Evidence-Based Climate Science: Developing an Automated Fact-Checking System.

Tue9AM_Group3
Claudia Caro. ID: 1298396
Andres Gutierrez. ID:1405461

Abstract

This study presents the development of an automated fact-checking system specifically designed to address misinformation in climate science. The system utilises natural language processing (NLP) techniques in two main phases: information retrieval and multiclass classification. In the retrieval phase, models such as TF-IDF, SBERT, and Doc2Vec combined with BM25 are employed to identify semantically relevant evidence from a corpus. The classification phase involves using a Support Vector Machine (SVM) as a base classifier to categorise claims into four categories. Additionally, the representation of the claim combined with their respective evidence using the Doc2Vec model was used to enhance performance. Despite the promising framework, the system's performance was constrained by several limitations, including small dataset size, model overfitting, and lack of data diversity, which hindered its generalisation and robustness.

1 Introduction

Climate change poses a challenging threat to humanity, and its consequences have raised widespread concern among scientists, policymakers, and the general public. Effective communication of climate science is crucial for informed decision-making and policy development. However, the spread of misinformation and unverified claims has distorted public opinion, making it challenging to distinguish fact from fiction.

Although existing approaches to combating misinformation in climate science, such as manual fact-checking by experts and public awareness campaigns, can provide the ground truth on the credibility of a topic, and there are plenty of researches addressing this approach (Zhou and Zafarani, 2018), they have limited impact in the face of the rapid proliferation of fake news. In contrast, cutting-edge NLP techniques for automatic Fact-Checking have demonstrated promising results in countering the spread of false claims, particularly in the digital age where information can go viral within minutes (Guo et al., 2022). This study aims to build on these successes by developing an NLP-based approach from scratch, without relying on pre-trained classification algorithms, to effectively address the challenges of climate science misinformation.

To address this challenge, we propose developing an automated fact-checking system from scratch designed to verify claims related to climate science. Our system aims to enhance the credibility of climate science by providing a user-friendly method for evaluating the veracity of climate-related claims. The system will operate in two stages: retrieval and classification. In the retrieval stage, the system will search a large corpus of evidence passages to find the most relevant information about a given claim. In the classification stage, the system will evaluate the retrieved evidence and categorise the claim into one of four categories.

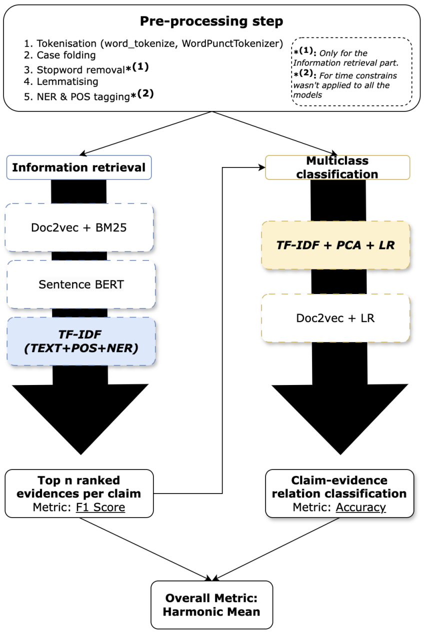
2 Approach selection

We employed a divide-and-conquer approach to achieve the project's primary objective - searching and retrieving a list of the most relevant evidence passages for a given claim and classifying the claim based on its most probable relation with the evidence. This approach is illustrated below:

By breaking down the task into smaller subtasks, we were able to tackle each component separately, developing targeted solutions for:

- Information retrieval (Passage ranking): retrieving the most relevant evidence passages for a given claim.
- Multi-class classification: determining the most probable relation between the claim and the evidence.

Figure 1: Data pipeline


Pre-processing step
1. Tokenisation (word_tokenize, WordPunctTokenizer)
2. Case folding
3. Stopword removal*(1)
4. Lemmatising
5. NER & POS tagging*(2)
*(1): Only for the Information retrieval part.
*(2): For time constrains wasn't applied to all the models
Information retrieval
Doc2vec + BM25
Sentence BERT
TF-IDF (TEXT+POS+NER)
Multiclass classification
TF-IDF + PCA + LR
Doc2vec + LR
Top n ranked evidences per claim
Metric: F1 Score
Claim-evidence relation classification
Metric: Accuracy
Overall Metric: Harmonic Mean

For the information retrieval component, we focused on measuring the similarity between natural language sentences through sentence similarity tasks, measuring the similarity between all claim-evidence pairs, ranking them, and retrieving the top *n* most relevant evidence. We explored three distinct approaches to calculate the similarity between claim and evidence texts. At the same time, for the multiclass classification task, we aimed to identify the most likely relationship between classified evidence texts and a given claim, proposing three models designed to maximise accuracy. Our ultimate goal is to achieve a high correlation between F1 score and accuracy, thereby boosting the harmonic mean of these metrics.

3 Preprocessing step

When preprocessing the data, we are interested in transforming the inputted texts into numerical matrices that can be passed through a machine and are the most accurate possible representations of the original text. With this in mind, we decided to apply a set of steps to the data, which are:

1. *Tokenisation*: The intention behind tokenising the texts was to split them into a notation allowing us to extract as much context as possible. For this project, we explored different tokenisation approaches and ultimately opted for word and word-punctuation tokenisers, depending on the specific model. These two approaches yielded the highest classification metrics among the methods we tested.

2. *Case folding*: As for the case folding scenario, we directly applied lowercase to all the imputed texts.

3. *Stopword removal*: We applied stop-word removal to some models created for this project. Specifically, we utilised the NLTK stopwords module, which comprises a curated list of common English words, to eliminate these words from our text data.

4. *Lemmatising*: Regarding the lemmatiser, we only employed it in the less complex models to boost processing speed. In this task, we found that the WordNetLemmatizer was the optimal choice for reducing tokens to their common root form, enhancing performance.

5. *NER & POS tagging*: As part of our data augmentation strategy, we leveraged the spaCy library to perform Name Entity Recognition (NER) and Part of Speech (POS) tagging on the texts, aiming to enrich the contextual representation of each sentence. However, due to the time-intensive nature of this process, we only applied it to a select subset of models, as processing all evidence and claim texts for all models through this tagging would have been prohibitively time-consuming.

4 Information Retrieval

4.1 Doc2Vec + BM25

The Paragraph Vector (Le and Mikolov, 2014) is a method for sentence similarity tasks due to its ability to capture the semantic meaning of the text in a multidimensional vector space. The implementation of the Paragraph Vector is called Doc2Vec. This method learns to represent all documents as dense vectors, encoding both the words used and the order in which they appear. This enables the model to understand context and subtle differences in meaning, which could be considered crucial for accurately determining sentence similarity.

On the other hand, BM25, short for Best Matching 25, is a ranking function used in information retrieval systems to rank documents based on their relevance to a given query. It is an extension of the probabilistic information retrieval model. BM25 calculates a document's relevance score by considering the frequency of query terms in the document (term frequency), the importance of these terms within the entire collection of documents (inverse document frequency), and the length of the document (Robertson and Zaragoza, 2009).

In the context of an automated fact-checking system, the primary goal is to retrieve relevant evidence that supports or refutes a given claim. The first attempt consisted of combining Doc2Vec and BM25, which allowed leveraging the strengths of both models to enhance the accuracy and robustness of the information retrieval process. Doc2Vec captures the semantic meaning of texts by generating dense vector representations, which helps in understanding the contextual relationship between claims and evidence. This semantic understanding is crucial for identifying relevant evidence that may not contain exact keyword matches but is contextually related to the claim (Le and Mikolov, 2014).On the other hand, BM25 handles the retrieval of evidence by ranking documents that contain relevant terms related to the claim higher. This model tried to handle keyword-based retrieval, ensuring that the most pertinent pieces of evidence containing relevant terms related to the claim are identified, prioritised, and given higher relevance scores (Robertson and Zaragoza, 2009).

The combined use of Doc2Vec and BM25 begins with the preprocessing and tokenising of the dataset, which consists of claims and their corresponding evidence. Doc2Vec is trained on these pairs to learn vector representations that capture the semantic meaning of the texts. Once trained, the model generates vectors for both the claims and the evidence, facilitating the measurement of their semantic similarity. Concurrently, BM25 is applied to score the relevance of evidence documents to each claim based on term frequency and document length normalisation.

The BM25 normalisation process adjusts for the length of documents to prevent longer documents from being unfairly penalised or favoured, ensuring that the relevance scores are more balanced and accurate (Bashir and Khattak, 2014). By integrating the semantic similarity scores from Doc2Vec with the relevance scores from BM25, the system could more accurately identify and rank the most pertinent pieces of evidence. This hybrid approach seeks to include contextually similar and keyword-relevant documents, resulting in a comprehensive information retrieval process for automated fact-checking.

4.2 SBERT base

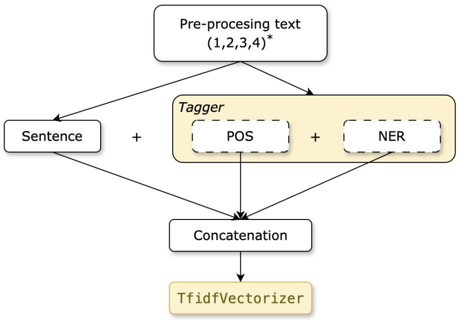
Sentence BERT (Reimers and Gurevych, 2019) is a specialised encoder architecture model designed for semantic textual similarity (STS) and transfer learning tasks. Sentence BERT builds upon the Bidirectional Encoder Representations from Transformers (BERT) model (Devlin et al., 2019). It is a powerful encoder that extracts contextualised representations from text sequences by leveraging the attention mechanisms inherent in transformer architectures (Vaswani et al., 2017). The Sentence-BERT (SBERT) architecture is built upon a Siamese network, which consists of two identical branches with shared weights, allowing for the simultaneous processing of two input sentences. The output from each stream is then fed into a pooling layer, flattening the dimensions into fixed-length vectors. Depending on the specific task, these vectors are subsequently passed through an optimisation function. For Semantic Textual Similarity (STS) tasks, a distance measure is applied to obtain a similarity score. Specifically, we employed Cosine Similarity, yielding values between -1 and 1, which were then trained against the ground truth labels for each claim-evidence pair. To ensure a balanced training set, we generated synthetic claim-evidence relations, maintaining an equal number of positive and negative pair relationships. However, this approach had an unforeseen consequence: the training set became less representative of the test set, which consisted of 153 claims paired with over 1,200,000 evidence texts. This significant mismatch in scale led to inaccuracies in the model's performance. Although we recognised this limitation, computational constraints prevented us from expanding the training set to match the test set's scale, ultimately hindering the model's performance.

4.3 TF-IDF base

Due to the limitations and challenges we encountered with Neural Network models, we opted for a simpler approach as our third solution for the Sentence Similarity task.

As shown in the image, we applied a comprehensive preprocessing pipeline to each claim-evidence pair separately. This involved refining the tokens in steps 1-4 and augmenting the data to gain more context. Specifically, we extracted Name Entity Recognition (NER) and Part-of-Speech (POS) vectors for each text and then concatenated and trans-

Figure 2: TF-IDF Encoder - Architecture



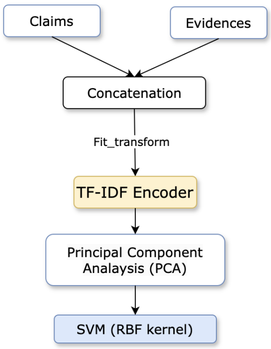
formed the three texts into a term-frequency inverse document-frequency (TF-IDF) representation. This step highlighted the least common relationships and extracted patterns that made each pair relation closest. Finally, we applied a distance measure to determine similarity, with the cosine similarity measure yielding the most accurate results for this model.

5 Multiclass classification

5.1 TF-IDF + PCA + SVM classifier

For the Support Vector Machine (SVM) classifier, we adopted the same extensive preprocessing approach employed for the previous model prior to feeding the data into the model. The following steps outline the comprehensive training pipeline we applied to the data: As illustrated in the diagram, the key distinction in the pre-processing step lies in the concatenation of data before encoding, which preserves the relationship between each claim and its corresponding evidence. The subsequent steps involve:

Figure 3: Training pipeline - SVM model



1. Principal Component Analysis (PCA) dimensionality reduction applied to the data to reduce the sparse matrix dimensions while retaining maximum variance. We opted for a 2-component model, as the differences were negligible with additional components.

2. A Support Vector Machine (SVM) model with dual optimisation function and RBF kernel was applied to the resulting components to predict the ground truth labels for the training set.

The results of the SVM model are discussed in the Experiments section.

5.2 Doc2Vec + LR Classifier

The automated fact-checking system's classification task involves categorising the claims' status given the evidence into predefined labels such as SUPPORTS, REFUTES, NOT_ENOUGH_INFO and DISPUTED. To achieve this, we utilise the Doc2Vec model for generating dense vector representations of the claims and evidence, followed by a Logistic Regression classifier to predict the labels based on these vectors. As mentioned, Doc2Vec captures the semantic meaning, encapsulating the contextual information necessary for adequate classification. This transformation allows the model to understand the nuances in the claims and their corresponding evidence, enabling the classifier to make informed decisions. The process begins with preprocessing and tokenising the dataset, which pairs each claim with its corresponding evidence. The trained Doc2Vec model generates a dense vector for each claim and its evidence, which is input to the classification model. After obtaining the vector representations, a Logistic Regression classifier is trained on these vectors to learn the mapping from text vectors to claim labels. Logistic Regression is chosen for its simplicity and effectiveness in multiclass classification tasks. It models the probability that a given input vector belongs to a particular class, making it the more straightforward choice for the classification problem.

6 Experiments

6.1 Information Retrieval

In the experiments conducted for the information retrieval component, we explored various parameter configurations for each of the developed models. The results of these experiments are presented in the following table, which provides a comprehensive overview of the performance of each model under different settings: As shown in the table,

Model	Train F1	Test F1 (Codalab)
Doc2Vec base	0.001	-
Doc2Vec best	0.003	-
SBERT base	0.004	0.001
SBERT best	0.011	0.007
TF-IDF base	0.062	0.052
TF-IDF best	0.076	0.060

Table 1: Information Retrieval models results

the Deep Neural Network (DNN) models underperform in measuring similarity, contrary to expectations. This shortcoming is attributed to the training limitations discussed in the respective sections, which resulted in the models being inaccurate when predicting on new, unbalanced data. On the other hand, the top-performing models are parametric variations of the TF-IDF base model, which achieves enhanced performance by incorporating Name Entity Recognition (NER) and Part-of-Speech (POS) taggers, resulting in a significant increase in the F1 score.

6.2 Multiclass classification

To evaluate the effectiveness of our approach, we conducted several experiments using Principal Component Analysis (PCA) for dimensionality reduction and Support Vector Machine (SVM) for classification as a baseline model and various configurations of the Doc2Vec model combined with Logistic Regression (Doc2Vec + LR). We experimented with different vector sizes and training epochs to assess the impact on classification performance.

For each configuration, we trained the Doc2Vec model on the claims and evidence dataset and then used the generated vectors as input features for the Logistic Regression classifier.

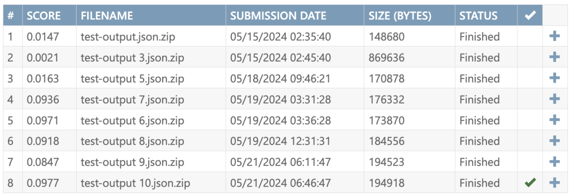
Model	Train Accuracy	Dev Accuracy
TF-IDF PCA SVM	0.4351	0.3972
Doc2Vec v50 e50	0.7565	0.3961
Doc2Vec v50 e100	0.7638	0.4545
Doc2Vec v100 e50	1.0	0.3376

Table 2: Multiclass Classification models results

The results indicate that incorporating Doc2Vec significantly improves the classification model's performance compared to the baseline Logistic Regression model. The best performance was achieved with Doc2Vec + LR using 50 as vector size and 100 epochs, with an accuracy of 0.39. This configuration provided the best balance between capturing semantic meaning and effective classification. These results highlight the effectiveness of combining Doc2Vec with Logistic Regression for classifying the status of claims given the evidence in automated fact-checking systems. The improvements in accuracy demonstrate the value of using semantic vector representations to enhance the classification process.

7 Results

Figure 4: Codalab final results



#	SCORE	FILENAME	SUBMISSION DATE	SIZE (BYTES)	STATUS	✓	
1	0.0147	test-output.json.zip	05/15/2024 02:35:40	148680	Finished		+
2	0.0021	test-output 3.json.zip	05/15/2024 02:45:40	869636	Finished		+
3	0.0163	test-output 5.json.zip	05/18/2024 09:46:21	170878	Finished		+
4	0.0936	test-output 7.json.zip	05/19/2024 03:31:28	176332	Finished		+
5	0.0971	test-output 6.json.zip	05/19/2024 03:36:28	173870	Finished		+
6	0.0918	test-output 8.json.zip	05/19/2024 12:31:31	184556	Finished		+
7	0.0847	test-output 9.json.zip	05/21/2024 06:11:47	194523	Finished		+
8	0.0977	test-output 10.json.zip	05/21/2024 06:46:47	194918	Finished	✓	+

During the Codalab competitions for the information retrieval part, we tested 8 model configurations. The first three configurations were parametrisations of the SBERT base model, while the remaining 5 were tuning tests of the TF-IDF base model. All models, except the last one, were run with random label classification for the classification part, as we didn't have a classification model ready at that time. The submitted classification model was an SVM classifier applied over the TF-IDF encoder. When running the models, we found out that the pre-processing part was a very important component of the model results, and consequently, during the last weeks, we advocated to improve these pre-processing steps. Our best obtained result in the competition was a SCORE of 0.097 which placed us in the 36th position in the final evaluation.

8 Conclusions

Developing an automated fact-checking system for climate science claims highlights the potential and challenges of applying NLP technologies to counter misinformation. While models like Doc2Vec, SBERT, and TF-IDF have shown potential in retrieving and classifying information, the performance was not proficient, primarily due to limitations in dataset size, model overfitting, and data diversity. The small size of the training dataset led to struggles in generalisation and the risk of overfitting. In contrast, the lack of diversity in the dataset affected the robustness of the models across different domains. Additionally, the inherent complexity of evaluating sentence similarity and classifying claims accurately remains a significant challenge.

9 Team contributions

Name	Task	Contribution
TF-IDF base	Andres	100
SBERT base	Andres	100
Doc2Vec base	Claudia	100
TF-IDF + PCA + SVM Classifier	Andres	100
Doc2Vec+LR Classifier	Claudia	100
Create presentation	Claudia	50
Create presentation	Andres	25
Create presentation	Shabahz	20
Create presentation	Wanhao	5
Report	Andres	50
Report	Claudia	50

Table 3: List of activities and contributions of each team member.

References

Shariq Bashir and Akmal Saeed Khattak. 2014. Producing efficient retrievability ranks of documents using normalized retrievability scoring function. *Journal of Intelligent Information Systems*, 42(3):457–484.

Jacob Devlin et al. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*, pages 4171–4186.

Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A survey on automated fact-checking. *Transactions of the Association for Computational Linguistics*, 10:178–206.

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. *Preprint*, arXiv:1405.4053.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using siamese bert-networks. In *EMNLP-IJCNLP*, pages 3982–3992.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Ashish Vaswani et al. 2017. Attention is all you need. In *NIPS*, pages 5998–6008.

Xinyi Zhou and Reza Zafarani. 2018. A survey of fake news: Fundamental theories, detection methods, and opportunities. *arXiv preprint arXiv:1812.00398*.