Winning ClimateCheck: A Multi-Stage System with BM25, BGE-Reranker Ensembles, and LLM-based Analysis for Scientific Abstract Retrieval

Wang Junjun, Chen Kunlong, Chen Zhaoqun, He Peng, Zheng Wenlu
Ant Group
xingruo.wjj@antgroup.com, cklwanfifa@gmail.com,
zhaoqun.czq@antgroup.com, penghe.hp@antgroup.com, zhengwenlu1@126.com

Abstract

The ClimateCheck shared task addresses the critical challenge of grounding social media claims about climate change in scientific literature. This paper details our winning approach for solving two subtasks. For abstract retrieval, we propose a multi-stage pipeline: (1) initial candidate generation from a corpus of ∼400,000 abstracts using BM25; (2) fine-grained reranking of these candidates using an ensemble of BGE-Reranker cross-encoder models, fine-tuned with a specialized training set incorporating both random and hard negative samples; and (3) final list selection based on an RRF-ensembled score. For the verification aspect, we leverage Gemini 2.5 Pro to classify the relationship between claims and the retrieved abstracts. Our system achieved first place in both subtasks. Part of the example code: https://github.com/cklcklcklckl/climatecheck_1st_solution.

1 Introduction

The widespread dissemination of misinformation on social media platforms represents a serious threat to informed public discourse (Wu et al., 2019), particularly regarding critical global issues such as climate change (Treen et al., 2020).

Addressing this challenge, the ClimateCheck shared task at the Fifth Workshop on Scholarly Document Processing (Abu Ahmad et al., 2025b) aims to bridge the gap between social media discourse on climate change and scientific literature. The ClimateCheck Dataset (Abu Ahmad et al., 2025a) is also provided in this shared task for model training and evaluation. Participants develop systems to: (1) Retrieve relevant scholarly abstracts for social media claims (Subtask I: Abstract Retrieval), and (2) Verify claims against these abstracts (Subtask II: Claim Verification).

2 Task Description

2.1 Subtask I: Abstract Retrieval

Given a claim c and a corpus of scientific abstracts $\mathcal{A}$ ($N \approx 400,000$), the goal is to retrieve a ranked list L_c of the top-K (here $K = 10$) most relevant abstracts. Formally, given a scoring function $s(c, a_i)$, $L_c = (a'_1, \ldots, a'_K)$ such that $a'_j \in \mathcal{A}$, $s(c, a'_j) \geq s(c, a'_{j+1})$, and L_c contains the K highest-scoring abstracts. Evaluation uses Recall@k ($k \in \{2, 5, 10\}$) and B-Pref.

2.2 Subtask II: Claim Verification

Given a claim-abstract pair (c, a), the task is to classify their relationship as 'support', 'refutes', or 'not enough information'. Evaluation uses precision, recall, and F1-score, scaled by Subtask I retrieval performance if applicable.

3 Method

Our winning solution employs a multi-stage process designed to maximize both recall and precision as illustrated in Figure 1. This approach for Subtask I involves: (1) coarse retrieval with BM25 (Robertson et al., 2009) for top 5,000 abstracts; (2) fine-grained reranking using an ensemble of fine-tuned BGE-Reranker models; and (3) final list selection using Reciprocal Rank Fusion (RRF). For Subtask II, we integrate a large language model (LLM), Gemini 2.5 Pro (Team et al., 2023), for claim-abstract relationship classification.

3.1 BM25 for Initial Retrieval

BM25 (Robertson et al., 2009) was chosen for the initial retrieval stage due to its proven effectiveness in lexical matching and its computational efficiency for large corpora.

3.1.1 Preprocessing

Our preprocessing pipeline consists of the following steps:

Proceedings of the Fifth Workshop on Scholarly Document Processing (SDP 2025), pages 276–280

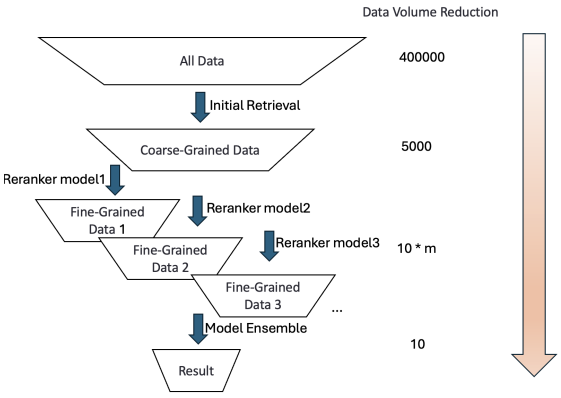

Figure 1: The Multi-Stage Text Retrieval Process with Ensemble Learning for solving subtask I. This diagram illustrates a three-stage text retrieval pipeline combining initial retrieval, multi-model reranking, and ensemble learning. Starting from 400,000 documents, the system first performs coarse-grained filtering (reducing to 5,000 candidates) through Initial Retrieval. The top results then undergo fine-grained ranking by multiple BGE-based reranker models, generating m candidate sets. Finally, an ensemble method combines outputs from all models to produce the final 10 results. The orange arrow emphasizes the progressive data volume reduction across stages

1. **Lowercasing:** All text is converted to lowercase to ensure case-insensitive matching.

2. **Punctuation Removal:** Punctuation marks are removed to focus on the textual content.

3. **Tokenization:** Text is split into individual words or tokens.

4. **Stopword Removal:** Common English stopwords (e.g., "the", "is", "in") are removed as they typically do not contribute significantly to relevance scoring (Manning, 2009). We use a standard list of English stopwords, specifically the one provided by the NLTK library.

This preprocessing pipeline was applied to all abstracts in the corpus to create a tokenized representation for BM25 indexing.

3.1.2 Retrieval Process

For each preprocessed claim, BM25 scores against all indexed abstracts are computed and sorted. The top $N_{BM25} = 5,000$ abstracts are selected as candidates. This large candidate pool size is chosen to maximize the likelihood of including true relevant documents, ensuring high recall for the subsequent reranking stage.

3.2 BGE-Reranker for Fine-Grained Ranking

For each claim, we aggregate the top 5,000 BM25-retrieved abstracts to generate a candidate pool that undergoes fine-grained reranking through our proposed method. We fine-tune multiple BGE-Reranker cross-encoders (Xiao et al., 2023), which excel at reranking through deep token-level interactions between claims and documents, yielding superior relevance judgments over bi-encoders.

3.2.1 Training Data with Hard Negatives

We trained the reranker using triplets (q, d^+, d^-) where q is a claim, d^+ a relevant abstract, and d^- an irrelevant abstract. For each positive pair:

- **Random Negatives:** Abstracts sampled uniformly from the corpus to teach broad distinctions.

- **Hard Negatives:** We generated hard negatives through a two-stage retrieval process:

  1. *BM25 Retrieval:* For each claim, we retrieved 1000 candidate abstracts using BM25.
  2. *BGE Re-ranking:* We re-ranked candidates using original BGE-Reranker.
  3. *Selection:* From the top-1000 re-ranked results, we excluded positive abstracts and selected the top-500 non-relevant abstracts as hard negative candidates.

  During training, hard negatives were randomly sampled from this candidate pool.

Combining both negative types (10 random + 5 hard negatives per positive example) created a robust training set for nuanced relevance learning.

3.2.2 Model Fine-tuning

We fine-tuned two pre-trained Transformer models, `BAAI/bge-reranker-large`[1] and `BAAI/bge-reranker-v2-m3`[2] to act as a cross-encoder. The model takes a claim and an abstract, tokenized together, as input and outputs a single relevance score. The fine-tuning process involved the following:

- **Input Representation:** Claims and abstracts were concatenated and tokenized using the model's specific tokenizer. Inputs were padded or truncated.

[1] https://huggingface.co/BAAI/bge-reranker-large
[2] https://huggingface.co/BAAI/bge-reranker-v2-m3

- **Triplet Loss Objective:** We employed a margin ranking loss. For each triplet (q, d^+, d^-), the model computes scores $s(q, d^+)$ and $s(q, d^-)$. The loss encourages $s(q, d^+) > s(q, d^-)$ plus a margin value.

We trained multiple reranker models by varying hyperparameters such as the margin value, learning rate, and batch size to encourage diversity in their predictions for later ensembling.

3.2.3 Ensemble of Reranked Lists

To leverage the strengths of different reranker models (each fine-tuned with slightly different hyperparameters or on different data shuffles, as shown in table 1), we ensembled their outputs. Our ensembling strategy, implemented in the function, is based on a variation of Reciprocal Rank Fusion (RRF) (Cormack et al., 2009). The core design principle is: *higher-ranked items should receive exponentially more weight, while maintaining balanced influence across different ranking systems*. For each claim:

1. Each of the M fine-tuned reranker models processed the top 5,000 candidates from BM25 and produced a ranked list of abstracts.

2. For each abstract a_j in the ranked list from model m_i, its rank r_{ij} was converted into a score s_{ij} using the formula: $s_{ij} = \frac{1}{k+r_{ij}}$.

3. The scores for each unique abstract a_j were then summed across all M models: $S_j = \sum_{i=1}^{M} s_{ij}$.

4. The abstracts were then re-ranked based on their aggregated scores S_j in descending order.

The final top-10 abstracts according to this ensembled ranking were selected as our submission for Subtask I. This ensemble approach helps to improve robustness and often yields better performance than any single model.

3.3 Claim-Abstract Relationship Classification with Gemini 2.5 Pro

The final component of our system, primarily designed to support downstream claim verification (akin to Subtask II), involves classifying the relationship between a given claim and each of its top-k retrieved abstracts. For this task, we leveraged the capabilities of a large language model, specifically Gemini 2.5 Pro.

Base model	batch size	margin
BGE-Reranker-large	8	0.2
BGE-Reranker-large	16	0.2
BGE-Reranker-large	8	0.25
BGE-Reranker-large	16	0.25
BGE-Reranker-v2-m3	16	0.2

Table 1: The different fine-tuned configurations used for model ensemble.

The objective was to categorize each claim-abstract pair into one of three predefined labels:

1. **Supports:** The abstract contains information that supports the assertion made in the claim.

2. **Refutes:** The abstract contains information that contradicts or refutes the assertion made in the claim.

3. **Not Enough Information (NEI):** The abstract does not provide sufficient information to either support or refute the claim.

To achieve this, we designed a specific prompt for Gemini 2.5 Pro (Team et al., 2023). The core elements of this prompt were:

- **Persona Setting:** The LLM was instructed to act as a "climate change expert."

- **Task Definition:** The model was tasked to analyze a given claim and a potentially related scientific abstract, identify relevant content within the abstract, and determine the relationship between the two.

- **Input Structure:** The prompt was designed to accept a list of claims and a corresponding list of abstracts, enabling batch processing.

- **Output Format Constraint:** A critical instruction was for the LLM to return only a numerical digit (1 for 'Not Enough Information', 2 for 'Supports', 3 for 'Refutes') for each pair, without any additional text or explanation. For batch inputs, a list of these digits was expected.

- **Output Distribution Guideline:** An explicit instruction was included to guide the model's output distribution: (Try to ensure that the proportion of '1' (Not Enough Information)

in your overall results is not less than 30%). This was intended to encourage the model to be conservative when explicit evidence was lacking, potentially mitigating over-confident "Supports" or "Refutes" classifications on ambiguous or tangentially related abstracts.

The process of generating this prompt is shown in Apprendix A.

For each claim from the dataset, its top-10 retrieved abstracts (from the ensemble reranking stage described in Section 3.2.3) were paired with the claim and fed to the Gemini 2.5 Pro model using this prompt structure. The resulting classification (Supports, Refutes, NEI) for each claim-abstract pair provides an additional layer of analysis. While our primary submission for Subtask I focused on the retrieval ranking, these LLM-derived relationship labels offer valuable insights for subsequent fact-checking efforts and could be directly utilized in a Subtask II system. The pipeline is shown in figure 2.

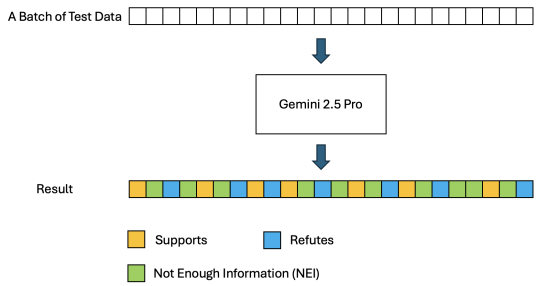

Figure 2: The pipeline of solving subtask II. Note that we input a batch of data (around 50 samples) at once in the prompt. The purpose of this approach is to ensure the model has a significantly higher probability of outputting "NEI". In the author's practice, when feeding individual data points separately, it becomes difficult to elicit "NEI" outputs from the model, with a probability of less than 1%.

4 Discussion

In this section, we discuss why we chose this cascaded architecture to solve the problem. For algorithm competitions, practicality in solving problems within limited timeframes needs consideration. Since we needed to iteratively submit results and receive feedback to refine our models, an efficient and accurate coarse-ranking solution became essential. We ultimately selected BM25 for recall because it could rapidly screen 5,000 candidate samples within a short time frame. In the subsequent second stage, we experimented with various models repeatedly and finally chose the BGE-reranker through offline evaluation. Ensemble learning effectively improved model accuracy, as demonstrated in our previous practice at KDD Cup 2024 (Chen et al., 2024). For Task II, we directly used Gemini Pro 2.5 for reasoning to obtain final results. This was because we believed that large-parameter closed-source LLM would achieve better performance in deep semantic understanding tasks compared to fine-tuned smaller-parameter models. However, resource constraints prevented us from conducting more sophisticated experiments.

Another benefit of our approach is that for Subtask I, the time required to process the data does not increase significantly when the data volume grows. This is because the first step, BM25, will only retain a fixed set of 5,000 entries. Naturally, for Subtask II, the processing time will increase linearly with the amount of data.

5 Conclusion

This paper describes our ClimateCheck shared task system, which won first place in both Subtasks. Our approach combines a BM25 sparse retriever for candidate pooling with an ensemble of fine-tuned BGE-Reranker models for semantic reranking. Training the rerankers on a dataset incorporating hard negative mining significantly improved their performance. For claim verification, Gemini 2.5 Pro effectively classified claim-abstract relationships. This hybrid pipeline demonstrates the efficacy of combining optimized retrieval with large language models.

Limitations

One practical drawback of our solution is that during the re-ranking phase for subtask I, we fine-tuned several different models using various parameters and combined them via ensemble learning. Although this is a common technique to boost rankings in data science competitions, it often proves impractical in real-world industrial applications due to its excessive complexity. Additionally, despite significant efforts, we failed to identify a practical method for effectively utilizing large language models in subtask I, even though we firmly believe LLM would raise the performance ceiling for this subtask. We believe that the recently studied large language model-based text retrieval methods (such

as qwen3-embedding [3]) can likely improve performance in subtask I without the need for ensemble learning. We will conduct further exploratory work on this subsequently.

References

Raia Abu Ahmad, Aida Usmanova, and Georg Rehm. 2025a. The ClimateCheck dataset: Mapping social media claims about climate change to corresponding scholarly articles. In *Proceedings of the 5th Workshop on Scholarly Document Processing (SDP)*, Vienna, Austria.

Raia Abu Ahmad, Aida Usmanova, and Georg Rehm. 2025b. The ClimateCheck shared task: Scientific fact-checking of social media claims about climate change. In *Proceedings of the 5th Workshop on Scholarly Document Processing (SDP)*, Vienna, Austria.

Kunlong Chen, Junjun Wang, Zhaoqun Chen, Kunjin Chen, and Yitian Chen. 2024. Llm-powered ensemble learning for paper source tracing: A gpu-free approach. *arXiv preprint arXiv:2409.09383*.

Gordon V Cormack, Charles LA Clarke, and Stefan Buettcher. 2009. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 758–759.

Christopher D Manning. 2009. *An introduction to information retrieval*.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Kathie M d'I Treen, Hywel TP Williams, and Saffron J O'Neill. 2020. Online misinformation about climate change. *Wiley Interdisciplinary Reviews: Climate Change*, 11(5):e665.

Liang Wu, Fred Morstatter, Kathleen M Carley, and Huan Liu. 2019. Misinformation in social media: definition, manipulation, and detection. *ACM SIGKDD explorations newsletter*, 21(2):80–90.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding.

[3] https://github.com/QwenLM/Qwen3-Embedding

A Prompt Implementation for Gemini 2.5 pro

In this section, we introduce how to generate prompts mentioned in section 3.3. Note that this prompt is written in Chinese in the competition.

```
def get_prompt(claims, abstracts):
    return f"""You are an expert in
    climate change. I will provide you
    with a claim and an abstract.
                A claim typically
    represents an assertion about
    climate change, while an abstract is
     a paper abstract potentially
    related to that claim.
                Your task is to identify
     content in the abstract relevant to
     the claim and analyze the
    relationship between the abstract
    and the claim.
                There are three possible
     relationship categories:
                1. 'Not Enough
    Information': Indicates the abstract
     neither supports nor refutes the
    claim;
                2. 'Supports': Indicates
     the abstract provides evidence
    supporting the claim;
                3. 'Refutes': Indicates
    the abstract provides evidence
    refuting the claim.

                Your response must
    consist ONLY of a single number
    between 1 and 3, representing the
    relationship category. Return ONLY
    the number, without any additional
    text.
                I will provide N claims
    and N abstracts simultaneously. You
    should return a list of N numbers.
                Important Notes:
                1. Ensure the returned
    numbers are strictly between 1 and
    3.
                2. Aim for the overall
    proportion of '1' (Not Enough
    Information) responses in your
    results to be at least 30%.

                Claims list: {claims}
                Abstracts list: {
    abstracts}
                """
```

Listing 1: Prompt generation function