CSCU at SemEval-2025 Task 6: Enhancing Promise Verification with Paraphrase and Synthesis Augmentation: Effects on Model Performance

Kittiphat Leesombatwathana, Wisarut Tangtemjit, Dittaya Wanwaree
Department of Mathematics and Computer Science, Faculty of Science,
Chulalongkorn University, Bangkok, Thailand
{6534404823, 6534460923}@student.chula.ac.th, dittaya.w@chula.ac.th

Abstract

The Promise Verification task involves classifying corporate commitments and their supporting evidence into multiple categories, and this study presents significant findings in machine learning and environmental, social, and governance (ESG) evaluation. We compared various approaches, including zero-shot and six-shot GPT-4o, Support Vector Machine (SVM) with Multilingual E5 text embeddings, and fine-tuned DistilBERT. Six-shot GPT-4o achieved the best performance, while zero-shot GPT-4o struggled due to a lack of in-context examples, with both models demanding considerable computational resources and reliance on external APIs. In contrast, SVM proved to be an economical alternative, effective in binary classification tasks especially when enhanced by data augmentation techniques. Although DistilBERT is more resource-intensive, it offers a scalable solution balancing efficiency and accuracy. Our experiments show that data augmentation, utilizing paraphrasing and synthesis techniques, generally improves performance, though it has limitations in clarity evaluation. These findings emphasize the trade-offs between performance, cost, and efficiency in selecting models for the Promise Verification task, offering valuable insights for the field.

1 Introduction

In recent years, the significance of environmental, social, and governance (ESG) commitments has increased, with stakeholders demanding greater corporate transparency and accountability. Accurately assessing these commitments is essential for evaluating a company's dedication to sustainable practices and ethical standards. However, the complexity and volume of these commitments present significant challenges in verification, highlighting the urgent need for effective evaluation methods.

One major obstacle in this area is the lack of comprehensive datasets designed for training models in promise verification tasks. This deficiency hampers the development of robust systems capable of effectively evaluating ESG commitments, emphasizing the need for further research and data collection. This gap presents a potential avenue for future work.

To address this issue, we have employed text augmentation techniques to improve the quality of promise verification tasks. Our approach involves synthesizing data using large language models to generate augmented datasets that maintain the original semantics while enhancing the training corpus. This strategy is not just a tool; it is a necessity aimed at improving the model's ability to accurately classify ESG-related promises.

Our focus is on English-language reports, specifically targeting the following classification tasks:

1. Promise Identification (PI): Detecting explicit commitments made by corporations.

2. Supporting Evidence (SE): Linking promises to corroborative actions or statements.

3. Clarity of Promise-Evidence Pair (CPEP): Evaluating the transparency and comprehensibility of the promise and its supporting evidence.

4. Timing for Verification (TV): Determining the specified time frame for fulfilling the promise.

By implementing text augmentation techniques, we aim to enhance the performance of promise verification systems and contribute to more reliable assessments of corporate ESG commitments.

2 Related Work

The BERT family of models has become a cornerstone for text classification tasks due to its high performance and efficiency. Devlin et al. introduced BERT (Devlin et al., 2019), which has since been adapted into variants such as DistilBERT, which

Proceedings of the The 19th International Workshop on Semantic Evaluation (SemEval-2025), pages 1935–1947
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

maintains 97% of BERT's performance while using fewer parameters (Sanh et al., 2020). Domain-specific adaptations, like ClimateBERT, highlight the benefits of pretraining on specialized datasets (Webersinke, 2022). Our approach utilizes DistilBERT as the base model, capitalizing on its efficiency while achieving comparable performance improvements through targeted data augmentation.

In recent years, large pre-trained models like GPT-4o[1] have been widely utilized for text classification tasks, including zero-shot and few-shot learning. A study by (Seki et al., 2024) demonstrated the effectiveness of GPT-4o for multi-task classification. They proposed a retrieval-augmented generation (RAG) approach that incorporates Multilingual E5 embeddings (Wang et al., 2024).

To help rating institutions assess the ESG scores of target companies, it is essential to categorize company-related documents into the corresponding ESG categories: Environment (E), Social (S), and Governance (G). However, not all company documents are relevant to these ESG topics. ESG-BERT (Goel et al., 2022), a BERT model that has been fine-tuned on ESG reports, has demonstrated superior performance in ESG-related classification tasks compared to the original BERT model. Additionally, this fine-tuned model shows strong results in ESG sentiment analysis (Kannan and Seki, 2023).

Although fine-tuning models like ESG-BERT can yield high performance, this process can be computationally expensive and resource-intensive. As a more cost-effective alternative, recent approaches employ in-context learning and few-shot prompting with large GPT models. These methods eliminate the need for extensive retraining. They have been successfully used to classify the impact of ESG initiatives, evaluate the credibility of ESG claims, and track the timelines of corporate promises (Tian and Chen, 2024; Chen et al., 2025).

Data augmentation is widely used to enhance dataset size and diversity, improving model generalization, particularly in low-resource settings. Wei and Zou demonstrated that training on a dataset with 50% augmented data can achieve performance comparable to using the entire dataset (Wei and Zou, 2019).

3 System overview

3.1 Dataset Description

The dataset includes five languages: English, French, Korean, Japanese, and Traditional Chinese. For this study, we will focus exclusively on the English dataset. The training and test sets contain a total of 400 instances, addressing the tasks of Promise Identification, Supporting Evidence, Clarity of the Promise-Evidence Pair, and Timing for Verification. The distribution of classes and additional details can be found in the overview paper (Chen et al., 2025).

3.2 Baselines

We implement few-shot learning baselines using Multilingual E5 Embeddings (Wang et al., 2024). First, we convert all training samples into embeddings with the Multilingual E5 model and store them in a vector database[2]. Next, we conduct zero-shot inference and six-shot retrieval-augmented generation (RAG). In this approach, we retrieve the six most similar training samples along with their corresponding answers and include them in a prompt for GPT-4o. These baselines were initially proposed in (Seki et al., 2024) and serve as our primary comparison points when evaluating our proposed approaches (see Appendix A for the prompts used).

3.3 Support Vector Machine Approach

We use text embeddings from the Multilingual E5 model as input for a Support Vector Machine (SVM) classifier, implemented with the scikit-learn library[3]. SVMs are particularly well-suited for high-dimensional data, such as text embeddings, as they can effectively handle large feature spaces and capture non-linear relationships.

The model is configured with `kernel="rbf"` to model non-linear patterns, `class_weight="balanced"` to address class imbalance, and `random_state=0` to ensure reproducibility. To further mitigate class imbalance, we use the RandomOverSampler from the imbalanced-learn library[4], which oversamples the minority class by duplicating existing samples.

[1] GPT-4o is a state-of-the-art language model developed by OpenAI.

[2] We utilize ChromaDB, an open-source vector database optimized for efficient embedding storage and retrieval. https://www.trychroma.com/

[3] scikit-learn: https://scikit-learn.org/stable/

[4] imbalanced-learn: https://imbalanced-learn.org/stable/

Although this approach may not outperform more complex models, it offers a computationally efficient and cost-effective solution for text classification tasks, making it a viable option when resources are limited.

3.4 Fine-tuned BERT Approach

In this study, we use DistilBERT, a smaller and more efficient variant of the Bidirectional Encoder Representations from Transformers (BERT) model. DistilBERT is specifically designed to maintain the language understanding capabilities of the original BERT architecture while significantly reducing computational complexity. For each task, we initialize a separate DistilBERT model with pre-trained weights from the `distilbert-base-uncased` and fine-tune it independently by optimizing the classification head for sequence classification.

To ensure consistency and reproducibility in our experiments, we systematically fix key hyperparameters, including the learning rate, batch size, and weight decay. This approach minimizes variability that could arise from hyperparameter tuning. Each model is fine-tuned for 10 epochs to achieve sufficient convergence while keeping computational efficiency in mind.

A comprehensive overview of the hyperparameter settings and training configuration can be found in Appendix E. The fine-tuning process is implemented using the *Hugging Face Transformers* library.[5]

4 Experiments

4.1 Dataset and Preprocessing

We utilize the PromiseEval 2025 Task 6 dataset (Chen et al., 2025), which consists of textual instances annotated for four classification tasks: Promise Identification (PI), Supporting Evidence (SE), Clarity of Promise-Evidence Pair (CPEP), and Timing for Verification (TV). To maintain experimental consistency, we randomly select 80% of the training data (320 instances) to create five independent training splits.

In addition to the original dataset, we implement a data synthesis process using Gemini-2.0-Flash[6] to generate augmented samples. We employ two distinct augmentation strategies:

- Paraphrase Augmentation: Large language models (LLMs) create paraphrased versions of the original text while preserving the labels.
- Synthesis Augmentation: LLMs produce new text that may alter the label as long as the synthesized content remains semantically aligned.

For the augmented training data, we combine 320 original samples with 320 augmented samples, maintaining a 1:1 ratio. Text preprocessing is conducted using the DistilBERT tokenizer, which incorporates fixed truncation and padding (with a maximum sequence length of 512 tokens) to ensure compatibility with transformer-based architectures.

4.2 Implementation Details

In our fine-tuned BERT approach[7], we employ cross-entropy loss as the objective function for training the classification models. This loss function is well-suited for both multi-class and binary classification tasks, as it quantifies the difference between the predicted probability distribution and the actual labels. Mathematically, the cross-entropy loss is defined as follows:

$$\mathcal{L} = -\sum_{i=1}^{N} y_i \log(\hat{y}_i) \tag{1}$$

where y_i represents the ground-truth label for the i-th sample, $\hat{y}_i$ is the model's predicted probability for the correct class, and N denotes the total number of samples in the dataset.

4.3 Evaluation Metrics

Following the protocol in Seki et al. (2024), we employ macro F1-score as the primary evaluation metric. The macro F1-score is computed for each task independently before averaging, providing a balanced measure in the presence of imbalanced class distributions. All experimental configurations are evaluated on a fixed test dataset, and results are aggregated over five runs.

5 Results

5.1 Variability in GPT-4o Responses

Our experiments showed slight variations in the results of the zero- and six-shot GPT-4o baselines in five runs. While the overall trend remained consistent, each API call yielded slightly different responses. We attribute this variability to the inherent

[5] `https://huggingface.co/docs/transformers`

[6] `https://gemini.google.com/`

[7] `https://github.com/wdittaya/promiseeval2025.git`

Table 1: Accuracies with Standard Deviation from Different Models

Model	Tasks			
	PI	TV	SE	CPEP
Baselines				
GPT zero-shot	**0.7860** ± **0.0026**	0.4151 ± 0.0070	0.7752 ± 0.0055	0.3624 ± 0.0073
GPT six-shot	0.7476 ± 0.0034	**0.4593** ± **0.0122**	**0.8091** ± **0.0075**	**0.4401** ± **0.0129**
Support Vector Machine Approach				
SVM	0.6949 ± 0.0159	0.4030 ± 0.0131	0.7188 ± 0.0131	0.3841 ± 0.0129
SVM$_{\text{para}}$	0.7081 ± 0.0135	**0.4184** ± **0.0110**	0.7217 ± 0.0121	0.3930 ± 0.0195
SVM$_{\text{syn}}$	**0.7216** ± **0.0167**	0.4093 ± 0.0169	**0.7266** ± **0.0096**	**0.3986** ± **0.0167**
Fine-tuned BERT Approach				
DistilBert	**0.6752** ± **0.0208**	0.3953 ± 0.0304	0.7747 ± 0.0095	0.4163 ± 0.0222
DistilBert$_{\text{para}}$	0.6646 ± 0.0278	**0.4297** ± **0.0344**	0.7716 ± 0.0091	**0.4259** ± **0.0082**
DistilBert$_{\text{syn}}$	0.6704 ± 0.0209	0.4252 ± 0.0369	**0.7835** ± **0.0152**	0.4103 ± 0.0222

Table 2: Overall Model Ranking by Average Performance Across All Tasks

Model	Average Score
GPT six-shot	0.6140
GPT zero-shot	0.5846
DistilBert$_{\text{para}}$	0.5730
DistilBert$_{\text{syn}}$	0.5724
DistilBert	0.5654
SVM$_{\text{syn}}$	0.5640
SVM$_{\text{para}}$	0.5603
SVM	0.5502

randomness in the GPT-4o model, which can produce different outputs for the same input due to the nondeterministic nature of the model. Additionally, we suspect that the default `temperature=1.0` parameter, which controls the level of randomness in the generation process, may contribute to these differences. When the temperature increases, the model generates more diverse and less predictable responses, which can explain the minor discrepancies observed in our results.

These variations, while minor, highlight the challenges of working with large language models in production environments, where outputs may not always be perfectly consistent. However, this behavior is expected and does not significantly affect the overall conclusions drawn from the experiments.

5.2 Overall Performance Comparison

Table 1 summarizes the mean and standard deviation of macro F1-scores across all models for four promise detection tasks, while Table 2 ranks the models by average performance. The models evaluated include GPT in zero- and six-shot configurations (averaged over five runs) and SVM/DistilBERT trained on an 80% split with three augmentation conditions: none, paraphrase, and synthesis. GPT-based methods show strong zero- and few-shot results, and data augmentation slightly improves SVM and DistilBERT performance.

As indicated in Table 1, GPT zero-shot excels in Promise Identification (PI), while GPT six-shot leads in Timing for Verification (TV), Supporting Evidence (SE), and Clarity of the Promise-Evidence Pair (CPEP).

Data augmentation consistently enhances SVM performance compared to non-augmented baselines. Synthesis augmentation produces the best results for PI and SE, while paraphrase augmentation shines in TV. DistilBERT with paraphrase augmentation ranks highest among non-GPT methods, followed closely by synthesis-augmented DistilBERT.

GPT-based approaches exhibit consistent performance with low standard deviations, while fine-tuned models show greater variability, indicating that augmentation can enhance performance but may also introduce inconsistencies. The CPEP task remains challenging for all models, with the best performance struggling to exceed an F1 score of

0.43. Additionally, synthesis-based augmentation ($\text{Group}_{\text{syn}}$) leads to higher variability, likely due to label noise, as shown in Table 1.

5.3 Data Augmentation Analysis

To better understand the reliability of the augmented datasets, we manually inspected the examples generated by the LLM (Gemini-2.0-Flash). For the Paraphrase Augmentation, where the goal was to rephrase the original text without altering its label, we observed that the LLM inadvertently changed the label in 14 out of 400 entries. This indicates that despite instructions to preserve labels during paraphrasing, some label noise was introduced, which could impact model training.

Conversely, for the Synthesis Augmentation, where the language model was allowed to both paraphrase and modify the label to balance class distributions, we found that only six entries had their labels changed. Ideally, we expected a greater number of label changes to better address the class imbalance. This suggests that the synthesis prompts might not have been sufficiently aggressive in encouraging label shifts.

Overall, these observations highlight that while LLM-based augmentation is a powerful tool, it introduces a degree of noise that must be carefully managed. In future work, improving the augmentation prompts or implementing post-processing strategies, such as rule-based filtering (e.g., if-else conditions to detect unintended label changes), could enhance the quality and reliability of the augmented data.

6 Conclusion

In this study, we explored several approaches to the Promise Verification tasks, including zero-and six-shot GPT-4o, SVM, and fine-tuned DistilBERT, with varying settings for data augmentation. Our results show that data augmentation generally improves performance for SVM and fine-tuned DistilBERT across tasks, with synthesis augmentation performing best for Promise Identification (PI) and Supporting Evidence (SE), while paraphrase augmentation yielded better results for Timing for Verification (TV). However, both augmentation techniques demonstrated mixed results for the Clarity of the Promise-Evidence Pair (CPEP) task, with paraphrase augmentation showing slight improvements but still remaining challenging overall, likely due to its multiclass nature and sensitivity to subtle semantic changes.

Among the models tested, six-shot GPT-4o outperformed all other approaches, establishing itself as a strong baseline for the Promise Verification task. Nevertheless, SVM and fine-tuned DistilBERT provide valuable alternatives, especially when computational efficiency or resource constraints are important. DistilBERT, in particular, offers a good balance between performance and efficiency, demonstrating that fine-tuning smaller models can achieve competitive results while being more resource-friendly than larger models like GPT-4o. Additionally, SVM provides a more economical solution for specific tasks, even if its performance is not always at the level of the more complex models.

These findings highlight the potential of combining synthesis augmentation with models like DistilBERT and SVM, suggesting that with further refinement, these models can be competitive alternatives to more significant, more resource-intensive approaches like GPT-4o. Future research could explore improving augmentation techniques, fine-tuning different models for better task-specific performance, and investigating more efficient strategies for multi-task text classification.

7 Future Work

Several directions could further enhance the findings of this study. First, exploring domain-specific language models such as FinBERT could be beneficial, as ESG-related texts often overlap with financial contexts. Fine-tuning FinBERT (Huang et al., 2023) for these tasks might lead to better representations and improved classification performance in finance.

Additionally, refining the prompting strategies for both paraphrase and synthesis augmentation is necessary to reduce the noise in generated data. In particular, improving prompts for paraphrase augmentation could help ensure that the original labels are preserved more reliably. Finally, developing a filtering system (e.g., rule-based validation mechanisms) to detect and correct label changes during paraphrase augmentation would be valuable for maintaining data quality and improving model robustness.

Acknowledgement

The manuscript is refined with ChatGPT and Grammarly.

References

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.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Tushar Goel, Vipul Chauhan, Suyash Sangwan, Ishan Verma, Tirthankar Dasgupta, and Lipika Dey. 2022. TCS WITM 2022@FinSim4-ESG: Augmenting BERT with linguistic and semantic features for ESG data classification. In *Proceedings of the Fourth Workshop on Financial Technology and Natural Language Processing (FinNLP)*, pages 235–242, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Allen H. Huang, Hui Wang, and Yi Yang. 2023. Finbert: A large language model for extracting information from financial text. *Contemporary Accounting Research*, 40(2):806–841.

Naoki Kannan and Yohei Seki. 2023. Textual evidence extraction for ESG scores. In *Proceedings of the Fifth Workshop on Financial Technology and Natural Language Processing and the Second Multimodal AI For Financial Forecasting*, pages 45–54, Macao. -.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *Preprint*, arXiv:1910.01108.

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. *Preprint*, arXiv:2411.04473.

Ke Tian and Hua Chen. 2024. ESG-GPT:GPT4-based few-shot prompt learning for multi-lingual ESG news text classification. In *Proceedings of the Joint Workshop of the 7th Financial Technology and Natural Language Processing, the 5th Knowledge Discovery from Unstructured Data in Financial Services, and the 4th Workshop on Economics and Natural Language Processing*, pages 279–282, Torino, Italia. Association for Computational Linguistics.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual e5 text embeddings: A technical report. *Preprint*, arXiv:2402.05672.

Nicolas Webersinke. 2022. Climatebert: A pretrained language model for climate-related text. In *AAAI 2022 Fall Symposium: The Role of AI in Responding to Climate Challenges*.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6382–6388, Hong Kong, China. Association for Computational Linguistics.

A Prompts Used in Baselines

A.1 Zero-Shot GPT-4o Prompt

You are an AI model that classifies paragraphs into four categories based on predefined labels.
Categories and Labels:
1. **Promise Status**: ‘Yes’ or ‘No’
2. **Verification Timeline**: ‘Already’, ‘Less than 2 years’, ‘2 to 5 years’, ‘More than 5 years’, ‘N/A’
3. **Evidence Status**: ‘Yes’ or ‘No’
4. **Evidence Quality**: ‘Clear’, ‘Not Clear’, ‘Misleading’, ‘N/A’

Task:
Classify the following paragraph into these four categories. Respond in JSON format.

Paragraph: {paragraph}

Output Format (JSON example):

```
{
"promise_status": "Yes or No",
"verification_timeline": "Already, Less than 2 years, 2 to 5 years, More than
5 years, or N/A",
"evidence_status": "Yes or No",
"evidence_quality": "Clear, Not Clear, Misleading, or N/A"
}
```

A.2 Six-Shot GPT-4o Prompt

You are an AI model that classifies paragraphs into four categories based on predefined labels.
Categories and Labels:
1. **Promise Status**: ‘Yes’ or ‘No’
2. **Verification Timeline**: ‘Already’, ‘Less than 2 years’, ‘2 to 5 years’, ‘More than 5 years’, ‘N/A’
3. **Evidence Status**: ‘Yes’ or ‘No’
4. **Evidence Quality**: ‘Clear’, ‘Not Clear’, ‘Misleading’, ‘N/A’

Examples:

Paragraph: {paragraph1}
- Promise Status: {label1_promise}
- Verification Timeline: {label1_verification}
- Evidence Status: {label1_evidence}
- Evidence Quality: {label1_quality}

Paragraph: {paragraph2}
- Promise Status: {label2_promise}
- Verification Timeline: {label2_verification}
- Evidence Status: {label2_evidence}
- Evidence Quality: {label2_quality}

⋮

Paragraph: {paragraph6}
- Promise Status: {label6_promise}
- Verification Timeline: {label6_verification}
- Evidence Status: {label6_evidence}
- Evidence Quality: {label6_quality}

Now classify the following paragraph:
{paragraph}

Output Format (JSON example):

```
{
"promise_status": "Yes or No",
"verification_timeline": "Already, Less than 2 years, 2 to 5 years, More than 5 years, or N/A",
"evidence_status": "Yes or No",
"evidence_quality": "Clear, Not Clear, Misleading, or N/A"
}
```

B Prompt Used in Data Augmentation

B.1 Paraphrase Augmentation Instruction

You're an ESG data synthesis and augmentation assistant. Your task is to **paraphrase** ESG-related text data while ensuring semantic meaning, logical structure, and contextual accuracy remain intact. The rewritten text must be diverse, realistic, and naturally aligned with the provided metrics.
Context of Metrics:

- **Verification Timeline**
  - "Already": ESG measures applied and with results that can already be verified.
  - "Less than 2 years": ESG measures whose results can be verified within 2 years.
  - "2 to 5 years": ESG measures whose results can be verified in 2 to 5 years.
  - "More than 5 years": ESG measures whose results can be verified in more than 5 years.
  - "N/A": Absence of promise removes the need for timeline verification.
- **Evidence Quality**
  - "Clear": Complete, logical, and intelligible evidence.
  - "Not Clear": Missing information, ranging from intelligible to superficial.
  - "Misleading": Irrelevant evidence used to distract.
  - "N/A": Absence of evidence negates the need for quality assessment.
- **Evidence Status**
  - "Yes": Evidence exists to back the promise.
  - "No": Evidence is absent.
- **Promise Status**
  - Binary classification of whether a segment qualifies as a promise ("Yes" or "No").

Paraphrasing Task:

1. **Rewrite and Maintain Semantic Meaning**
   - Generate a 1:1 paraphrased dataset while preserving the original meaning, logical structure, and assigned metrics.
   - Ensure natural variations in phrasing, word choice, and sentence structure while keeping the content consistent with ESG-related communications.
   - Include realistic company or agency names that align with the ESG context, such as "GreenFuture Initiative," "EcoFleet Logistics," or "Global Sustainability Group."
2. **Output Format**
   - Export the paraphrased dataset in a JSON file structured as follows:
   - `"URL"` and `"page_number"` fields must remain masked as `"xxx"`.
   - The `"data"` field should contain paraphrased text between 100-300 tokens, ensuring coherence and contextual accuracy.

3. **Guidelines**
   - Avoid altering the core intent or meaning of the original text.
   - Maintain consistency in tone, vocabulary, and language style used for ESG-related communications.
   - Do not introduce new information or change the original classification of any metric.
   - Ensure company or agency names fit naturally within the context of the ESG topic.

4. **Evaluation and Consistency**
   - Ensure that the paraphrased dataset aligns with the provided ESG classification metrics without modifying the dataset's original balance.
   - Include a summary in the output describing:
     - The paraphrasing strategy used.
     - Examples of rewritten company or agency names.
     - Any notable challenges in maintaining semantic accuracy.

Example: Input (Original):

```
[{
    "URL": "[URL]",
    "page_number": "[Page Number]",
    "data": "[Data]",
    "promise_status": "Yes",
    "verification_timeline": "2 to 5 years",
    "evidence_status": "Yes",
    "evidence_quality": "Clear"
}]
```

Output (Paraphrased Example):

```
[{
    "URL": "xxx",
    "page_number": "xxx",
    "data": "[Data]",
    "promise_status": "Yes",
    "verification_timeline": "2 to 5 years",
    "evidence_status": "Yes",
    "evidence_quality": "Clear"
}]
```

B.2 Synthesis Augmentation Instruction

You're an ESG data synthesis and augmentation assistant. Your task is to **generate diverse, realistic, and contextually accurate ESG-related text data** while balancing all target classes across four tasks. Ensure that the names of agencies and companies included in the data align naturally with the context and the described metrics.

Context of Metrics:

- **Verification Timeline**
  - "Already": ESG measures applied and with results that can already be verified.
  - "Less than 2 years": ESG measures whose results can be verified within 2 years.
  - "2 to 5 years": ESG measures whose results can be verified in 2 to 5 years.
  - "More than 5 years": ESG measures whose results can be verified in more than 5 years.
  - "N/A": Absence of promise removes the need for timeline verification.
- **Evidence Quality**
  - "Clear": Complete, logical, and intelligible evidence.
  - "Not Clear": Missing information, ranging from intelligible to superficial.
  - "Misleading": Irrelevant evidence used to distract.
  - "N/A": Absence of evidence negates the need for quality assessment.
- **Evidence Status**
  - "Yes": Evidence exists to back the promise.
  - "No": Evidence is absent.
- **Promise Status**
  - Binary classification of whether a segment qualifies as a promise ("Yes" or "No").

Synthesis Task:

1. **Rewrite and Paraphrase**
   - Generate a 1:1 dataset for each class, ensuring semantic meaning and logical structure remain intact.
   - Include realistic company or agency names that naturally fit the ESG context and metrics. For example, names like "GreenFuture Initiative," "EcoFleet Logistics," or "Global Sustainability Group" should align with the subject matter.
2. **Augment Across All Tasks**
   - Modify the dataset proportionally to balance positive and negative samples for **evidence_status** and **promise_status**.
   - Generate diverse examples for each class of **verification_timeline** and **evidence_quality**, avoiding over-representation of any class.

3. **Output Format**
   - Export data to a JSON file structured as follows:
   - `"URL"` and `"page_number"` fields must remain masked as `"xxx"`.
   - Ensure the `"data"` field contains text between 100-300 tokens, maintaining coherence and relevance.
   - The generated dataset must include realistic agency and company names to enrich contextual accuracy.

4. **Guidelines**
   - Retain the logical connections between the promise, evidence, and timeline metrics.
   - Maintain consistency in tone, vocabulary, and language style used for ESG-related communications.
   - Avoid introducing ambiguous, contradictory, or overly generic names or concepts.
   - Use the provided metrics exclusively for synthesis; do not introduce external or new metrics.

5. **Evaluation and Balance**
   - After synthesis, analyze the class proportions across tasks to ensure a balanced representation in the training data.
   - Include a summary in the output describing:
     - How class balance was achieved.
     - Examples of generated company or agency names.
     - Any discrepancies observed and the steps taken to address them.

Example: Input (Original):

```
[{
    "URL": "[URL]",
    "page_number": "[Page Number]",
    "data": "[DATA]",
    "promise_status": "Yes",
    "verification_timeline": "2 to 5 years",
    "evidence_status": "Yes",
    "evidence_quality": "Clear"
}]
```

Output (Paraphrased Example):

```
[{
    "URL": "xxx",
    "page_number": "xxx",
    "data": "[Data]",
    "promise_status": "Yes",
    "verification_timeline": "2 to 5 years",
    "evidence_status": "Yes",
    "evidence_quality": "Clear"
}]
```

C Examples of Data Augmentation and Synthetic Data

C.1 Original Data Sample

> Reliability of digital transaction channels 2022 highlights. We made structural and process changes in response to material system outages in the first half of 2022, enabling us to improve system stability and resilience. We experienced no material incidents in the second half of the year. No client-impacting cyber incidents or cyber-related losses were experienced in 2022. Our response and recovery times significantly improved in the second half of the year. We can successfully fail over (switch to a standby system) in the event of near misses, mitigating the risk of outages and impacts to the client. In South Africa, our customer NPS scores across all digital channels showed improvement in the second half of 2022. We improved our communications approach in the event of outages experienced, to ensure that we can effectively and timeously engage impacted customers. Group technology had positive engagements with the group board, regulators, and investors, to ensure clear understanding of what caused the outages in early 2022, and what we have done to improve system resilience.

Annotation:

- Promise Identification: *Yes*
- Timing for Verification: *Less than 2 years*
- Supporting Evidence: *Yes*
- Clarity of Promise-Evidence Pair: *Clear*

C.2 Paraphrase Augmentation Example

> Standard Bank Group enhanced system stability and resilience through structural and process adjustments following significant system outages in early 2022, resulting in no material incidents in the latter half of the year. There were no client-impacting cyber incidents or losses in 2022. Response and recovery times improved significantly, with successful failover capabilities to mitigate outage risks. Customer NPS scores improved across digital channels in South Africa. The bank improved communications during outages to effectively engage impacted customers. Group technology engaged with the board, regulators, and investors to ensure understanding of the outages and implemented improvements.

Annotation:

- Promise Identification: *Yes*
- Timing for Verification: *Less than 2 years*
- Supporting Evidence: *Yes*
- Clarity of Promise-Evidence Pair: *Clear*

C.3 Synthetic Data Generation Example

> Following significant system failures in early 2022, DigitalTrust Bank implemented structural and process improvements to enhance system stability. By the latter half of the year, no major incidents were reported, and cyber incidents were avoided. Recovery times improved significantly, and the bank enhanced its communication strategies during outages to keep customers informed. These efforts led to improved customer satisfaction scores across all digital channels, with positive feedback from regulators and investors.

Annotation:

- Promise Identification: *Yes*
- Timing for Verification: *Less than 2 years*
- Supporting Evidence: *Yes*
- Clarity of Promise-Evidence Pair: *Clear*

D Data Splitting Procedure

To ensure a robust evaluation, we performed five different train-test splits of the data set using the scikit-learn train_test_split function. Each split allocated 80% of the data for training and 20% for testing, with a different random_state for each split (random_state = 0, 21, 42, 63, 84) to introduce variability while maintaining reproducibility.

Note: Only the training portion (80%) was used for model training and evaluation. The test portion (20%) was not utilized in this study.

E Hyperparameters and Training Configuration

For training the fine-tuned DistilBERT models, we use a consistent setup across all tasks. The training arguments include `logging_strategy="epoch"` and `save_strategy="epoch"` to ensure checkpoint and performance tracking at the end of each epoch. The hyperparameters for each task are listed in Table 3.

Table 3: Hyperparameter settings for fine-tuning DistilBERT on each task. LR refers to the learning rate, BS denotes the batch size, and WD represents weight decay. These values were selected to ensure stable convergence and optimal performance across tasks.

Task	**LR**	**BS**	**WD**
Promise Identification	0.0001	8	0.01
Timing for Verification	0.00005	8	0.01
Supporting Evidence	0.00002	8	0.1
Clarity of Promise-Evidence Pair	0.0001	8	0.01