. 2019. Towards algorithmic analytics for large-scale datasets. *Nature Machine Intelligence*, 1(7):296–306.

Erik Cambria and Bebo White. 2014. Jumping nlp curves: A review of natural language processing research. *IEEE Computational intelligence magazine*, 9(2):48–57.

Chung-Chi Chen, Yohei Seki, Hakusen Shu, Anaïs Lhuissier, Juyeon Kang, Hanwool Lee, Min-Yuh Day, and Hiroya Takamura. 2025. SemEval-2025 task 6: Multinational, multilingual, multi-industry promise verification. In *Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025)*, Vienna, Austria. Association for Computational Linguistics.

Jiequan Cui, Zhuotao Tian, Zhisheng Zhong, Xiaojuan Qi, Bei Yu, and Hanwang Zhang. 2025. Decoupled kullback-leibler divergence loss. *Advances in Neural Information Processing Systems*, 37:74461–74486.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Andrea Gasparetto, Matteo Marcuzzo, Alessandro Zangari, and Andrea Albarelli. 2022. A survey on text classification algorithms: From text to predictions. *Information*, 13(2):83.

Lars Hillebrand, Maren Pielka, David Leonhard, Tobias Deußer, Tim Dilmaghani, Bernd Kliem, Rüdiger Loitz, Milad Morad, Christian Temath, Thiago Bell, et al. 2023. sustain. ai: a recommender system to analyze sustainability reports. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, pages 412–416.

Qilu Jiao and Shunyao Zhang. 2021. A brief survey of word embedding and its recent development. In *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, volume 5, pages 1697–1701. IEEE.

Mikhail V Koroteev. 2021. Bert: a review of applications in natural language processing and understanding. *arXiv preprint arXiv:2103.11943*.

Anqi Mao, Mehryar Mohri, and Yutao Zhong. 2023. Cross-entropy loss functions: Theoretical analysis and applications. In *International conference on Machine learning*, pages 23803–23828. PMLR.

Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. 2023. Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Computing Surveys*, 56(2):1–40.

Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. 2016. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544.

Yohei Seki, Hakusen Shu, Anaïs Lhuissier, Hanwool Lee, Juyeon Kang, Min-Yuh Day, and Chung-Chi Chen. 2024. Ml-promise: A multilingual dataset for corporate promise verification. *arXiv preprint arXiv:2411.04473*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Tim Van Erven and Peter Harremos. 2014. Rényi divergence and kullback-leibler divergence. *IEEE Transactions on Information Theory*, 60(7):3797–3820.

Haifeng Wang and Dejin Hu. 2005. Comparison of svm and ls-svm for regression. In *2005 International conference on neural networks and brain*, volume 1, pages 279–283. IEEE.

Jie Wang, Jin Wang, and Xuejie Zhang. 2024. Ynu-hpcc at semeval-2024 task 9: Using pre-trained language models with lora for multiple-choice answering tasks. In *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, pages 471–476.

Geoffrey I Webb, Eamonn Keogh, and Risto Miikkulainen. 2010. Naïve bayes. *Encyclopedia of machine learning*, 15(1):713–714.

Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, Tie-Yan Liu, et al. 2021. R-drop: Regularized dropout for neural networks. *Advances in neural information processing systems*, 34:10890–10905.

Guangmin Zheng, Jin Wang, and Xuejie Zhang. 2022. Ynu-hpcc at semeval-2022 task 6: Transformer-based model for intended sarcasm detection in english and arabic. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 956–961.