Oath Breakers at SemEval-2025 Task 06: Leveraging DeBERTa and Contrastive Learning for Promise Verification

Muhammad Khubaib Mukaddam[†*], **Owais Aijaz**[†], **Ayesha Enayat**[†]
[†]Habib University, School of Science & Engineering, Pakistan
Correspondence: mk07218@st.habib.edu.pk

Abstract

We present *Oath Breakers*, our system for SemEval-2025 Task 06: Promise Verification in ESG (Environmental, Social, and Governance) texts (Chen et al., 2025) which aims to identify and verify promises made within company reports. We fine-tune `microsoft/deberta-v3-base` with a contrastive loss to better separate promise vs. non-promise embeddings, and apply generative augmentation via `Mistral-7B-Instruct-v0.3`—manually validated—to balance the timeline classes. On the English official test set, we achieved F1=0.6003 (33% split, 3rd place) and F1=0.5733 (67% split, 2nd place), making our final ranking 2nd place on the English test dataset. These results validate the effectiveness of combined contrastive and generative strategies in promise verification.

1 Introduction

Recognizing the critical role of transparency and accountability, SemEval-2025 Task 6: PromiseEval aims to assess a company's commitment and adherence to its Environmental, Social, and Governance (ESG) promises. To this end, the organizers have compiled a diverse collection of ESG-related texts from company reports and news articles. This task aims to enhance enhance transparency and compel organizations and public figures to uphold their commitments. The insights derived would empower consumers, investors, and the broader public to make informed decisions grounded in verifiable actions and stated objectives. Ultimately, it aims to seek tangible progress on global sustainability, social justice, and ethical governance. Additional details about the task and dataset can be found at the official project page: `https://sites.google.com/view/promiseeval/promiseeval`.

*corresponding author

2 Problem Definition

The primary objective of this research is **Promise Verification**. Given a report or a part of a report from a company, the goal is to identify and verify promises made within that report. Specifically, we aim to determine whether a statement qualifies as a promise based on three key criteria:

- The statement must be related to Environmental, Social, and Governance (ESG) criteria **(required)**.
- The statement should outline a principle, commitment, or strategy that the company intends to uphold **(required)**.
- The statement should be supported by at least one piece of evidence (optional).

The Promise Verification process follows a pipeline approach, with multiple subtasks:

1. **Promise Classification**: Initially, we classify whether a statement constitutes a promise based on the criteria above.
2. **Evidence Verification**: If a promise is mentioned in the report, we need to evaluate whether it also contains evidence that supports the promise.
3. **Evidence Classification**: If evidence is mentioned for the promise, we evaluate its nature—whether the evidence is misleading, clear, or falls into another category.
4. **Timeline Verification**: If a promise is identified, we verify whether the timeline of the promise has been fulfilled or determine when it is expected to be fulfilled.

This structured approach allows for a comprehensive assessment of promises, ensuring that they are both identifiable and verifiable within the scope of ESG-related commitments.

Proceedings of the The 19th International Workshop on Semantic Evaluation (SemEval-2025), pages 1718–1723

3 Data Description

The dataset used for the Promise Verification task consists of company reports, primarily focusing on Environmental, Social, and Governance (ESG) commitments. Each entry in the dataset provides detailed information regarding specific ESG-related statements. Below is an outline of the dataset structure, including key fields and preprocessing steps undertaken.

Out of the 600 records given in the dataset, each record in the dataset includes the following fields:

- **URL**: A link to the source document, providing context and allowing for traceability.
- **page_number**: The page in the document where the statement is located.
- **data**: The textual content of the statement, which may contain potential promises.
- **promise_status**: A binary label indicating whether the statement contains a promise ("Yes") or not ("No").
- **verification_timeline**: Already, Less than 2 years, 2 to 5 years, More than 5 years, N/A
- **evidence_status**: A binary indicator of whether evidence supporting the promise is present ("Yes") or not ("No").
- **evidence_quality**: Assesses the quality of any provided evidence, categorized as "Clear," "Misleading," or "Not Clear."

This dataset provides a structured approach to assess ESG promises by capturing essential attributes related to promises, timelines, and evidence, which are critical for the Promise Verification pipeline.

4 Related Work

In recent years, several research efforts have focused on developing robust methodologies for classification and verification tasks in deep learning and natural language processing (NLP). A particularly comprehensive study by Henning et al. (2023) categorizes a wide range of techniques aimed at addressing class imbalance in NLP. Their analysis spans sampling strategies, data augmentation, staged learning, and the application of instance-level weighting, all of which are highly relevant to our work. In the context of evidence verification, we adopt oversampling methods inspired by these strategies to mitigate imbalance, especially concerning the *Misleading* class.

Another pertinent line of research by Mirzaei et al. (2023) explores the classification of implicit negative intentions in questions—an area often overlooked in mainstream NLP research. By introducing the *Question Intention Dataset*, they provide a framework for detecting both explicit and implicit negative intentions using a TF-IDF-based dictionary and Transformer models such as RoBERTa. Their emphasis on polarity classification and nuanced intention detection has informed our understanding of subtle linguistic cues, which we incorporate into the task of evidence classification within Promise Verification.

Complementary to these efforts, Heinisch et al. (2023) and Prabhu et al. (2023) demonstrate the effectiveness of contrastive learning in multilingual, multi-label framing detection tasks. Their models learn to differentiate between similar and dissimilar frame representations by employing contrastive loss functions, which draw semantically close instances together while pushing apart unrelated ones. Inspired by this technique, we integrate contrastive loss into our own model to enhance the semantic separation of misleading and accurate evidence, thereby improving verification accuracy.

Collectively, these studies lay the groundwork for our approach, which builds upon class imbalance handling, intention-aware representation, and contrastive learning. Our method synthesizes these components to address the unique challenges posed by the Promise Verification task, particularly in the classification and interpretation of evidence.

5 Methodology

The methodology section provides a detailed outlook on the progression of the task and how the baseline approach was complemented via the new methods and techniques.

5.1 Baseline

As a reference, we fine-tune `bert-base-uncased` on each subtask using only cross-entropy loss (no contrastive objective or augmentation). This establishes the baseline F1 in Table 2.

5.2 Subtask 1: Promise Classification

Initially, a `BERT`-based model was used for sequence classification, but the results were suboptimal. To improve performance, the model was upgraded to `DeBERTa`, a transformer architecture known for its superior contextual understanding and language representation. DeBERTa's robust contextual modeling capabilities outperformed BERT in handling nuanced language, making it an ideal choice for this subtask.

We fine-tune `microsoft/deberta-v3-base` with a joint classification + contrastive objective via a custom `ContrastiveTrainer`. Let $\mathcal{L}_{cls}$ be the standard cross-entropy loss on the binary labels. At each forward pass, we extract the [CLS] token embeddings

$$\mathbf{h}_i = \text{outputs.hidden\_states}[-1]_{i,0,:},$$

and build all positive pairs $(\mathbf{h}_i, \mathbf{h}_j)$ when $y_i = y_j$, and negative pairs when $y_i \neq y_j$. We then apply PyTorch's `CosineEmbeddingLoss` with margin 0.5:

$$\mathcal{L}_{contrastive} = \frac{1}{|\mathcal{P} \cup \mathcal{N}|} \sum_{(u,v)\in\mathcal{P}\cup\mathcal{N}} \ell_{\cos}(u, v,\ s_{u,v})$$

$$\text{where } s_{u,v} = \begin{cases} +1 & (u,v) \in \mathcal{P}, \\ -1 & (u,v) \in \mathcal{N}. \end{cases}$$

We weight the contrastive term by $\alpha = 0.1$:

$$\mathcal{L} = \mathcal{L}_{cls} \ + \ 0.1\, \mathcal{L}_{contrastive}.$$

Here, $\mathcal{P} = \{(u,v) \mid y_u = y_v\}$ and $\mathcal{N} = \{(u,v) \mid y_u \neq y_v\}$, with $|\mathcal{P} \cup \mathcal{N}|$ their combined count. The function $\ell_{\cos}(u, v, s)$ is implemented which for a positive pair ($s = +1$) minimizes $1 - \cos(u, v)$, and for a negative pair ($s = -1$) enforces

$$\max\big(0, \cos(u,v) - \text{margin}\big),$$

with margin $= 0.5$. Dividing by the total number of pairs balances the contributions of both positive and negative samples during training.

This formulation encourages same-label examples to cluster in embedding space and pushes apart opposite-label examples. In practice, this yields a +0.83% F1 gain over our `bert-base-uncased` baseline.

```python
def compute_loss(self, model, inputs,
    return_outputs=False,
    num_items_in_batch=None):

    labels = inputs.get("labels")
    outputs = model(**inputs,
        output_hidden_states=True)
    classification_loss = outputs.loss

    embeddings = outputs.hidden_states
        [-1][:, 0, :]
    positive_pairs, negative_pairs =
        create_pairs(embeddings, labels)
    contrastive_loss = 0

    if positive_pairs:
        pos_emb1 = torch.stack([p[0] for
            p in positive_pairs])
        pos_emb2 = torch.stack([p[1] for
            p in positive_pairs])
        cx = contrastive_loss_fn(
            pos_emb1, pos_emb2, torch.
            ones(pos_emb1.size(0)).to(
            pos_emb1.device))
        contrastive_loss += cx

    if negative_pairs:
        neg_emb1 = torch.stack([n[0] for
            n in negative_pairs])
        neg_emb2 = torch.stack([n[1] for
            n in negative_pairs])
        cp = contrastive_loss_fn(
            neg_emb1, neg_emb2, -torch.
            ones(neg_emb1.size(0)).to(
            neg_emb1.device))
        contrastive_loss += cp

    total_loss = classification_loss +
        0.1 * contrastive_loss
    return (total_loss, outputs) if
        return_outputs else total_loss
```

Listing 1: Contrastive Loss Method for Classification

5.3 Subtask 2: Evidence Verification

Given the similarity of this task to Subtask 1 in terms of task formulation, we utilized the `DeBERTa` model here as well, leveraging its contextual embedding capabilities for binary classification. We followed the same training pipeline, ensuring consistency in model optimization and hyperparameter tuning.

To preprocess the dataset, we encoded the `evidence_status` labels into numerical values, mapping `"Yes"` to `1` and `"No"` (including missing values) to `0`. The dataset had a near-balanced distribution, with **343** instances labeled as **1 (Supporting Evidence)** and **256** instances labeled as **0 (Non-Supporting Evidence)**. This balance eliminated the need for data augmentation or threshold adjustments.

Furthermore, to ensure robustness in feature

representation, we confirmed that the encoded labels were stored in `int64` format, with unique values restricted to `[0, 1]` for consistency.

This approach allowed the model to achieve strong performance on the **Evidence Verification** task, ensuring reliable classification of supporting and non-supporting evidence.

5.4 Subtask 3: Evidence Classification

Initially, we used `BERT`, but later transitioned to `DeBERTa`, which provided better results due to its disentangled attention mechanism and enhanced contextual representations. This improved our F1-scores and overall subtask accuracy, demonstrating better generalization in evidence classification.

A significant challenge in this task was **class imbalance**, particularly the underrepresentation of the `Misleading` class. To address this, we applied filtering techniques to extract instances where both `promise_status` and `evidence_status` were positive. Additionally, we utilized the `Gemini API` for data augmentation, generating synthetic samples for the `Misleading` class. This oversampling strategy increased the number of minority class samples, ensuring the model had sufficient data to learn effectively and improving its capacity to generalize across imbalanced classes.

We encoded the `clarity` labels numerically, mapping `'Clear'` to `0`, `'Not Clear'` to `1`, and `'Misleading'` to `2`, while handling missing values by assuming `'Not Clear'` as the default. This preprocessing step standardized the dataset and ensured consistency in model training.

Standard classification loss was used, along with careful validation, to ensure that the augmented data did not introduce noise or compromise the quality of predictions. The final model demonstrated improved performance in distinguishing between clear, unclear, and misleading evidence.

```
response = models.generate_content([
 f"Paraphrase the sentence: '{sentence}'"
 f"Reflect evidence status: '{evidence}',
     and quality: '{quality}'."
 f"Ensure exactly 500 characters,
     including the entity name."
 f"Do not start with a number,
     or special character." ])
```

Listing 2: Contrastive Loss Method for Classification

5.5 Subtask 4: Timeline Verification

After initial experimentation with the baseline model, we transitioned to `DeBERTa`, leveraging its superior attention mechanisms to capture subtle differences in verification timelines. Additionally, we performed preprocessing by filtering data where both `promise_status` and `evidence_status` were positive, ensuring that only relevant instances were considered. The final distribution showed dominant categories like `Already` (212 instances) and `2 to 5 years` (58 instances), ensuring a well-structured class balance.

To mitigate the class imabalances, we augmented the data to provide a more holistic data for the model to learn from. The Table 1 shows the distribution of the labels for Verification subtask. The augmentation process was carried out using the following steps:

1. Identified the class distribution and set the target count based on the majority class.
2. Determined the number of augmented samples needed per class.
3. Utilized `Mistral-7B` to generate synthetic text samples based on existing data.
4. Ensured that the generated samples maintained coherence with the dataset.
5. Incorporated the augmented samples into the training data.

We manually reviewed a random subset of 100 synthetic samples to ensure coherence and label consistency before adding them to training.

Table 1: Class Distribution: Verification Timeline

Timeline	Before	After
Already	212	212
2 to 5 years	58	212
More than 5 years	45	212
Less than 2 years	28	212

5.6 Training Arguments and Optimizations

For overall training optimization, we used a batch size of 16 (training and evaluation), with a 2e-5 learning rate under cosine decay scheduling. To maintain stability and prevent overfitting, we applied 0.01 weight decay, gradient accumulation

Table 2: F1 Score Comparison: Baseline vs Final Models

Task	Base F1	Final F1	Improv.
Subtask 1	76.67%	77.50%	+0.83%
Subtask 2	72.80%	80.00%	+7.20%
Subtask 3	59.80%	76.20%	+16.40%
Subtask 4	41.20%	72.90%	+31.70%

Table 3: Official Test Results and Leaderboard Positions (English track)

Split	F1	Rank
33% test	0.600	3rd
67% test	0.573	2nd

steps of 1, and gradient clipping with maximum norm 1.0. We enabled mixed precision (FP16) to improve memory efficiency. These optimizations ensured stable training while maximizing computational resource utilization.

6 Evaluation

We measure macro-averaged F1 on both our development split and the official test set. For all experiments, the dataset was split into **80% training, 15% validation, and 5% testing**, ensuring a robust evaluation of the models while maintaining a sufficient amount of data for generalization. The Table 2 compares the BERT baseline vs. our final DeBERTa+contrastive+augmentation system on dev. Table 3 then reports the locked leaderboard results on the English test data. Table 4 shows the task wise results on the official test dataset.

7 Analysis and Insights

- **Promise Classification:** The BERT baseline already achieved 76.67% F1, limiting room for improvement. Our final system's modest +0.83% gain indicates that promise detection primarily relies on surface cues (e.g., modal verbs, commitment phrases) which both models capture. Contrastive learning adds fine-grained separation of borderline cases, but further gains may require external world knowledge or document-level context.

- **Evidence Verification:** Here, DeBERTa's enhanced contextual embeddings, combined with contrastive loss, yield a substantial +7.20% gain. This subtask benefits from clearer signal patterns (presence/absence of explicit evidence markers), so additional representation power translates to more accurate binary judgments. Further improvements might be achievable with larger datasets and better context understanding.

Table 4: Official Subtasks F1 score on Test Leaderboard (English track)

	Promise	Evidence	Clarity	Timing
f1	0.739	0.770	0.669	0.465

- **Evidence Quality:** With a baseline of 59.80% F1, evidence-quality classification suffers from subtle semantic distinctions between "clear," "not clear," and "misleading." Data augmentation of the rare `Misleading` class closed the gap significantly, producing a +16.40% gain. This demonstrates that synthetic examples—manually validated—help the model generalize complex judgment criteria underrepresented in the original data. The use of richer annotation schemas and more robust training objectives, such as reinforcement learning with human feedback (RLHF), could further enhance performance.

- **Timeline Verification:** The largest gain (+31.70%) stems from both synthetic oversampling of underrepresented timeline categories and DeBERTa's stronger sequence modeling. Timeline inference requires understanding temporal expressions and domain-specific timeline characteristics, which benefit greatly from additional examples. Manual review of 100 augmented samples ensured no label drift occurred. Future approaches could take into account knowledge graphs to refine performance further.

- **Language Considerations:** Although we performed experiments on only the English track, we expect contrastive learning and generative augmentation to be similarly effective in other languages (French, German) provided high-quality synthetic data and language-specific pre-trained encoders. Future work should validate cross-lingual consistency and investigate whether language-specific idioms impact performance gains.

8 Discussion

- The significant improvements in **Evidence Classification**, **Evidence Quality**, and **Verification Timeline** demonstrate the value of iterative development and the inclusion of sophisticated techniques like contrastive learning, few-shot learning, and augmented datasets.

- The relatively smaller gain in **Promise Classification** may point to task saturation, where the current methods are already close to the upper performance bound for the dataset used. It could also suggest that this subtask is less sensitive to the advanced techniques applied in the final model.

- While the improvements are promising, the relatively low scores for **Evidence Quality** and **Verification Timeline** highlight areas that require further exploration, particularly in terms of data diversity, annotation quality, and advanced modeling techniques.

9 Limitations

Despite the improvements achieved, this study has several limitations. First, the dataset's size and diversity may have constrained the model's ability to generalize, particularly in subtasks such as **Evidence Quality** and **Verification Timeline**. Limited training samples and potential annotation inconsistencies could have introduced biases, affecting performance. The small size of the dataset restricted our ability to train models on a fully representative dataset that was large enough to capture all the nuances.

Second, while advanced techniques like contrastive learning and few-shot learning improved results, their effectiveness was not uniform across all subtasks. **Promise Classification** showed marginal gains, suggesting that additional refinements or task-specific adaptations might be necessary.

Finally, computational constraints limited the exploration of more complex architectures, such as large-scale transformers or graph neural networks. This hindered our experimentation and we could not try out more resource-intensive models like LLaMA.

10 Conclusion

Our participation in SemEval-2025 Task 6 showcased the effectiveness of a carefully curated methodology that integrated contrastive learning, data augmentation, and semantic-aware representations to tackle the multifaceted challenge of Promise Verification. Achieving the 2nd position overall in the leaderboard in the English dataset. The performance gains observed in the more complex subtasks affirm the importance of addressing class imbalance and leveraging semantic alignment through contrastive loss. Beyond competitive results, our model architecture and training strategy provide a strong foundation which we speculate can be used for generalizing across multilingual and multi-domain verification tasks. In summary, our system demonstrates promising strides in automated evidence verification, and we hope our insights contribute meaningfully to the broader research community working at the intersection of fact-checking, framing, and NLP-based reasoning.

References

C.-C. Chen et al. 2025. Semeval-2025 task 06: Multinational, multilingual, multi-industry promise verification. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval-2025)*, Vienna, Austria.

P. Heinisch, M. Plenz, A. Frank, and P. Cimiano. 2023. Accept at semeval-2023 task 3: An ensemble-based approach to multilingual framing detection. In *Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023)*, pages 1358–1365.

Sophie Henning, William Beluch, Alexander Fraser, and Annemarie Friedrich. 2023. A survey of methods for addressing class imbalance in deep-learning based natural language processing. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 523–540, Dubrovnik, Croatia. Association for Computational Linguistics.

M. S. Mirzaei, K. Meshgi, and S. Sekine. 2023. What is the real intention behind this question? dataset collection and intention classification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*.

P. Prabhu, S. Dammu, H. Naidu, M. Dewan, Y. Kim, T. Roosta, A. Chadha, and C. Shah. 2023. Claimver: Explainable claim-level verification and evidence attribution of text through knowledge graphs. *arXiv preprint arXiv:2403.09724*.