NLP-Based Climate Claim Verification - Group 40

Abstract

Climate change has become one of the most important global challenges. However, arguments around its severity remain, particularly in the era of misinformation. It is critical to develop systems that can assess the validity of climate-change-related claims. This paper presents a fact-checking method to connect claims with supporting evidence to verify their validity. We implement a two-stage system that combines a sentence transformer model for evidence retrieval and a transformer-based classification model to classifying the relationship between claims and evidence into four categories: *SUPPORTS*, *REFUTES*, *NOT_ENOUGH_INFO*, or *DISPUTED*. Our system was evaluated against a baseline BRNN model. The result shows a 45.7% increase in classification accuracy as the system achieves a final F-score of 0.0387 and a classification accuracy of 43.51%, with both training and inference completed within 30 minutes. This demonstrates the system's scalability to handle larger content efficiently while maintaining reasonable performance.

1 Introduction

Nowadays, most scientists and much of the public agree that climate change contributes to environmental challenges, such as extreme weather, resource scarcity, and others. While many view it as an urgent threat, skeptics question its severity and existence. As a result, conflicting statements appear in academic papers and across the web, fueling debates. With the rise of the internet and generative AI, people can easily access this content, while there is no perfect method to verify the truth of claims, especially in controversial areas like climate change. This highlights the importance of systems that connect evidence to claims and assess their validity.

Wang (2023) provided a review of existing climate-related fact-checking systems. The systems follow a multi-stage pipeline involving claim detection, evidence retrieval from trusted corpora, and claim verification using fine-tuned large language models (LLMs). Performance continues to improve, but challenges remain. This was caused by the emerging topics and nuanced language.

Similarly, Rojas et al. (2024) proposed a two-stage hierarchical model to detect and categorise climate change misinformation on Twitter. The first part implemented a transformer-based binary classifier to distinguish between *convinced* and *contrarian* content. The second stage applied a taxonomy classifier to further categorise contrarian posts into a misinformation category. Their system achieved an F1-score of 81.1 for binary classification and 53.6 for taxonomy classification.

We implement a fact-checking system that integrates a sentence transformer and cosine similarity to retrieve relevant and reliable evidence for a given claim. A transformer-based classification model then predicts the relationship between the claim and evidence into one of four categories: *SUPPORTS*, *REFUTES*, *NOT_ENOUGH_INFO*, or *DISPUTED*, thereby validating or rejecting the claim. The system automates the process, reducing reliance on manual analysis while improving scalability and accuracy.

2 Approach

2.1 Data Processing

2.1.1 Loading JSON Files

Three JSON files are loaded: `train-claims.json` and `dev-claims.json`, which contain claims information for training and development, and `evidence.json`, which contains supporting evidence texts. These files are provided in raw data structures for further processing.

2.1.2 Exploratory Data Analysis

Our model is trained on the `train-claims.json` dataset, which comprises 1,228 climate-related

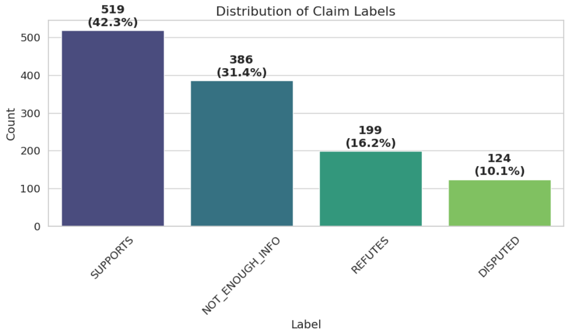

Figure 1: Distribution of Claim Labels.

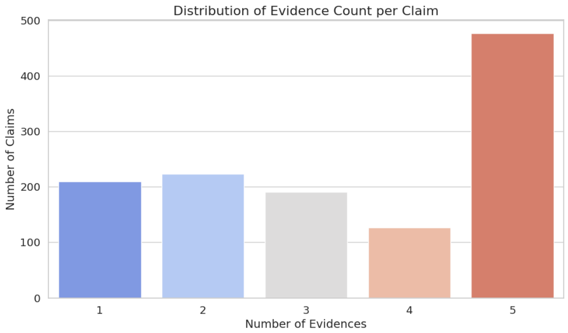

Figure 2: Distribution of Evidence Count per Claim.

claims, and the corresponding `evidence.json` file containing 1,208,827 textual statements, each composed of several sentences. Figure 1 and 2 illustrate key statistics of the training dataset. Additionally, we filtered out duplicate evidence entries, reducing the total number of evidence statements from 1,208,827 to 1,193,821.

2.1.3 Text Pre-processing

This function standardises the text data following the steps suggested by Anandarajan et al. (2018). First, it converts all text to lowercase, removes punctuation and numbers using regular expressions, and splits the text into individual tokens. Next, English stopwords from the NLTK corpus are removed. Lemmatisation is then applied to transform words into their base forms, ensuring that linguistic variations of the same word are treated uniformly. Finally, all tokens are rejoined into a sentence in their original order.

2.2 Word Embedding

We deploy `all-MiniLM-L6-v2` from the `sentence-transformers` library to convert texts into word embeddings. According to Reimers and Gurevych (2019), Sentence-BERT-style models generate semantically meaningful sentence embeddings and outperform other state-of-the-art sentence embedding methods. Among those models, `all-MiniLM-L6-v2` is known for its lightweight, making it suitable for our task, which involves processing over 1.1 million evidence.

The data set for fine-tuning is constructed by pairing each claim with its associated evidence passage and label, resulting in 4,122 (claim, evidence, label) tuples. Each claim and evidence pair is tokenised jointly using a pre-trained tokeniser and separated by a `[SEP]` token. We apply truncation to ensure the combined sequence does not exceed 256 tokens and use padding to maintain uniform input length across all examples. The tokeniser outputs both `input_ids` and `attention_mask`, which indicate actual tokens versus padding.

We fine-tune the pre-trained encoder model alongside a newly added classification head composed of two fully connected layers with a ReLU activation. The `[CLS]` token embedding from the last hidden state is used as the input to the classification model.

The objective is to predict the correct label for each (claim, evidence) pair by minimising cross-entropy loss. The model is optimised using the AdamW optimiser (Loshchilov and Hutter, 2019) with a learning rate of 2e-5 and trained using a batch size of 32 for 10 epochs.

After fine-tuning, claim and evidence embeddings are extracted by passing each text through the encoder. Embeddings are obtained from the hidden state of the `[CLS]` token in the final transformer layer, designed to represent the whole sentence. These embeddings are then used in the similarity-based evidence retrieval process.

2.3 Evidence Retrieval

Given a claim, our goal is to identify the pieces of evidence with the highest semantic similarity, which we measure using cosine similarity. Cosine similarity compares the angle between two normalised vectors and is computed as the inner product of unit vectors:

$$cosine\_similarity(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u} \cdot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|} \quad (1)$$

We utilise FAISS (Facebook AI Similarity Search) (Douze et al., 2024), a library optimised for efficient similarity search over large-scale data, to perform evidence retrieval. All embeddings are normalised to unit length before a FAISS index is created using the IndexFlatIP structure, which enables inner product search that approximates cosine similarity due to vector normalisation.

For each claim, the system retrieves the top 10 most similar evidence. A heuristic filter is applied to retain only the evidence whose similarity score differs by no more than 0.05 from the previously selected one. The selection process may select more than five evidence for each claim, sacrificing the recall but boosting precision and overall F-score after comparing with just selecting five evidence for each claim.

We compute the F1-score for each claim by comparing the retrieved evidence set with its related ground-truth evidence. The overall performance score is calculated by averaging these F1-scores across the entire dataset.

2.4 Classification

The goal of this part is to categorise each (claim, evidence) pair into one of four categories: SUPPORTS, REFUTES, NOT_ENOUGH_INFO, or DISPUTED. The input is a concatenated text of the claim and evidence, separated by the [SEP] token. This format allows the model to treat the task as a sentence-pair classification problem, which aligns with BERT standard practice.

We build on the pre-trained bert-base-uncased model, which is fine-tuned by appending a classification head. The head consists of a fully connected layer followed by a ReLU activation, and a final softmax layer to output the probability distribution across the four classes. The input to the classification model is the final-layer embedding of the special [CLS] token. The output is computed by applying a softmax function over a linear transformation of the [CLS] token representation. For training, we use the AdamW optimiser with a learning rate of 2e-5, a batch size of 16, and train the model for 10 epochs. Tokenisation applies truncation at 256 tokens to manage input length.

To ensure the model learns effectively across all four categories, we applied label balancing during training since we observed label distribution imbalance during exploratory data analysis (refer to Figure 1). This was done by downsampling the larger classes to match the number of samples in the smallest class. This step was necessary to avoid the model being biased toward more frequent labels and underperforming on less frequent ones. Our classification model builds on the output of the evidence retrieval system, taking the retrieved top evidence as input along with the claim.

2.5 Baseline: BRNN

We built two Bidirectional Recurrent Neural Networks (BRNNs) from scratch as our baseline. Implementation was done using bidirectional GRU units with attention over word embeddings. The first BRNN is used to retrieve the relevant evidence for given claims. It used a bidirectional GRU unit to understand the full context of the input evidence and claim sequence from both forward and backwards. The second BRNN is used to classify the label for each given claim and evidence pair. It creates two independent encodings to process both claim and evidence texts, each using a bidirectional GRU unit, then combines them for a final classification decision. Then we combine the two BRNNs by adjusting their input and output format, so the second BRNN can use the output result of the first BRNN to perform its classification.

3 Experiments

3.1 Evaluation Methodology

We take a multi-faceted approach to evaluate our fact-checking system. Performance is first measured using the **Evidence Retrieval F-score**, which assesses the relevance and completeness of the supporting evidence retrieved for each claim. Next, we use **Validation Accuracy** to evaluate the correctness of claim classification across the categories.

To holistically assess system performance, we compute the **Harmonic Mean** of the F-score and accuracy, which balances the trade-offs between retrieval and classification performance.

Additionally, we generate a classification report that includes **precision**, **recall**, and **F1-scores** for each class. This comprehensive methodology ensures that our fact-checking system not only retrieves relevant evidence but also accurately categorises claims based on that evidence.

3.2 Experiment 1: Fine-tuning the Epochs

We fine-tuned the sentence transformer using raw text with a learning rate of 2e-5 across varying numbers of training epochs, ranging from 3 to 15, and observed the resulting performance.

3.2.1 Experiment 1 Result

From Table 1, we found that higher epochs do not always result in higher F-score. This may be caused

Epoch	F. Loss	F-score	Acc.	H. Mean
3	0.2885	0.01607	0.35714	0.03076
5	0.6236	0.03589	0.34415	0.06500
10	0.1252	0.00754	0.40909	0.01481
15	0.0632	0.01052	0.35064	0.02044

Table 1: Results of different fine-tuning epochs for the sentence transformer.

by the incompatibility between the fine-tuning approach and the sentence transformer. This condition could potentially cause the model to diverge during training. Another possibility is overfitting since the fine-tuning dataset is relatively small with only 4,122 instances. Further experimentation is necessary to confirm this theory. Interestingly, the classification model performance does not improve linearly with the F-scores. This could be due to low F-scores differences, which might not impact classification accuracy significantly.

3.3 Experiment 2: Pre-processing Text

For LLMs such as Transformers, text pre-processing is often considered unnecessary, as they are trained on raw text. However, in this experiment, we applied the previously described text pre-processing steps to claims and evidence before generating embeddings and classifying. The aim was to evaluate whether simplifying and cleaning the input could lead to improved performance. We conducted three experimental runs, fine-tuning the embedding model using a learning rate of 2e-5 and training for five epochs, and the classification model using a learning rate of 2e-5 and training for 10 epochs, comparing results between the pre-processed and original text inputs.

3.3.1 Experiment 2 Result

Run	Preproc.	F-score	Accuracy	H. Mean
1	no	0.03138	0.27272	0.05629
2	no	0.04720	0.35064	0.08320
3	no	0.04995	0.29870	0.08559
	Average	**0.04284**	**0.30735**	**0.07503**
Run	**Preproc.**	**F-score**	**Accuracy**	**H. Mean**
1	yes	0.02344	0.45454	0.04459
2	yes	0.01560	0.40909	0.03005
3	yes	0.03083	0.35064	0.05668
	Average	**0.02329**	**0.40476**	**0.04377**

Table 2: Evidence retrieval and classification results on raw or pre-processed text

From Table 2, we found that text pre-processing improved the classification model's performance but did not improve the evidence retrieval model. However, the difference in F-score was minor (around 0.02). One possible hypothesis is that numeric information that was removed during pre-processing may be valuable for the evidence retrieval model. In contrast, the classification model performed lower on raw text, which was the opposite of our expectations.

4 Result: Baseline vs. Final Model

Metric	Baseline	F. Model
Evidence Retrieval F-score (F)	0.0000	0.0387
Claim Classification Acc. (A)	0.2987	0.4351
Harmonic Mean of F and A	0.0000	0.0711

Table 3: Comparison of evaluation metrics between baseline and our model on the validation set.

The claim-evidence retrieval model achieved an F-score of 0.0387, while the claim classification model reached an accuracy of 43.51%. The harmonic mean of these two metrics is 0.0711. These results significantly outperform the BRNN baseline, with a 45.7% improvement in classification accuracy.

Label	Precision	Recall	F-score	Support
SUPPORTS	0.51	0.71	0.59	68
REFUTES	0.32	0.56	0.41	27
NOT_ENOUGH_INFO	0.80	0.10	0.17	41
DISPUTED	0.00	0.00	0.00	18

Table 4: Classification report for four labels.

As shown in Table 3, the baseline model is unable to retrieve any correct evidence for a given claim, leading to an F-score of 0. The claim classification accuracy is 29.87%. This result is close to the expected performance of random guessing among four labels (i.e., 25%). The performance might be due to the model's simple architecture. It might struggle to convert textual inputs into meaningful embeddings and fail to capture the semantic relationships between claims and evidence.

In contrast, as seen in Table 3, our system achieves an evidence retrieval F-score of 0.0387, a classification accuracy of 43.51%, and a harmonic mean of F-score and accuracy of 0.0711. These outcomes represent a huge improvement over the baseline, though there is still room for further enhancement. The improved evidence retrieval performance can be attributed to our use of a pre-trained sentence transformer, which has been trained on a large corpus and is better at capturing semantic meaning compared to the baseline model.

It is worth noting that our system is not limited to retrieving exactly five evidence sentences per claim. Allowing more than five retrieved evidence may sacrifice recall but boost precision, which in turn results in a higher F-score compared to the strict five-evidence retrieval setup we tried previously. In addition, the approximately 13% increase in classification accuracy compared to the baseline may be an outcome of retrieving more semantically relevant evidence. Even when the retrieved evidence is not entirely correct, it often bears some semantic similarity to the claim, thereby still contributing to improved classification performance.

A closer observation of the classification model's performance through the report shows that categories with more instances are likely to achieve higher accuracy. However, the recall scores indicate that the model struggles to reliably distinguish `NOT_ENOUGH_INFO`, although it is the second largest class. Among all four labels, `DISPUTED`, which has the fewest instances, shows the lowest precision and recall. On the other hand, the class with the highest number of instances (`SUPPORTS`) achieves the highest and most consistent results across all metrics. This finding suggests instability in the model's predictions, and data imbalance may influence its performance.

5 Conclusion

This paper focuses on building a scalable fact-checking system to assess climate-related claims through modern NLP methodologies. Our approach integrated a sentence transformer for generating semantic embeddings, FAISS for efficiently searching related evidence given a claim based on cosine similarity. We also implement a BERT-based classification model for predicting the relationship between claims and retrieved evidence. The system reached a classification accuracy of 43.51%, an evidence retrieval F-score of 0.0387, and a harmonic mean of 0.0711.

Our findings indicate that text pre-processing prior to model input is not a decisive factor for performance since the improvement or decline are minor. Similarly, a higher F-score does not always produce higher accuracy. Additionally, we observe that using fewer training epochs tends to yield better performance for the evidence retrieval model.

One limitation in our research is the minor difference in F-score across different configurations, making it difficult to determine whether the classification model genuinely benefits from retrieving more accurate evidence. In other words, it remains unclear whether the classifier is meaningfully learning from the fine-tuning dataset. Furthermore, our reliance on FAISS for efficient retrieval over 1.1 million evidence may introduce trade-offs, as it prioritises speed at the potential cost of retrieval accuracy.

Future work may explore different fine-tuning strategies that align with our objectives. It could adopt an alternative learning approach using loss functions such as `MultipleNegativesRankingLoss` to model semantic similarity rather than framing evidence retrieval as a classification problem. Instead of using cosine similarity, we could implement a more distinct similarity metric. It may capture the entailment, contradiction, or neutrality relationships between claims and evidence better. Finally, data augmentation can be applied to balance class distributions and enhance the robustness and generalisability of the system by enabling training on a larger and even dataset.

References

Murugan Anandarajan, Chelsey Hill, and Thomas Nolan. 2018. *Practical Text Analytics: Maximizing the Value of Text Data*. Springer International Publishing, Switzerland.

Matthijs Douze, Jeff Johnson, and Hervé Jégou. 2024. Faiss: A library for efficient similarity search.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*.

Cristian Rojas, Frank Algra-Maschio, Mark Andrejevic, Travis Coan, John Cook, and Yuan-Fang Li. 2024. Augmented cards: A machine learning approach to identifying triggers of climate change misinformation on twitter. *Preprint*, arXiv:2404.15673.

Wang. 2023. *Artificial Intelligence in Science: Challenges, Opportunities and the Future of Research*. OECD Publishing, Paris. Accessed: 2025-05-16.