COMP90042 Project 2023

Automated Fact Checking For Climate Science Claims

1067750

1 Introduction

Climate change is consistently a primary concern for humanity, and its related information has become more easily accessed along with the development of the Internet platform. However, once the information could be easily spread, the climate change 'claim' reliability requires more effective fact-checks to avoid twisting public opinion.

In this research, evidence retrieval algorithms are designed or revised to obtain precise climate science 'claim-evidence' matching. Besides, once the evidence is retrieved, a claim labelling model is built for distinguishing the 'claim' for fact-checking usage. Furthermore, the current research limitation is discussed after obtaining the result and indicate future direction and consideration.

2 Data Exploration

In this research, training, development and test sets are provided in JSON, where true claim-evidence(s) pairs are recorded in both training and development sets with each claim-fact's label stored after each pair, while the test sets only include the claim's data. Claims and evidence are presented in sentences, indicating an embedding method is needed to process the data before modelling.

Besides acknowledging the data type and format, exploration of the data distribution before creating and implementing the algorithm is also necessary since it guides the development of the model and leads to the decision-making of model choosing.

Evidence	Count	Count (Unique)
Training	4122	3121
Development	491	463
TOTAL	1208827	1193766

Table 1: Evidence Usage in Training and Development

Presented in Table 1, data accessed is unbalanced, The training evidence only occupies 0.4% of the evidence corpus, which is insufficient to reveal the overall evidence's feature, indicating an unsupervised model will be preferred, especially in Task 1 that retrieves evidence from the large corpus.

Label	Train	Develop
SUPPORTS	519	68
REFUTES	199	27
NOT_ENOUGH_INFO	386	41
DISPUTED	124	18
TOTAL	1228	154

Table 2: Claim's Fact Label Distribution

Besides, unbalance also exists in claim's fact distribution, where 'SUPPORT' occupied the majority from Table 2. Therefore, the baseline (e.g. 0-R Model) of the fact-checking performance can be obtained for evaluating and filtering the designed claim labelling algorithm in Task 2.

3 Task 1: Evidence Retrieval

3.1 Preprocessing

Preprocessing steps such as 'Letter-Case Uniformation' and 'Stopwords Removal' are investigated. The lowercase operation is applied to both claim and evidence text, which reduces the embedded vector's sparsity. However, 'stopwords' were preserved even though the sparsity can be further reduced once removed. Presented in the article (Wilame, 2019), when processing the text classification linked to the sentiment analysis, removing stopwords will alter the sentence meaning. Similarly, in this research, stopwords removal will change claim or evidence's perspective, leading to a mismatching after embedding.

3.2 Embedding

Since the claim and evidence are provided in sentence(s), it is essential to convert them into vectors, which represent the text's meaning and can be fed into models (Selva Birunda and Kanniga Devi, 2021). Besides, informative text-embedded vectors

are required since an unsupervised model based on matching similarity is preferred. Thus, this task will prioritise obtaining a more precise vectoriser.

3.2.1 Bag-of-Word (BoW)

BoW was first chosen. It counts the word in a sentence and represents the sentence in a vector with length as the number of words in the corpus. This method is easy to realise since it embeds all words independently and equally and has been treated as the baseline for vectorising the text (Tran et al., 2015). In this task, the BoW embeds both claim and evidence sentences, allowing us to retrieve the evidence with the highest similarity in word occurrence to the investigated claim.

3.2.2 doc2vec

Beyond the word2vec embedding, which maps the word into a higher dimension for representing the semantic similarity between words, doc2vec has extended its functionality to a document level and biassed towards the content words, which brings robust performance in matching the sentences. According to the evaluation research (Lau and Baldwin, 2016), the doc2vec embedding has outperformed the word2vec in the duplicated questions detection task, which mainly relies on cosine similarity between the embedded vectors. Since the evidence retrieval relies on similar logic that computes the similarity between the claim and evidence, doc2vec can be extended to this task.

3.2.3 BERT

BERT (pre-trained bert-base-uncased) was chosen in this task for embedding since it is the state-of-the-art method that captures the deep bidirectional representation rather than only reading sentences left-to-right (Devlin et al., 2018). Besides, rather than only focusing on the word when embedding by traditional or static methods, applying the Masked Language Model based on BERT can represent the whole sentence context by one vector (e.g. vector of [CLS]). Therefore, BERT is theoretically the optimal embedding method when handling a task requiring the vectorisation of claim and evidence sentences for downstream application.

3.2.4 TF-IDF

TF-IDF is an empirical choice for embedding. It computes the number of times the token appears in a sentence (tf) and the inverse of its appearance in the corpus (idf) (Tata and Patel, 2007). By multiplying tf and idf, weight(s) of the rare and sentence-specific token(s) is leveraged higher and becomes the sentence's topic representative. By vectorising the claim and evidence text with the TF-IDF, each sentence's keyword(s) could obtain a higher weight(s) and reduce the noise from the common uninformative word for calculating the similarity between the vectorised claim and evidence.

3.3 Algorithms' Evaluation & Analysis

In this task, kNN based on cosine-similarity is applied as the downstream model for retrieving evidence(s) after embedding, which reveals the performance of the selected vectoriser (Lau, 2023b).

Model	Dev: F	Partial Test: F
BoW-kNN	0.0305	0.0509
doc2vec-kNN	0.0000	0.0000
BERT-kNN	0.0286	0.0143
TF-IDF-kNN	0.0911	0.0883

Table 3: kNN (k=3) F-Socre with Different Embedding

In Table 3, the performance of kNN (k = 3 since average evidence amount of training pairs is 3.18) with different vectoriser is presented in F-Score after processing the development and partial test set. It is also used as a guideline for further investigation on the best-performing vectoriser.

3.3.1 BoW-kNN

The kNN model with k = 3 and BoW embedded vector has an overall low performance. This result is reasonable since BoW has treated all words equally and has been misled by the high frequency but uninformative word(s) when calculating the cosine-similarity. Compared to the TF-IDF version, the model with BoW embedding underperforms and will not be further investigated.

3.3.2 doc2vec-kNN

The doc2vec embedding method has performed worst based on the Table 3 result, which two reasons might trigger. Firstly, the doc2vec requires a hyper-parameter of the dimension to which the word is mapped, which is hard to obtain its optimal value. Besides, since the claim and evidence word corpus is vast, using a lower dimension for mapping will reduce its performance, while mapping to a similar dimension as the word corpus led to RAM overflow. Based on these issues, doc2vec embedding is unsuitable for the current task.

3.3.3 BERT-kNN

The model with pre-trained BERT embedding has unexpectedly underperformed despite capturing the word's context during vectorisation. One potential reason for this poor performance is inability to obtain the task-specific word's weight. The BERT is pre-trained based on Wikipedia + BookCorpus (Lau, 2023a) to obtain an initialised weight of each token. However, the task is to retrieve evidence based on the claim text from the 'Climate Science' domain, which requires greater attention to the keyword (e.g. Atmosphere). That is, without reweighing each word before vectorisation, the pre-trained BERT cannot leverage out the climate-science-related evidence precisely, leading to failure in matching evidence with the given claim.

3.3.4 TF-IDF-kNN

According to Table 3, under the same kNN conditions, the TF-IDF version has obtained the optimal performance with F-Score significantly greater than others. TF-IDF has allocated greater weights to the rare and topic-specific words during embedding, making the climate science terminology word receive higher weight in each embedded sentence's vector. Thus, when computing the cosine-similarity between the claim and evidence's vectors, a more significant difference between matched and unmatched pairs will produce a better retrieval result. Since the TF-IDF-kNN has optimal performance, it is worth further research for improvement.

3.3.5 TF-IDF Further Investigation

According to the research (Yadav et al., 2020), retrieving evidence can be treated as a Question Answering (QA) task where claims are questions and evidence(s) is the answer. Thus partial logic from QA, extracting potential documents/passages based on the question's key feature (Lau, 2023c), can be concat to the current model. Using the climate science keywords summarised from the training text to filter the corpus before retrieving evidence.

In this research, the keywords have chosen the most frequent word(s) (except stopwords) that appear in the training evidence, which guarantees the collected word is mostly from climate science. After collecting the keywords, they are then applied to filter the evidence corpus by only choosing the topic-related evidence for TF-IDF embedding and text-matching usage. In optimising the model, two hyper-parameters require tuning (Number of Keywords and Number of Neighbours k).

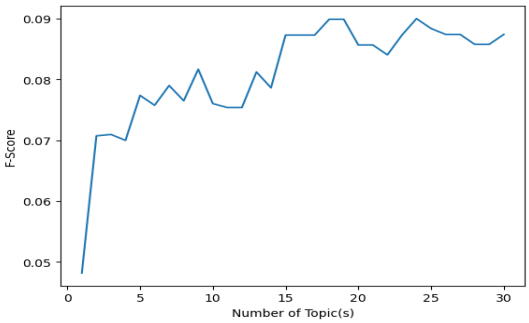

Figure 1: kNN (k=3) with Topic-Filtered TF-IDF

Figure 1 presented the optimal F-Score under the 3NN model with TF-IDF embedding when 20 keywords were selected. Following a greedy approach, different models are tested by altering the k value from one to five (min to max amount of matched evidence from the training set) after extracting the potential evidence with 20 keywords. Then, presented in Figure 2 below, k = 2 optimises the model and concludes the current best algorithm for task 1.

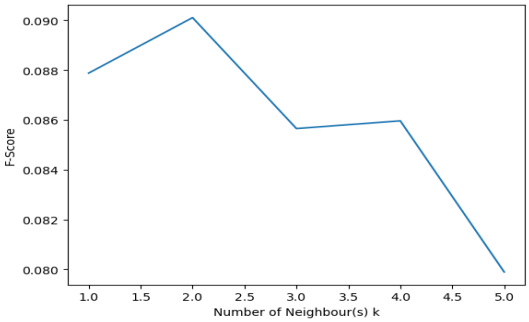

Figure 2: k Tuning with Topic-Fildered TF-IDF

3.3.6 Task 1 Final Evaluation

One model was chosen automatically based on the partial test set performance for final evaluation, which is the 2NN with TF-IDF embedding all claims and evidence after filtering the corpus with 20 keywords. The F-Score is 0.029, which is unexpectedly lower compared to the local development test with F = 0.0901, indicating an overfitting during training. The overfitting is likely due to the biased in topics keywords distribution, in which the captured keywords from training and development evidence are exclusively independent of the unrevealed test claim-related evidence.

4 Task 2: Claim Labelling (Classification)

Task 2 aims to label the claim based on the optimal retrieved evidence(s), similar to text classification. Since claim-evidence(s) content is input for classification, preprocessing of merge the matched claim-evidences sentences and embed it with Task 1 TF-IDF is required before building models.

4.1 Model Construction

4.1.1 Support Vector Machine (SVM)

SVM, the empirical supervised method for text classification, was considered since it can handle large text/embedded input and applies non-linear kernel tricks for better performance (Lau, 2023d). In this task, by treating the merged matched claim-evidence(s)' vector as 'x' and the fact label of the claim as 'y', a multi-class non-linear rbf-SVM was trained with the logic of 'one-versus-rest'.

4.1.2 Feed-Forward Neural Network (FNN)

Since the TF-IDF vector of claim-evidence has less sequential dependency but reveals more in vector feature, FNN was then considered for processing the embedded text multi-class classification (Prasanna and Rao, 2018). In this task, following the 'Universal Approximation Theorem', non-linear activation functions are added between hidden layers and obtain the fitted parameter of each embedded merged claim-evidences. Furthermore, Dropout was applied to avoid overfitting the training set before predicting the claim's label.

4.1.3 Weighted Voting Model

The unsupervised model is suitable for text classification. The kNN-based majority voting has classified the news to different classes (Usman et al., 2016). With this finding, improvements can be made by introducing a weighted voting model based on the kNN measured in similarities.

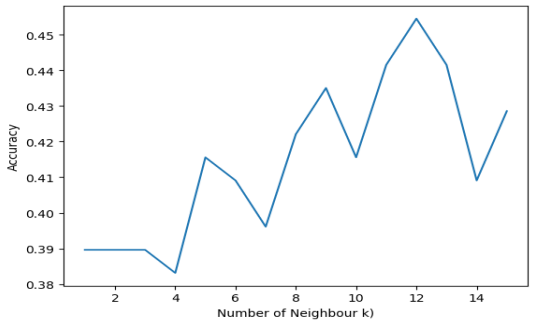

Figure 3: kNN Tuning for Downstream Voting

Presented in Figure 3, the hyper-parameter k = 12 is first obtained to optimise the whole model based on evaluation in the development set. Then, for each embedded text's vector, 12 nearest neighbour training data are recorded. However, rather than counting the matched training data labels, the similarity is aggregated, which quantifies the effectiveness of the training to test data (further the neighbour, less contribution to affect labelling) and picks the labels with the greatest aggregated similarity.

4.2 Model Evaluation & Analysis

From Table 4, FNN performed poorly, which might be related to still overfitting the training data even though dropout is processed between layers or unsuitable architecture is constructed and fail to capture input patterns. The weighted voting performs worse in the test claim's label, potentially related to the unbalanced distribution of the training label since the most frequently appeared 'SUPPORTS' is captured easier and distorts the voting result. Both are discarded as FNN worse than the 0-R (0.44155) while weighted voting has dropped significantly though it initially outperforms 0-R.

Model	Dev: A	Partial Test: A
SVM	0.4416	0.4474
FNN	0.3442	0.3684
Weighted Voting	0.4545	0.3289

Table 4: Task 2 Model's Accuracy Rate

The SVM performs well overall since it would not be affected by the unbalanced training label's distribution but focus on how to compute a non-linear boundary that precisely distinguishes classes, making it as the local-optimal model for Task 2.

5 Conclusion, Final Evaluation & Future

Summarising the above two tasks, the automated fact-checking algorithm can be constructed with Evidence Retrieval Model (Evidence Filtering + TF-IDF Embed + 2NN) and Claim Labelling Model (SVM based on embedded Task 1 matched claim-evidence text). Evaluating the algorithm's development performance, a score of 0.1504 (harmonic mean, which considers the performance of Task 1 and 2 together) is obtained.

However, due to the limited accessible evidence data from the training set, the Evidence Retrieval model has overfitted in finding the keywords for filtering evidence, leading to low test performance of 0.0542 in Harmonic Mean even though classification task performs well. Considering this issue, designing the method for filtering a group of text without representative/sufficient data should be proceeded more cautiously in future studies.

Furthermore, more robust model can be applied in future such as fine-tuning a BERT model. It embede of the text in a task-corpus-specific way after altering the initialised weight of each word by retraining the pre-trained BERT vectoriser with a supervised dataset, which is able to handle the specified text matching and classification tasks.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jey Han Lau. 2023a. Lecture notes on contextual representation. Lecture Notes. Accessed on May 12, 2023.

Jey Han Lau. 2023b. Lecture notes on distributional semantics. Lecture Notes. Accessed on May 12, 2023.

Jey Han Lau. 2023c. Lecture notes on question answering. Lecture Notes. Accessed on May 13, 2023.

Jey Han Lau. 2023d. Lecture notes on text classification. Lecture Notes. Accessed on May 12, 2023.

Jey Han Lau and Timothy Baldwin. 2016. An empirical evaluation of doc2vec with practical insights into document embedding generation. *arXiv preprint arXiv:1607.05368*.

P Lakshmi Prasanna and D Rajeswara Rao. 2018. Text classification using artificial neural networks. *International Journal of Engineering & Technology*, 7(1.1):603–606.

S Selva Birunda and R Kanniga Devi. 2021. A review on word embedding techniques for text classification. *Innovative Data Communication Technologies and Application: Proceedings of ICIDCA 2020*, pages 267–281.

Sandeep Tata and Jignesh M Patel. 2007. Estimating the selectivity of tf-idf based cosine similarity predicates. *ACM Sigmod Record*, 36(2):7–12.

Quan Hung Tran, Duc-Vu Tran, Tu Vu, Minh Le Nguyen, and Son Bao Pham. 2015. Jaist: Combining multiple features for answer selection in community question answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 215–219.

Muhammad Usman, Zunaira Shafique, Saba Ayub, and Kamran Malik. 2016. Urdu text classification using majority voting. *International Journal of Advanced Computer Science and Applications*, 7(8).

Wilame. 2019. Why is removing stop words not always a good idea. https://medium.com/@limavallantin/why-is-removing-\stop-words-not-always-a-good-idea-\c8d35bd77214. Accessed: May 11, 2023.

Vikas Yadav, Steven Bethard, and Mihai Surdeanu. 2020. Unsupervised alignment-based iterative evidence retrieval for multi-hop question answering. *arXiv preprint arXiv:2005.01218*.