Two Stage Transformer Model for COVID-19 Fake News Detection and Fact Checking

Rutvik Vijjali* rutvikvijjali30@gmail.com
Prathyush Potluri* potluri.prathyush@gmail.com
Siddharth Kumar* kumar.sidiyer@gmail.com
Sundeep Teki sundeep.teki@gmail.com

Abstract

The rapid advancement of technology in online communication via social media platforms has led to a prolific rise in the spread of misinformation and fake news. Fake news is especially rampant in the current COVID-19 pandemic, leading to people believing in false and potentially harmful claims and stories. Detecting fake news quickly can alleviate the spread of panic, chaos and potential health hazards. We developed a two stage automated pipeline for COVID-19 fake news detection using state of the art machine learning models for natural language processing. The first model leverages a novel fact checking algorithm that retrieves the most relevant facts concerning user claims about particular COVID-19 claims. The second model verifies the level of "truth" in the claim by computing the textual entailment between the claim and the true facts retrieved from a manually curated COVID-19 dataset. The dataset is based on a publicly available knowledge source consisting of more than 5000 COVID-19 false claims and verified explanations, a subset of which was internally annotated and cross-validated to train and evaluate our models. We evaluate a series of models based on classical text-based features to more contextual Transformer based models and observe that a model pipeline based on BERT and ALBERT for the two stages respectively yields the best results.

1 Introduction

Electronic means of communication have helped to eliminate time and distance barriers to sharing and broadcasting information. However, despite all its advantages, faster means of communication has also resulted in extensive spread of misinformation. The world is currently going through the deadly COVID-19 pandemic and fake news regarding the disease, its cures, its prevention and causes have been broadcast widely to millions of people. The spread of fake news and misinformation during such precarious times can have grave consequences leading to widespread panic and amplification of the threat of the pandemic itself. It is therefore of paramount importance to limit the spread of fake news and ensure that accurate knowledge is disseminated to the public.

In this work, we propose a robust, dynamic fake news detection system, that can not only estimate the "correctness" of a claim but also provides users with pertinent information regarding the said claim. This is achieved using a knowledge base of verified information that can be constantly updated. Previous work on fake news detection has primarily focused on evaluating the relationship measured via a textual entailment task between a header and the body of the article. However, such a method is insufficient for identifying specific fake news claims without any knowledge of the facts relevant to the claim. This warrants the use of a new dataset specific to the COVID-19 pandemic.

Developing a solution for such a task involves generating a database of factual explanations, which becomes our knowledge base, that serves as ground truth for any given claim. We compute the entailment between any given claim and explanation to verify if the claim is true or not. Querying for claim, explanation pairs for each explanation in our knowledge base is computationally expensive and slow, so

* These authors contributed equally.

Proceedings of the 3rd NLP4IF Workshop on NLP for Internet Freedom, pages 1–10
Barcelona, Spain (Online), December 12, 2020

we propose generating a set of candidate explanations which are contextually similar to the claim. We achieve this by using a model trained with relevant and irrelevant claim explanation pairs, and using a similarity metric between the two to match them. Therefore, in the pipeline, firstly, for a given claim, a set of candidate explanations from the knowledge base have to be fetched in real-time. Then, the claim validated for truth using relevant candidate explanations. In this work, we have explored the use of Transformer (Vaswani et al., 2017) based models to both fetch relevant explanations as well as measure the entailment between a given claim and a factual explanation.
We then evaluate our model on the basis of count of relevant explanation results fetched as well as the accuracy in verifying a given claim. We demonstrate the effectiveness of pre-trained multi-attention models in terms of overall accuracy when compared with other natural language processing (NLP) baselines while maintaining near real-time performance.

2 Related Work

The paper (Riedel et al., 2017) uses traditional approaches with a simple classifier model that makes use of Term Frequency(TF), Term Frequency and Inverse Document Frequency(TF-IDF), and cosine similarity between vectors as features to classify fake news. They have provided a baseline for fake news stance detection on Fake News Challenge (FNC-1) dataset[1]. We have implemented their approach on our dataset and results can be seen in Table 3.
In (Nie et al., 2019), the authors present a connected system consisting of three homogeneous neural semantic matching models that perform document retrieval, sentence selection, and claim verification on the FEVER Dataset (Thorne et al., 2018) jointly for fact extraction and verification. Their Neural Semantic Matching Network (NSMN) is a modification of the Enhanced Sequential Inference Model (ESIM) (Chen et al., 2017), where they add skip connections from input to matching layer and change output layer to only max-pool plus one affine layer with ReLU activation. They use three stage pipeline in which given a claim, they first retrieve candidate documents from the corpus, followed by retrieving candidate sentences from the selected candidate documents and finally, the last stage classifies the sentence in to one of three classes. They used Bidirectional LSTM (BiLSTM) to encode the claim and sentences using GloVe (Pennington et al., 2014) and ELMO (Peters et al., 2018) embeddings. However, these tasks are concerned with static or slowly evolving domains, on topics that do not require domain expertise to annotate.
Despite the partial success of the above methods, there were still certain shortcomings in terms of the accuracy of the results. Bidirectional encoder representations from Transformers (BERT) (Devlin et al., 2018) is a pre-trained language model trained on a large corpus comprising the Wikipedia and Toronto Book Corpus, and is shown to perform well on several natural language tasks like GLUE (Wang et al., 2018). Transformer based pretrained models achieved state of the art results in several NLP subtasks, their ease of fine-tuning makes them adaptable to newer tasks. In (Jwa et al., 2019), the authors propose a model based on the BERT architecture to detect fake news by analyzing the contextual relationship between the headline and the body text of news. They demonstrate that using Transformer based models like BERT and fine-tuning them for the task of fake news detection gives better results than other models like stackLSTM (Hanselowski et al., 2018; Hermans and Schrauwen, 2013) and featMLP (Davis and Proctor, 2017). They further enhanced their model performance by pre-training with domain specific news and articles, and countered the class imbalance for the final classification task by the use of a weighted cross entropy loss function. However, this approach can not be adapted for our task of fetching relevant explanations for each claim.
(Naudé, 2020) and (Bullock et al., 2020) extensively studied the use of machine learning strategies to address various issues regarding COVID-19. The latter also describes fake news and misinformation as a major issue in the ongoing pandemic and further highlights the problem as solving an infodemic. In (Gallotti et al., 2020), the authors develop an Infodemic Risk Index (IRI) after analyzing Twitter posts

[1] http://www.fakenewschallenge.org/

Example 1

False claim: *The title of an article suggests the low - cost steroid dexamethasone will heal anybody with COVID-19 .*
True claim: *Steroid dexamethasone is not proven cure covid-19, it has shown to help critically ill patients only.*
Explanation: *Steroid dexamethasone was found to be effective only on critically - ill COVID-19 patients that require ventilators and supplemental oxygen . There was no observed benefit on patients who did not require respiratory support .*

Example 2

False claim: *Salty and sour foods cause the " body of the COVID-19 virus " to explode and dissolve .*
True claim: *Salty and sour foods do not help prevent or cure COVID-19.*
Explanation: *Consuming fruit juices or gargling with warm water and salt does not protect or kill COVID-19 , the World Health Organization Philippines told VERA Files .*

Figure 1: Cross validated data examples

across various languages and calculate the rate at which a particular user from a locality comes across unreliable posts from different classes of users like verified humans, unverified humans, verified bots, and unverified bots. In (Mejova et al., 2018), the authors examine Facebook advertisements across 64 countries and find that around 5% of advertisements contained possible errors or misinformation. But none of these mentioned works tackle the problem of misinformation by reasoning out the given fake claim with an explanation.

3 Dataset

Using an existing misinformation dataset will not serve as a reliable knowledge base for training and evaluating the models due to the recent and uncommon nature i.e., the vocabulary used to describe the disease and the terms associated with the COVID-19 pandemic. It is important to generate real and timely datasets to ensure accurate and consistent evaluation of the methods. To overcome this drawback, we manually curated a dataset specific to COVID-19. Our proposed dataset consists of 5500 claim and explanation pairs. We describe the collection and annotation process in Section 3.1.

3.1 Covid-19 Claims Dataset

There are multiple sources on the web that are regularly identifying and debunking fake news on COVID-19. We scraped data from "Poynter"[2] , a fact checking website which collects fake news and debunks or fact-checks them with supporting articles from more than 70 countries, covering more than 40 languages. The Poynter website has a database exclusively for COVID-19 with over 7000 fact checks. Each fact check contains the corresponding claim that is being checked, the rating indicating the type of the claim, for example - 'False', 'Mostly False' or 'Misleading', the name of the fact checker, the explanation given for the current claim, the location of origin of the claim and the date when the fact check was done. This data can be used to update our "explanation" look-ups in a timely fashion so our database is constantly evolving as we learn more about the virus and the facts change.

For each fact check, we collect only the "claim" and the corresponding "explanation" from this database which were rated as 'False' or 'Misleading'. In this way, we collected about 5500 false-claim and explanation pairs. We further manually rephrase these false claims to generate true claims, as the ones that align with the explanation so as to create an equal proportion of true-claim and explanation pairs. We have taken claim-explanation pairs from the time period between January 1,2020 to May

[2]https://www.poynter.org/ifcn-covid-19-misinformation/

Dataset	Number of sentence pairs
False claim - Explanation pairs	5500
Cross validated False claim - True claim - Explanation pairs for train data	1000
Cross validated False claim - True claim - Explanation pairs for test data	200

Table 1: Dataset Information

15,2020 for our training data and from May 18,2020 to July 1,2020 for our test data. In this way, we evaluate the generalization of the model on completely new and unseen data. Our collected data follows the structure:

[false claim, explanation]

The subset of the data that we annotated and cross validated follows the structure:

[false claim, true claim, explanation]

Figure 1 shows some examples of the cross validated data.

3.2 Dataset Statistics

The current proposed Covid-19 dataset contains cross-validated claim-explanation sentence pairs. Statistics about the distribution of labels are provided in Table 1. This is a dynamic dataset and we are continually collecting and curating additional claim-explanation pairs. We plan to open source this dataset to facilitate more research in this domain.

4 Methodology

The architecture consists of a two stage model, we will refer to the first model as "Model A" and the second model as "Model B". The objective of Model A is to fetch the candidate "true facts" or explanations for a given claim, which are then evaluated for entailment using the Model B. Next, we describe the training procedure as well as intended run time behaviour for both Model A and Model B.

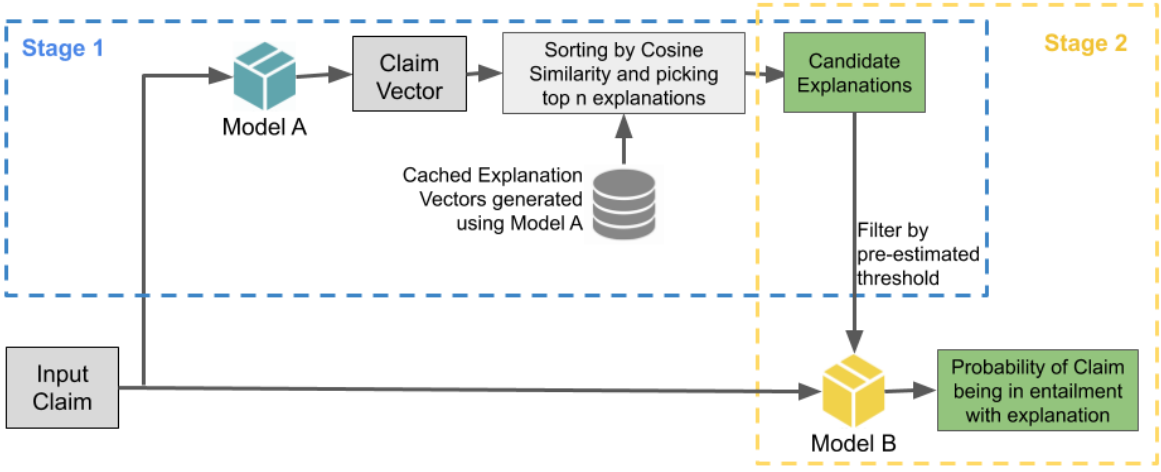

Figure 2: Block diagram of our two stage model pipeline

4.1 Model A

First, to fetch relevant explanations, we train our Transformer model on a binary sentence entailment task, where the claims and explanations are the two sentences fed in as input separated by a [SEP] tag. We generate negative claim-explanation pairs through random sampling to ensure that equal proportions of positive and negative pairs are present. Training multi-attention network with our COVID-19 specific data enables the model to capture long-range correlations between the vector representations of claims

and explanations of similar contexts. We train our models with a base encoder and a sequence classification head on top for binary classification of the labels. The model is trained to optimise the cross entropy loss.

Through our experiments, we find that, on this trained model, if we generate embeddings for a single sentence (either claim or explanation individually) and compare matching [claim, explanation] embeddings using the cosine similarity metric, there is a distinction in the distribution of similarity scores between related and unrelated [claim,explanation] pairs. Therefore, for faster near real-time performance, we cache the embeddings for all our explanations (knowledge base) beforehand, and compute the cosine similarity between the claim and the cached embeddings of the explanations. The vector of the [CLS](the start of sentence) token of the final layer works as a strong representation of the entire sentence, although we found that taking element-wise mean over all the token vectors leads to better performance. We fetch the top explanations for any given claim exceeding a certain threshold of sentence similarity as there could be several explanations relevant for a given claim. This threshold is determined on the basis of the summary statistics of the cosine similarity metric between the claim and relevant explanations in the validation set, as described in Section 5.3. These retrieved explanations serve as candidates for verifying the accuracy of the claim through Model B.

4.2 Model B

The second part of the pipeline is to identify the veracity of a given claim. Model A fetches the candidate explanations while Model B is used to verify whether the given claim aligns with our set of candidate explanations or not. We can therefore treat this task as a textual entailment problem (Dagan et al., 2013; Adler et al., 2012). To train the Model B, we use a smaller subset of "false claim" and "explanation" pairs from our original dataset, and cross validate each sample with "true claim" or in other words, claims that align with the factual explanation. However, this small annotated data is not sufficient to train the model effectively. Therefore, the parameters of the Model A, which was trained on a much larger dataset were used as initial parameters for Model B, and fine-tuned further using our cross validated dataset.

We trained Model B in a similar fashion as Model A i.e. as a sequence classification problem with cross entropy loss. Once we have the candidate explanations for a given claim, we use Model B to estimate the probabilities of alignment of claim with each of the candidate explanations. We used the statistic of mean probability score and standard deviation of aligning and non-aligning claim and explanation pairs in the validation set to determine the thresholds for Model B classification. We trained and evaluated both Model A and Model B using several approaches based on classical NLP methods as well as more sophisticated pre-trained Transformer models. The flow of the Model A + Model B pipeline is shown in Figure 2.

5 Experiments

5.1 Baseline Models

For baselining our model on classical NLP approaches, we use two simple bag-of-words (BOW) representations for the text inputs: term frequency (TF) and term frequency-inverse document frequency (TF-IDF). We followed the architecture proposed by (Riedel et al., 2017). The representations and features extracted from the claim and explanation pairs consist of the following: • The TF vector of the claim; • The TF vector of the explanation; • The cosine similarity between the 2-normalised TF-IDF vectors of the claim and explanation.

For the TF vectors, we extract a vocabulary of the 5,000 most frequent words in the training set and exclude stop words (the NLTK (Bird et al., 2009) stop words for the English language). For the TF-IDF vectors, we use the same vocabulary and set of stop words. The TF vectors and the TF-IDF cosine similarity values are concatenated to form a feature vector with total dimension of 10,001 and is fed to the classifier.

The classifier takes an input of 10,001 dimensional vector followed by a feed-forward network with 50, 20 and 2 dimensional dense layers respectively with each hidden layer having tanh activation and the last layer is a softmax layer. We trained in mini-batches over the entire training set using the Adam

Model	Model A Val Accuracy	Model B Val Accuracy
TF-IDF	0.832	0.799
GloVe	0.781	0.777
MobileBERT	0.921	0.877
ALBERT	0.927	**0.956**
BERT	**0.944**	0.927

Table 2: Model Performance on Validation Set

optimiser (Kingma and Ba, 2014) with a learning rate of 0.001.
The second approach involves use of word vectors for which we used 300-dimensional GloVe (Pennington et al., 2014) embeddings, pretrained on 2014-Wikipedia and Gigaword, and averaged over token embeddings to compute sentence vectors. So, for a given claim and explanation pair, we have a 300-dimension vector for claim as well as the explanation, both of which are concatenated to form a 600-dimensional vector that serves as input to our dense layer classifier. This model is a simple feed-forward neural network with 4 hidden layers having 200, 100, 50 and 2 hidden units respectively with the first three layers having ReLU activation while the last layer is a softmax layer. We trained in mini-batches of 32 over the entire training set with back-propagation using the Adam optimizer with a learning rate of 0.001.

5.2 Transformer Models

We trained and evaluated three Transformer based pre-trained models for both Model A and Model B using the training strategy described in Section 4. As our focus was to ensure that the proposed pipeline can be deployed effectively in a near real-time scenario, we restricted our experiments to models that could efficiently be deployed using inexpensive compute. We chose the following three models - BERT(base), ALBERT (Lan et al., 2019) and MobileBERT (Sun et al., 2020). The authors of MobileBERT demonstrated that using a teacher-student learning technique for progressive knowledge transfer from the BERT to MobileBERT model helps them achieve a task-agnostic model similar to BERT and can be deployed easily on resource limited devices due to faster inference speeds and lower memory consumption. The ALBERT model was proposed to increase the training and inference speed of BERT besides lowering the memory consumption. The authors demonstrate that the use of their parametric reduction techniques and a custom self supervised loss helps it to achieve results similar to BERT while having fewer parameters. Model A was trained on 5000 claim-explanation pairs on the sequence classification task to optimize the softmax cross entropy loss using a learning rate of 3e-5. This trained model was then validated on a test set comprising of 1000 unseen claim-explanation pairs. The training data structure here looks like:

[claim, relevant explanation, 1]

[claim, irrelevant explanation, 0]

Model B was trained on a smaller subset of 800 cross validated [claim,explanation,label] data, on the same sequence classification task, where the label was assigned based on whether the claim aligned with the explanation - 1 or not - 0. This was validated on 200 unseen data-points. The loss function used was softmax cross-entropy with a uniform learning rate of 1e-5. The training data structure here looks like:

[true claim, relevant explanation, 1]

[false claim, relevant explanation, 0]

5.3 Evaluation Metrics

For evaluating the performance of the overall pipeline model, we first evaluate the performance of Model A in its ability to retrieve relevant explanation. For this we use Mean Reciprocal Rank(MRR) (Craswell, 2009) and Mean Recall @10 (Malheiros et al., 2012), that is the proportion of claims for which the relevant explanation was present in the top 10 most contextual explanation by cosine similarity and their

mean inverse rank. Equation 1 shows the MRR formula for our evaluation.

$$MRR = \frac{1}{C}\sum_{i=1}^{C}\frac{1}{rank_i} \tag{1}$$

where $rank_i$ is the position of the explanation that is relevant to a particular claim according to test data and C is the total number of claims.
Equation 2 shows the Recall@10 formula for our evaluation.

$$Recall@10 = \frac{1}{C}\sum_{i=1}^{C} f(\text{top-10 explanations}) \tag{2}$$

$$where$$

$$f(\text{x}) = \begin{cases} 1 & \text{if } true\_exp \in \text{x} \\ 0 & \text{otherwise} \end{cases}$$

Here, $true\_exp$ is the actual relevant explanation for a particular claim according to test data and C is the total number of claims.
Once, Model A has retrieved relevant explanations, we evaluate the performance of Model B on computing the veracity of the claim. Here, we only used explanations that exceed an empirically defined threshold in cosine similarity between the claim and the explanation. Through our experiments, we found that a threshold of *mean - standard deviation of cosine similarity* over the validation data worked well for picking relevant explanations. For evaluating the accuracy, we take a mean of the output probabilities for each $claim, explanation_i$, defined by the Equation 3.

$$p_{truth} = \frac{1}{n}\sum_{i} modelB(claim, exp_i) \tag{3}$$

$$where$$
$$exp \in exp_A \mid \cos(c\_vec_A, exp\_vec_A) > t$$

Here, exp_A are the top-10 explanations returned by Model A. c_vec_A and exp_vec_A are the vector representations generated by running claim and explanations individually through Model A and t refers to the threshold.

5.4 Results and Discussion

The performance of several Transformer based models as well as classical NLP models were compared using the evaluation metrics described in Section 5.3. The results of the experiments on the test set are summarised in Table 3.

Model	MRR	Recall@10	Accuracy
TF-IDF	0.477	0.635	0.525
GloVe	0.182	0.410	0.579
MobileBERT	0.561	0.735	0.710
BERT	0.632	0.795	0.810
ALBERT	0.582	0.675	0.825
BERT+ALBERT	0.632	0.795	0.855

Table 3: Model Performance on test set

The results in Table 3 clearly illustrate that Transformer based models are significantly better than classical NLP models. An interesting observation was that some models are better at retrieval of relevant

Model	Latency per claim (in seconds)	Memory (in MB)
TF-IDF	0.108	16
GloVe	0.003	990
MobileBERT	0.607	1200
ALBERT	2.376	942
BERT	3.106	1910
BERT + ALBERT	2.471	1398

Table 4: Model Compute performance and Memory usage

explanations while others have a better classification performance. We find that a combination of the best performing Model A (BERT) and best performing Model B (ALBERT) yielded the highest MRR, Recall@10 and Accuracy on the test set for fact checking. We however do acknowledge that our models could still make errors of two kinds: firstly, Model A might not fetch a relevant explanation which automatically means that the prediction provided by Model B is irrelevant, and secondly, Model A might have fetched the correct explanation(s) but Model B classifies it incorrectly. We show some of the errors our models made in Table 5.

Table 4 shows the memory usage and latencies of the implemented models. The memory consumption and latency per claim in the classical NLP models was observed to be quite low in comparison to the Transformer based models. This is expected due to the lower parameter size of the TF-IDF and GloVe models. Among the Transformer based models, MobileBERT had the least latency per claim as expected and explained in Section 5.2 while ALBERT consumed the least memory. The best performing BERT+ALBERT model utilized a memory of 1398MB and fetched relevant explanations of each claim in 2.471 seconds. The model latencies and memory usage were evaluated on an *Intel Xeon - 2.3GHz Single core - 2 thread* CPU.

Claim: *Cannabis could help prevent coronavirus infection .* **True Explanation**: *The study claiming that marijuana can cure coronavirus did not pass the peer review , it was conducted on artificial human tissues and not on real organisms . It is a classic preliminary research that may even fail . The authors themselves speak of the need for further studies and research .* **Top Fetched Explanation**: *The vaccine can provide stable immunity , while the presence of antibodies does not prevent reinfection .* **Remark**: *Model A fetched irrelevant explanation for this claim*
Claim: *The vaccine is not the final solution against the novel coronavirus but antibodies are .* **Explanation**: *The vaccine can provide stable immunity , while the presence of antibodies does not prevent reinfection .* **Probability score**: *0.340* **Remark**: *Model B misclassified this claim-explanation pair as True*
Claim: *WHO recommends wearing masks in public spaces to slow down the spread of coronavirus* **Explanation**: *The WHO changed its position about masks by now recommending community masks in areas with many infections . And it says that masks have to be used properly and alone can't protect you from COVID-19 .* **Probability score**: *0.686* **Remark**: *Model B misclassified this claim-explanation pair as False*

Table 5: BERT+ALBERT Model Faulty predictions

6 Conclusions and Future work

In this work, we have demonstrated the use and effectiveness of pre-trained Transformer based language models in retrieving and classifying fake news in a highly specialized domain of COVID-19. Our proposed two stage model performs significantly better than other baseline NLP approaches. Our knowledge base, that we prepare through collecting factual data from reliable sources from the web can be dynamic and change to a large extent, without having to retrain our models again for as long as the distribution is consistent. All of our proposed models can run in near real-time with moderately inexpensive compute.
Our work is based on the assumption that our knowledge base is accurate and timely. This assumption might not always be true in a scenario such as COVID-19 where "facts" are changing as we learn more about the virus and its effects. Therefore a more systematic approach is needed for retrieving and classifying claims using this dynamic knowledge base. Our future work consists of weighting our knowledge base on the basis of the duration of the claims and benchmarking each claim against novel sources of ground truth.
Our model performance can be further boosted by better pre-training, through domain specific knowledge. In one of the more recent work by (Guo et al., 2020), the authors propose a novel semantic textual similarity dataset specific to COVID-19. Pre-training our models using such specific datasets could help in better understanding of the domain and ultimately better performance.
Fake news and misinformation is an increasingly important and a difficult problem to solve, especially in an unforeseen situation like the COVID-19 pandemic. Leveraging state of the art machine learning and deep learning algorithms along with preparation and curation of novel datasets can help address the challenge of fake news related to COVID-19 and other public health crises.

References

Meni Adler, Jonathan Berant, and Ido Dagan. 2012. Entailment-based text exploration with application to the health-care domain. In *Proceedings of the ACL 2012 System Demonstrations*, pages 79–84.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Joseph Bullock, Katherine Hoffmann Pham, Cynthia Sin Nga Lam, Miguel Luengo-Oroz, et al. 2020. Mapping the landscape of artificial intelligence applications against covid-19. *arXiv preprint arXiv:2003.11336*.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Diana Inkpen, and Si Wei. 2017. Neural natural language inference models enhanced with external knowledge. *arXiv preprint arXiv:1711.04289*.

Nick Craswell. 2009. Mean reciprocal rank. *Encyclopedia of database systems*, 1703.

Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto. 2013. Recognizing textual entailment: Models and applications. *Synthesis Lectures on Human Language Technologies*, 6(4):1–220.

Richard Davis and Chris Proctor. 2017. Fake news, real consequences: Recruiting neural networks for the fight against fake news.

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*.

Riccardo Gallotti, Francesco Valle, Nicola Castaldo, Pierluigi Sacco, and Manlio De Domenico. 2020. Assessing the risks of" infodemics" in response to covid-19 epidemics. *arXiv preprint arXiv:2004.03997*.

Xiao Guo, Hengameh Mirzaalian, Ekraam Sabir, Aysush Jaiswal, and Wael Abd-Almageed. 2020. Cord19sts: Covid-19 semantic textual similarity dataset. *arXiv preprint arXiv:2007.02461*.

Andreas Hanselowski, Avinesh PVS, Benjamin Schiller, Felix Caspelherr, Debanjan Chaudhuri, Christian M Meyer, and Iryna Gurevych. 2018. A retrospective analysis of the fake news challenge stance detection task. *arXiv preprint arXiv:1806.05180*.

Michiel Hermans and Benjamin Schrauwen. 2013. Training and analysing deep recurrent neural networks. In *Advances in neural information processing systems*, pages 190–198.

Heejung Jwa, Dongsuk Oh, Kinam Park, Jang Mook Kang, and Heuiseok Lim. 2019. exbake: Automatic fake news detection model based on bidirectional encoder representations from transformers (bert). *Applied Sciences*, 9(19):4062.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Yuri Malheiros, Alan Moraes, Cleyton Trindade, and Silvio Meira. 2012. A source code recommender system to support newcomers. In *2012 IEEE 36th Annual Computer Software and Applications Conference*, pages 19–24. IEEE.

Yelena Mejova, Ingmar Weber, and Luis Fernandez-Luque. 2018. Online health monitoring using facebook advertisement audience estimates in the united states: evaluation study. *JMIR public health and surveillance*, 4(1):e30.

Wim Naudé. 2020. Artificial intelligence against covid-19: An early review.

Yixin Nie, Haonan Chen, and Mohit Bansal. 2019. Combining fact extraction and verification with neural semantic matching networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6859–6866.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Benjamin Riedel, Isabelle Augenstein, Georgios P Spithourakis, and Sebastian Riedel. 2017. A simple but tough-to-beat baseline for the fake news challenge stance detection task. *arXiv preprint arXiv:1707.03264*.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. Mobilebert: a compact task-agnostic bert for resource-limited devices. *arXiv preprint arXiv:2004.02984*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.