Fact-Checking System for Claims Related to Climate Science

Tue10AM-Group5

Abstract

This report presents approaches for automated fact-checking of claims related to climate science. It describes the development of an evidence retrieval module to identify relevant evidence passages from a corpus, and a label classification module to categorize claims based on the retrieved evidences. Various techniques like TF-IDF filtering, transformer encoder models, and recurrent neural networks (LSTM, RNN, GRU) are explored. Experiments reveal limitations in both modules leading to suboptimal performance, and highlighting the need for further improvements in areas such as data quality, model architectures, and training strategies.

1 Introduction

Climate change is a critical global issue where misinformation and false claims cloud the scientific consensus. Consequently, there is a pressing need for reliable automated fact-checking methods to verify climate-related claims and provide evidence-based information. This report aims to address this problem by proposing a solution for automated fact-checking of claims about climate science. The proposed approach consists of two main modules: 1) An evidence retrieval module to identify the most relevant evidence passages from a large corpus based on a given claim, and 2) A label classification module to categorize the claim into one of four classes (SUPPORTS, REFUTES, NOT_ENOUGH_INFO, DISPUTED) using the retrieved evidence. The report describes the data, approaches, experiments, and results for developing these modules.

2 Data Set

The following data set will be used in this project:

a) Training Data Set:

train-claims.json, each instance contains the claim ID, claim text, claim label (one of the four classes: {SUPPORTS, REFUTES, NOT_ENOUGH_INFO, DISPUTED}), and a list of evidence IDs. (1228 instances in total)

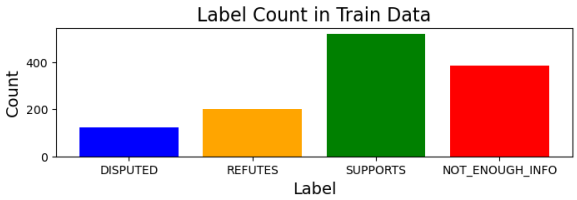

Figure 1: Claim Label Distribution of Training Data Set

The analysis of the data distribution reveals a significant imbalance across the different classes in the training dataset, which can be observed in Figure 1. It is evident that the instances labelled as 'SUPPORTS' and 'NOT_ENOUGH_INFO' are substantially more numerous compared to those labelled as 'DISPUTED' and 'REFUTES'. This imbalanced class representation poses challenges to the model's ability to learn and generalize effectively, as the underrepresented classes may not receive sufficient representation during the training process (Khan et al., 2017).

b) Development Data Set

dev-claims.json: JSON files for the labelled development set (154 instances in total).*dev-claims-baseline.json*: JSON file containing predictions of a baseline system on the development set

c) Test Data Set

test-claims-unlabelled.json: JSON file for the unlabelled test set (153 instances in total)

d) Evidence DataSet

evidence.json: JSON file containing a large number of evidence passages (1,208,827 instances in total). Out of the total evidence passages, 1,103,245 are in English.

3 Approaches

3.1 Pre-processing

For pre-processing, we first perform tokenization to split text into words, remove stopwords, and apply lemmatization to convert words to their base forms. This process helps to clean, normalize, and prepare textual data for better performance and interoperability. Additionally, we want to address the reason that we use lemmatization here since most modern NLP models may not require explicit lemmatization due to their ability to capture word semantics implicitly. For models trained on small datasets without pre-trained weights, lemmatization aids learning by reducing vocabulary size, mitigating sparsity, and grouping inflectional forms to capture word relationships better (Bergmanis, 2020).

3.2 Evidence Retrieval Module

The evidence retrieval module identifies and retrieves the most pertinent evidence passages from an extensive corpus of climate-related documents based on a given claim.

3.2.1 Initial Filtering

Since the evidence data set is too big and contains evidences related to all domains, it is necessary to narrow down the massive evidence corpus to only passages relevant to climate change and environmental topics.

1. TF-IDF (frequency-inverse document frequency) Filtering

To find the relevant evidence for a given claim, we can utilize the TF-IDF embedding technique to vectorize both the claim and the evidence. TF-IDF is a numerical statistic that reflects the importance of a word to a document within a corpus (Bafna et al., 2016). Once the claim and evidence are represented as TF-IDF vectors, we can calculate the cosine similarity between the claim vector and each evidence vector, providing a measure of their similarity independent of magnitude. Then, we can rank the evidence by their similarity scores and select the top 100 pieces of evidence as the most relevant to the claim.

2. Latent Dirichlet Allocation (LDA) Filtering LDA is a generative statistical model that discovers abstract topics within a collection of documents (Blei et al., 2003). By applying LDA to the evidence corpus, the system can identify passages that discuss climate change and related topics based on the co-occurrence of words.

3.2.2 Modelling

Firstly, we will use training data set to train a Word2Vec model. Then, we will this Word2Vec to generate word embeddings for claim and evidence, capturing the semantics and contextual relationships of words in the text. Using Word2Vec can provide useful initialization for the deep learning model, accelerating convergence and improving performance (Mahsuli and Safabakhsh, 2017). Afterwards, we will construct a Transformer encoder, which is mainly composed of the following components: 1. Word embedding layer, mapping each word in the text into a vector representation, using embeddings from self-trained Word2Vec as the initialization weights for the Transformer model's embedding layer. 2. Positional encoding, adding positional information to capture word order. 3. Encoder layer, using a self-attention mechanism to process the input sequence, capturing the contextual relationships between words. 4. Pooling layer, extracting key features from the encoder output. 5. Fully connected layer and output layer, mapping features to the prediction of the matching relationship between claims and evidence. Within the on-topic subset, the Transformer encoder will produce a probability score for a given claim and evidence pair. We will choose the top three evidence passages with the highest probability scores as our retrieved pieces of evidence.

3.3 Class Label Classification Module

The second step in the automated fact-checking process is to classify the claim into one of four categories: SUPPORTS, REFUTES, NOT_ENOUGH_INFO, or DISPUTED. This is done using neural network architectures, including LSTM, RNN, and GRU models. Each model uses an embedding layer to convert the input text into dense vector representations. These embeddings are passed through bidirectional recurrent layers, which process the input sequence in both forward and backward directions, capturing context from past and future word (Arisoy et al., 2015) (Yao and Huang, 2016).

(1) **LSTM:** The LSTM model uses stacked bidirectional LSTM layers with dropout regularization to prevent overfitting. LSTM is a type of recurrent neural network that can learn long-term dependencies by using gating mechanisms to control the flow of information (Hochreiter and Schmidhuber, 1997). The model also employs L2 regularization

on the LSTM kernels to further reduce overfitting.

(2) **RNN:** The RNN model uses simpler bidirectional RNN layers, which do not have the gating mechanisms of LSTM. RNNs are a class of neural networks that can process sequential data by maintaining a hidden state that is updated at each time step (Salehinejad et al., 2017). While RNNs are simpler than LSTMs, they can still capture important patterns in the input sequence.

(3) **GRU**: The GRU model uses bidirectional GRU layers, which are a variation of the LSTM architecture. GRUs have fewer parameters than LSTMs, as they combine the forget and input gates into a single update gate (Shewalkar et al., 2019). This makes GRUs computationally more efficient while still maintaining the ability to capture long-term dependencies (Mateus et al., 2021).

All three models have a dense output layer with softmax activation, which produces a probability distribution over the four classification categories. During training, the models use *SparseCategoricalCrossentropy* and incorporate techniques like early stopping to prevent overfitting and save the best-performing weights.

4 Experiments

4.1 Evidence Retrieval Module

4.1.1 Initial Filtering

Experiment with LDA Filtering: We tested generate 5, 10, 15, 20, 50 topics from evidence data set, however, by human evaluation, none of the generated topics is related to climate. These skewed expectations could be caused by the quality of the evidence data itself. The dataset might contain only a tiny amount of evidence related to the climate.

4.1.2 Modelling

Hyper-parameter Tuning for Word2Vec: We will evaluate performance using the average F-score, computed by calculating the F-score for each claim by comparing retrieved evidences with ground truth, then averaging across all claims. We chose to experiment with the Word2Vec vector size, which determines the representational capacity of word embeddings to capture semantic and contextual text relationships.

From Table 1 we can observe that increasing the vector_size from 300 to 700 did not improve the Average F-score and caused a slight decrease in performance. However, further increasing it to 900 led to an improvement, suggesting that a larger vector_size may capture more complex relationships in the data. The relationship between vector_size and Average F-score appears non-linear, with a decrease from 300 to 700 followed by an increase at 900. The optimal vector_size likely depends on the dataset, task, and other hyperparameters used in the Word2Vec model. Among the tested values, a vector_size of 900 maximized the Average F-score.

vector size	Average F-score
300	0.058
700	0.053
900	0.063

Table 1: The Average F-score of Word2Vec vs vector size (top k = 3, window 5, min-count 2, workers 4, epoches 20)

input dimension	Evidence Retrieval F-score
300	0.012368583797155226
700	0.05088126159554732
1500	0.09831478045763758

Table 2: The Evidence Retrieval F-score of Transfomer vs Input Dimension (top k = 3, hidden-dim = 512, n-heads 15, n-layers = 6, droppout-rate = 0.2, loss function BCELoss, epoch 5, learning rate 0.001)

Hyper-parameter Tuning for Transformer: We choose to tune the input dimension aimed at allowing higher-dimensional word embeddings to better capture semantic and contextual nuances within claim and evidence texts. This trend in Table 2 suggests that increasing the input dimension for the Transformer model has a positive impact on its ability to retrieve relevant evidence accurately. A higher input dimension likely allows the model to capture more complex relationships and patterns in the input data, leading to better performance in evidence retrieval. However, it's important to note that increasing the input dimension may also increase the computational complexity and memory requirements of the model, which could make training and inference more resource-intensive.

4.2 Class Label Classification

Experiments with Early Stopping: Small datasets like the 1228 sample training set for this project increase the risk of neural networks overfitting by excessively memorizing training data and noise patterns instead of learning the true underlying distribution.

For LSTM, training and validation accuracy fluc-

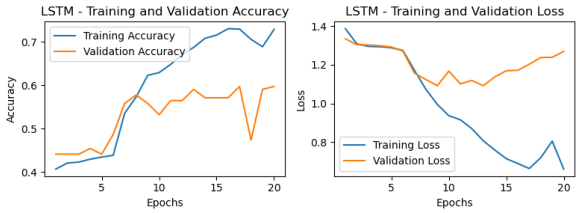

Figure 2: Training and Validation Accuracy and Loss of LSTM on Label Classification Task

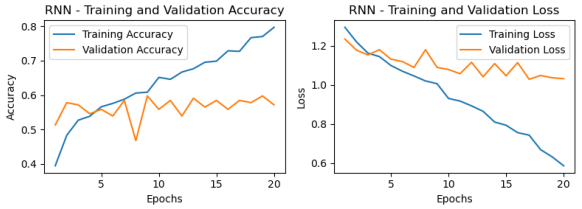

Figure 3: Training and Validation Accuracy and Loss of RNN on Label Classification Task

tuate but generally stabilize around 0.7 and 0.6 separately. The validation loss steadily decreases, indicating good generalization. However, spikes in the training loss suggest potential overfitting or instability.

For RNN, the training accuracy reaches around 0.8, but validation accuracy remains low around 0.5-0.6, indicating overfitting to training data and poor generalization, further confirmed by a large gap between training and validation loss curves.

For GRU, training accuracy curves show a significant increase in training accuracy towards the end, while validation accuracy dips slightly, indicating potential overfitting. The training loss decreases sharply, diverging from the relatively stable validation loss, further suggesting overfitting.

Based on the above experiments, it is essential to conduct actions to prevent overfitting, we employ early stopping in this project.

For the LSTM model, the accuracy on the dev

Model	Early Stopping	Accuracy
LSTM	Yes	0.597402597402
LSTM	No	0.597402597402
RNN	Yes	0.5
RNN	No	0.53246753246
GRU	Yes	0.448051948051
GRU	No	0.493506493506

Table 3: Table of The Effect of Early Stopping on LSTM, RNN, and GRU (epochs = 20, batch_size = 64)

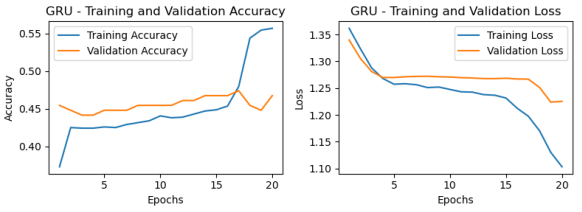

Figure 4: Training and Validation Accuracy and Loss of GRU on Label Classification Task

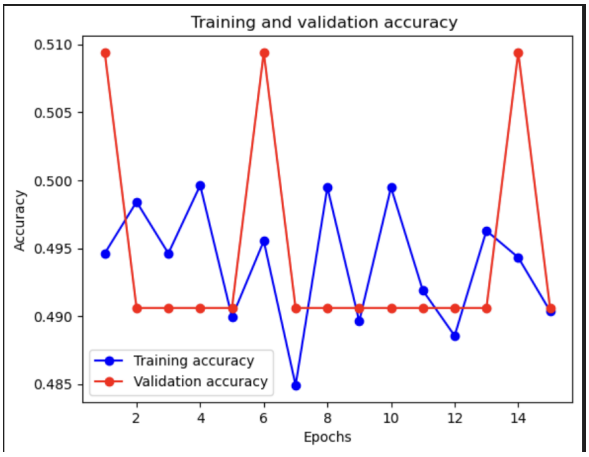

Figure 5: Training and Validation Accuracy of Transfomer Encoder

set is the same with or without early stopping. This could mean that the model has already converged before reaching the maximum number of epochs. For the RNN model, the accuracy on the dev set is higher without early stopping compared to with early stopping. This suggests that early stopping may have stopped the training prematurely before the model could fully converge and reach its maximum potential performance. For the GRU model, the accuracy on the dev set is higher without early stopping compared to with early stopping. Similar to the RNN model, early stopping may have prevented the GRU model from achieving its best possible performance.

5 Result

5.1 Evidence Retrieval Module Result

Based on above experiment, the best F-score we achieved for evidence retrieval module is 0.09831478045763758 which is quite low .This performance could be attributed to several factors in the module's working mechanism. The initial filtering stage using TF-IDF may have had a low recall, filtering out many relevant evidence passages. From Figure 5, we can observe that the training

accuracy curve exhibits a fluctuating pattern, indicating potential overfitting, where the model is memorizing the training data rather than learning generalizable patterns. Furthermore, there is a significant gap between the training and validation accuracy curves, especially in the later epochs. This gap suggests that the model is overfitting to the training data and not generalizing well to new, unseen data. Additionally, both curves exhibit peaks and valleys, indicating instability in the training process, which could be due to factors like learning rate, regularization, or the complexity of the model architecture. Addressing these potential issues across the various components of the module is crucial to improve its evidence retrieval capability.

5.2 Label Classification Module Result

Given the relatively low performance of the evidence retrieval module, to mitigate the potential for error propagation and accurately evaluate the true capabilities of the label classification module, we will analyze the accuracy conditioned on the correct prediction of evidence.

Model	Accuracy
LSTM	0.597402597402
RNN	0.532467532467
GRU	0.493506493506
Majority Vote Classifier	0.441558441558

Table 4: Label Classification Accuracy on Development Data Set (Predicted on True Evidence List)

The best performance among trials is achieved using LSTM with a accuracy of 0.597402597402, while being the best among the evaluated models, does not seem to meet the expected level of accuracy. An accuracy of 0.597402597402 for a multi-class classification task is relatively modest and leaves room for improvement. Considering that the Majority Voting Classifier, which is a simple baseline, achieved an accuracy of 0.441558441558, the performance of LSTM, RNN, GRU is better than this naive approach, which means this model did learned something from the training data set, but not much.

There could be several reasons why these model's performance is not exceptional: (1)The task of classifying claims into one of four classes based on retrieved evidence passages may be inherently complex, involving nuances and ambiguities that make it challenging for the models to achieve higher accuracy. (2) The quality and quantity of the training data could be a limiting factor. Our training data is imbalanced and insufficient as Section 2 demonstrated, it may hinder the models' ability to learn effective representations and generalize well.

	Baseline
Evidence Retrieval F-score (F)	0.337770562770
Classification Accuracy (A)	0.350649350649
Harmonic Mean of F and A	0.344089490135
	Our Best System
Evidence Retrieval F-score (F)	0.0983147804576
Classification Accuracy (A)	0.6038961038961
Harmonic Mean of F and A	0.1690999504469

Table 5: The Performance of Baseline and Our Best System

5.3 Whole System Result

Compared to the baseline system's harmonic mean of 0.344089490135, our system performs significantly worse with a harmonic mean of 0.1690999504469. This poor performance is primarily due to the low F-score of 0.0983147804576 in the evidence retrieval module, which is only about 1/3 of the baseline. Despite the reasons mentioned in section 5.1, the most likely reason is that we didn't employ a pre-trained language model that offers transfer learning capabilities, contextual understanding, semantic matching, and domain adaptation.

6 Conclusion

In conclusion, while our proposed automated fact-checking system showed some learning capability, its overall performance was limited by challenges in the evidence retrieval and label classification modules. Key issues included filtering problems, overfitting, training instability, task complexity, and data quality/quantity constraints. The modest results underscore the need for improvements through alternative architectures, data augmentation, ensemble methods, and error analysis. Despite current limitations, this work lays a foundation for further research towards developing more reliable and robust fact-checking solutions to combat climate change misinformation.

References

Ebru Arisoy, Abhinav Sethy, Bhuvana Ramabhadran, and Stanley Chen. 2015. Bidirectional recurrent neu-

ral network language models for automatic speech recognition. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5421–5425. IEEE.

Prafulla Bafna, Dhanya Pramod, and Anagha Vaidya. 2016. Document clustering: Tf-idf approach. In *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, pages 61–66. IEEE.

Toms Bergmanis. 2020. Methods for morphology learning in low (er)-resource scenarios.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Salman H Khan, Munawar Hayat, Mohammed Bennamoun, Ferdous A Sohel, and Roberto Togneri. 2017. Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE transactions on neural networks and learning systems*, 29(8):3573–3587.

Mohammad Mahdi Mahsuli and Reza Safabakhsh. 2017. English to persian transliteration using attention-based approach in deep learning. In *2017 Iranian Conference on Electrical Engineering (ICEE)*, pages 174–178. IEEE.

Balduíno César Mateus, Mateus Mendes, José Torres Farinha, Rui Assis, and António Marques Cardoso. 2021. Comparing lstm and gru models to predict the condition of a pulp paper press. *Energies*, 14(21):6958.

Hojjat Salehinejad, Sharan Sankar, Joseph Barfett, Errol Colak, and Shahrokh Valaee. 2017. Recent advances in recurrent neural networks. *arXiv preprint arXiv:1801.01078*.

Apeksha Shewalkar, Deepika Nyavanandi, and Simone A Ludwig. 2019. Performance evaluation of deep neural networks applied to speech recognition: Rnn, lstm and gru. *Journal of Artificial Intelligence and Soft Computing Research*, 9(4):235–245.

Yushi Yao and Zheng Huang. 2016. Bi-directional lstm recurrent neural network for chinese word segmentation. In *Neural Information Processing*, pages 345–353, Cham. Springer International Publishing.

7 Team Contribution

We divided the tasks into: Analysis Task 1, Task 2, experimental data collection, and report writing. Yun Zhu, Steven, and Yufan Zhang: participated in every stage of the project. Each member not only contributed their expertise in their areas of specialization but also actively contributed to other areas, ensuring the project was of high quality and efficiency from start to finish.