Automated Fact Checking for Climate Science Claims

Student number: 1439683

Abstract

The report presents a solution to the problem of automated fact-checking of claims related to climate science. The solution consists of two components: a retrieval module that uses a sentence-transformer model to search for relevant evidence passages from a knowledge source, and a classification module that uses also uses a transformer model to classify a claim based on the found evidence. The report evaluates the performance of the solution on a dataset of climate claims and evidence passages and discusses its strengths and limitations.

1 Introduction

Climate change is one of the most pressing issues facing humanity and the planet. However, the scientific consensus on the causes and consequences of climate change is often challenged by misinformation and false claims. Therefore, it is important to have reliable and accurate methods to verify the validity of such claims and provide evidence-based information to the public. This report aims to address this problem by proposing a solution for automated fact-checking of claims related to climate science.

The proposed solution consists of two components: a retrieval module and a classification module. The retrieval module is responsible for finding the most relevant evidence passages from a large corpus of climate-related documents, given a claim. The retrieval module uses a sentence-transformer model that is fine-tuned on a dataset of claim-evidence pairs to measure the semantic similarity between sentences. The classification module is responsible for predicting the status of the claim, given the retrieved evidence passages. The classification module uses a transformer model that is fine-tuned on a dataset of claim-evidence-label triples to perform a four-way classification task. The possible labels are {SUPPORTS, REFUTES, NOT_ENOUGH_INFO, DISPUTED}, indicating whether the evidence supports or refutes the claim, whether there is not enough information to make a judgment, or whether the claim is disputed by different sources.

The report evaluates the performance of the proposed solution on a test set of climate claims and evidence passages. The report also discusses the strengths and limitations of the proposed solution and suggests some directions for future work.

2 Literature Review

In this section, we briefly review literature relevant to our automated fact-checking system. First, we introduce the dataset used in this project, after which we analyze the literature.

2.1 Dataset

The dataset used for this project consists of 1535 claims. All of which have a claim text. 1382 of these claims have evidence indexes and labels and are used for training and validation. The remaining 153 claims only have claim text and are used as test data to evaluate both the retrieval and classification modules.

Additionally, there is a large evidence file which links 1208827 evidence indexes to their respective evidence text. It is important to note that the evidence file is very large and may require a lot of memory and processing power to load and query. Therefore, the retrieval module must be implemented in such a way that it is able to handle this.

2.2 Related literature

As mentioned previously, our task of evidence-based claim verification consists of two steps: evidence retrieval and claim classification.

One of the most popular and effective approaches for both steps is to use transformer models, such as BERT (Devlin et al., 2019) or it's robustly optimized variant RoBERTa (Liu et al., 2019). Transformer models are pre-trained on large corpora using self-supervised objectives and then fine-tuned on specific tasks using supervised learning. Transformer models can capture rich semantic and syntactic information from natural language texts and generate contextualized representations for words and sentences. It is worth mentioning that our problem necessitates the use of embeddings for text corpora. Therefore, we can leverage the benefits of Sentence-BERT (Reimers and Gurevych, 2019) or its variants. Sentence-BERT is a variant of BERT that is tailored to generate embeddings for text corpora. It achieves this by utilizing a Siamese network architecture to learn sentence embeddings that are optimized for semantic similarity.

One approach that may benefit us for the evidence retrieval part of the project is BM25, a ranking function based on the bag-of-words model that assigns scores to documents according to their relevance to a query. BM25 uses term frequency and inverse document frequency to measure how important a word is in a document and in a collection of documents, respectively. BM25 can be used as a baseline or a filtering method for evidence retrieval.

3 Methods

In this section, we will discuss the methods we used to develop our fact-checking system. We will first examine the retrieval module, then the classification module. We will also briefly touch upon some of our unsuccessful attempts.

3.1 Evidence Retrieval Module Methods

Initially, we experimented with a method that used BM25 to retrieve the top 200 similar pieces of evidence for each claim, generated encodings for all potential evidence using Google's Universal Sentence Encoder (USE) (Cer et al., 2019), and then calculated the cosine similarity between the claim encodings and each sentence encoding. If the similarity exceeded a certain threshold, we considered the evidence as being associated to the claim. This proved to yield bad results, however. The main reason for which being that even though USE is a general-purpose model, it may not capture the nuances and subtleties that are relevant in finding evidences that are related to a claim.

Thus, We opted for a different approach. We fine-tune a sentence DistilRoBERTa transformer model (a smaller, faster, and more resource-efficient version of RoBERTa) to get encodings more suitable for our task. The way that our pipeline works is that initially, we prepare the data in such a way that we have all the related claim-evidence pairs. For example, if claim1 is known as being associated with evidence23 and evidence25, then the following would be prepared: {["claim1 text", "evidence23 text"], ["claim1 text", "evidence25 text"]}. Next, we fine-tune the model on all claim-evidence pairs using a Multiple Negative Rank loss function which is suitable for this task. Once the training phase is complete, the model will be capable of producing sentence encodings that capture the relevant subtleties in finding evidence.

Finally, we repeated our previous approach, which involved identifying the top candidate evidence using BM25. However, instead of encoding the claim and potential evidence texts using USE, we used the fine-tuned model for encoding. It must be noted that if none of the candidate evidences were similar enough to the claim, then we select a few evidences that have the most similarity to the claim, with this number being a tunable hyperparameter.

.

3.2 Claim Classification Module Methods

In order to create a claim classifier, we also opted to fine-tune a sentence transformer model. However, we had to consider the differences between this task and evidence retrieval task. In this particular task, our objective is to classify a claim based on its text and the associated evidence texts into one of four possible labels. As a result, we had to modify the architecture of the DistilRoBERTa model by replacing its output layer with a new dense layer that has a number of output nodes equal to the number of classes (in our case, four). We preprocessed the supervised training data by merging each claim text and its corresponding evidence texts into a single text that can be fed into the model. To achieve this, we combined the claim and evidence texts using a special separator token. In the case of DistilRoBERTa, the separator token is "</s>". For instance, if claim1 is associated with evidence23 and evidence25, we preprocess the text as ["claim1 text </s> evidence23 text </s> evidence25 text"].

Following this, we fine-tune the model on all the merged claim-evidence texts. After the training phase is complete, the model will now be capable of predicting the labels of claims based on the evidences they are associated with.

4 Results

In this section, we will discuss the results we obtained using the discussed methods. Similar to the previous section, we will first examine the retrieval module, then the classification module.

4.1 Evidence Retrieval Module Results

Due to the high computation time that was required to fine-tune the model and inference from it, we first fine-tuned a model with one single epoch and a batch size of 32. We then used this model to try different hyperparameters relating to using BM25.

These hyperparameters included the number of candidates to pick (#num_candidates), the threshold relating to how similar an encoded candidate must be to the encoded claim based on their cosine distance (Threshold), and the number of top evidences to add in the case that none of the candidate evidences were selected (#to_add).

Table 1 displays the F-Scores obtained after tuning these hyperparameters on the validation set. It is apparent that the best hyperparameters based on our search is (200, 0.8, 2).

We continued to fine-tune the model for an additional two rounds, with the number of epochs set to five for each round. One round involved using the original training set as the training data, while the other round involved using the complete labeled dataset (train+validation) as the training data.

#num_candidates	Threshold	#to_add	F-score
150	0.85	0	0.036
200	0.85	1	0.095
200	0.80	2	0.131
200	0.85	2	0.07

Table 1: retrieval module scores

After following the BM25 procedure on the validation set using both models, results as seen in Table 2 were obtained.

4.2 Claim Classification Module Results

Training data	F-score
Validation-set	0.146
Train + validation	0.187

Table 2: F-scores using varying models

We first analyzed whether the data we had was balanced or not by plotting the number of labels of each class present in the training and validation data (figure 1)

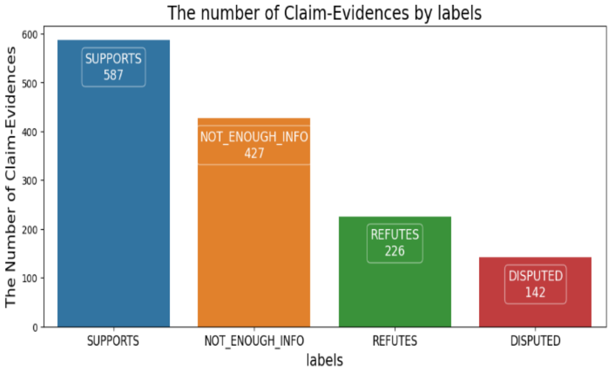

Figure 1: Data labels of each class

From this figure, it is apparent that the data is imbalanced, with a significant bias towards the "SUPPORTS" and "NOT_ENOUGH_INFO" labels. To address this bias, the model can be trained by incorporating class weights. This involves assigning weights to each class in a manner such that classes with high label counts are weighted less, and vice versa.

Our primary objective was to achieve good performance on the test data. We anticipated that the test data itself may also be imbalanced. As a result, we decided to fine-tune two DistilRoBERTa models - one taking into account the class weights, and the other not - to evaluate their respective performances.

However, due to the very high computing time of fine-tuning these models, we were only able to train a model which doesn't take into account the class weights. This trained model was trained over 5 epochs with a batch size of 16.

Table 3 displays the training loss and validation loss over the iterations taken while training.

Steps	**Training Loss**	**Validation Loss**
50	1.24	1.26
100	1.00	1.00
150	0.91	0.93
200	0.87	0.97

Table 3: loss over training iterations

After training this model, it was inferenced one final time on the validation set and a classification accuracy of 0.55 was obtained on it.

5 Conclusion

We developed an automated fact-checking system that can link potential evidence to climate science claims and classify the claims based on the evidence. This was accomplished by fine-tuning state-of-the-art NLP transformer models. It is important to note that the training of these models has high computational complexity, and thus, the best hyperparameters may not have been identified. Future work can focus on exploring hyperparameter optimization to improve the system's performance.

References

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.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A Robustly Optimized BERT Pre-training Approach with Post-training. In Proceedings of the 20th Chinese National Conference on Computational Linguistics, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

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 and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal Sentence Encoder for English. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 169–174, Brussels, Belgium. Association for Computational Linguistics.