Evaluating Deep Learning Approaches for Covid19 Fake News Detection

Apurva Wani[1], Isha Joshi[1], Snehal Khandve[1], Vedangi Wagh[1], and Raviraj Joshi[2]

[1] Pune Institute of Computer Technology, Pune
[2] Indian Institute of Technology Madras, Chennai
apurva.wani06, ishajoshi.211, snehal.khandve07, vedangikwagh, ravirajoshi@gmail.com

Abstract. Social media platforms like Facebook, Twitter, and Instagram have enabled connection and communication on a large scale. It has revolutionized the rate at which information is shared and enhanced its reach. However, another side of the coin dictates an alarming story. These platforms have led to an increase in the creation and spread of fake news. The fake news has not only influenced people in the wrong direction but also claimed human lives. During these critical times of the Covid19 pandemic, it is easy to mislead people and make them believe in fatal information. Therefore it is important to curb fake news at source and prevent it from spreading to a larger audience. We look at automated techniques for fake news detection from a data mining perspective. We evaluate different supervised text classification algorithms on Contraint@AAAI 2021 Covid-19 Fake news detection dataset. The classification algorithms are based on Convolutional Neural Networks (CNN), Long Short Term Memory (LSTM), and Bidirectional Encoder Representations from Transformers (BERT). We also evaluate the importance of unsupervised learning in the form of language model pre-training and distributed word representations using unlabelled covid tweets corpus. We report the best accuracy of 98.41% on the Covid-19 Fake news detection dataset.

Keywords: fake news, convolutional neural networks, long short term memory, transformers, language model pretraining

1 Introduction

Technology has been dominating our lives for the past few decades. It has changed the way we communicate and share information. The sharing of information is no longer constrained by physical boundaries. It is easy to share information across the globe in the form of text, audio, and video. An integral part of this capability is the social media platforms. These platforms help in sharing personal opinions and information with much a wider audience. They have taken over traditional media platforms because of speed and focussed content.

However, it has become equivalently easy for nefarious people with malicious intent to spread fake news on social media platforms.

Fake news is defined as a verifiably false piece of information shared intentionally to mislead the readers [22]. It has been used to create a political, social, and economic bias in the minds of people for personal gains. It aims at exploiting and influencing people by creating fake content that sounds legit. On the extreme end, fake news has even led to cases of mob lynching and riots [6]. Thus, it is extremely important to stop the spread of fake content on internet platforms. It is especially desirable to control fake news during the ongoing Covid-19 crisis [25]. The pandemic has made it easy to manipulate a mentally stranded population eagerly waiting for this phase to end. Some people have reportedly committed suicide after being diagnosed with covid due to the misrepresentation of covid in social and even mainstream media [2]. The promotion of false practices will only aggravate the covid situation.

Recently, researchers have been actively working on the task of fake news detection. While manual detection [8,1,7] is the most reliable method it has limitations in terms of speed. It is difficult to manually verify the large volumes of content generated on the internet. Therefore automatic detection of fake news has gained importance. Machine learning algorithms have been employed to analyze the content on social media for its authenticity [27]. These algorithms mostly rely on the content of the news. The user characteristics, the social network of the user, and the polarity of their content are another set of important signals [31]. It is also common to analyze user behavior on social platforms and assign them a reliability score. The fake news peddlers might not exhibit normal sharing behavior and will also tend to share more extreme content. All these features taken together provide a more reliable estimate of authenticity.

In this work, we are specifically concerned with fake news detection related to covid. The paper describes systems evaluated for Contraint@AAAI 2021 Covid-19 Fake news detection shared task [19]. The task aims in improving the classification of the news based on Covid-19 as fake or real. The dataset shared is created by collecting data from various social media sources such as Instagram, Facebook, Twitter, etc.

The fake news detection task is formulated as a text classification problem. We solely rely on the content of the news and ignore other important features like user characteristics, social circle, etc which might not always be available. We evaluate the recent advancements in deep learning based text classification algorithms for the task of fake news detection. The techniques include pre-trained models based on BERT and raw models based on CNN and LSTM. We also evaluate the effect of using monolingual corpus related to covid for language model pretraining and training word embeddings. In essence, we rely on these models to automatically capture discriminative linguistic, style, and polarity features from news text helpful for determining authenticity.

2 Related Works

Fake news detection on traditional outlets of news and articles solely depends on the reader's knowledge about the subject and the article content. But detection of fake news that has been transmitted via social media has various cues that could be taken into consideration. One of the cues can be finding a user's credibility by analyzing their followers, the number of followers, and their behavior as well as their registration details. In addition to these details [10] have used other factors such as attached URLs, social media post propagation features, and content-based hybrid model for classifying news as fake or genuine. Another research [16] based on structural properties of the social network is used for defining a "diffusion network" which is the spread of a particular topic. This diffusion network together with other social network features can be helpful in the classification of rumors in social media with classifiers like SVM, random forest, or decision tree.

Besides using the characteristics of user-patterns and user details who share fake news, another context useful for classifying any social media news post is the comments section. [32] have performed a linguistic study and have found comments like "Really?", "Is it true?" in the comment section of some of the fake posts. They have further implemented a system that clusters such inquiry phrases in addition to clustering simple phrases for classifying rumors.

Another approach of considering the tri-relationship between the publishers, the news articles, and users of fake news can be considered. This relationship has been used to create a tri-relationship embedding framework TriFN in [23] for detection of fake news articles on social media. Four types of embeddings namely the news content embedding, user embedding, user-news interaction embeddings as well as publisher-news relation embeddings with contributions to spread fake news are generated coupled with a semi-supervised classifier are used in TriFN to identify fake news.

The propagation knowledge of fake news articles such as its path construction and transformation can also be useful for primary detection of fake news [17]. Further, this transformation path has been represented into vectors for classification with deep neural network architectures namely the RNN for global variation and CNN for local variation of the path.

Apart from user-context and social-context features the content of the fake news has also been a proven way of detecting fake news and rumors. A recent approach utilizes explicit as well as latent features of the textual information for further classification of news [30]. Basic Deep Convolutional neural networks have also been used to get contextual information features from fake news articles for identifying them [14].

3 Architecture Details

In this section, we describe the techniques we have used for text classification. We also describe the hyper-parameters used in each of these models. The model

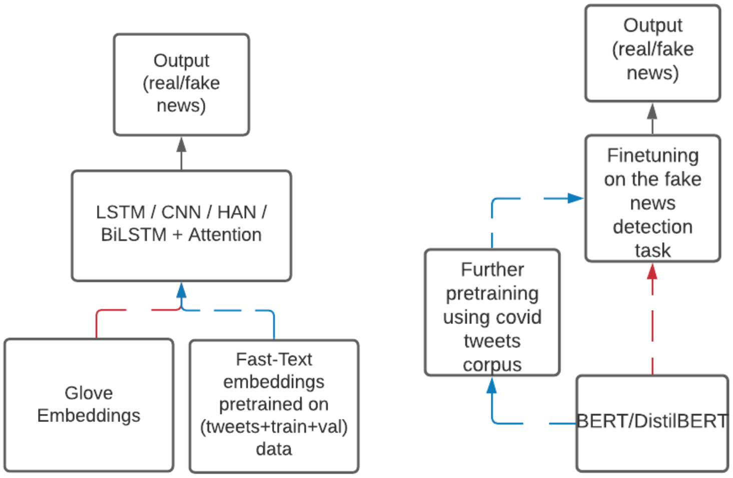
Output (real/fake news)
LSTM / CNN / HAN / BiLSTM + Attention
Glove Embeddings
Fast-Text embeddings pretrained on (tweets+train+val) data
Output (real/fake news)
Finetuning on the fake news detection task
Further pretraining using covid tweets corpus
BERT/DistilBERT

Fig. 1. Model summary showing two approaches of Simple models(L) and two approaches of Transformer based models(R) using different colours.

summary is shown in Fig. 1 for the two types of architectures explored in this work.

3.1 CNN

Although CNN is mostly used for image recognition tasks, text classification is also recognized as one of the applications of CNN [15]. The CNN layers extract useful features from the word embeddings to generate the output. The 300-dimensional fast text embeddings are used as input to the first layer. We use a slightly deep architecture with initial five parallel 1D Conv layers. The kernel size for these parallel convolutions is size 2,3,4,5,6. The number of filters used in these conv layers is 128. The output of these conv layers are concatenated and then fed to two sequential blocks of 1D conv layer followed by 1D MaxPooling layer. Three dense layers of sizes 1024, 512, and 2 are subsequently added to the entire architecture. There is a dropout of 0.5 added after the final two conv layers and the first two dense layers. This CNN model is trained on a batch size of 64 samples and an Adam optimizer is used. The batch size and optimizer are constant for all non-BERT models.

3.2 LSTM

Long Short-Term Memory (LSTM) is a type of Gated-RNN architecture along with the feedback connections [12]. With the input length equal to the length of the longest tweets in train data, the embedding layer is the first layer. It is followed by a single LSTM layer with 128 units, a single dropout layer with a dropout rate of 0.5, and two dense layers with units 128 and 2 respectively.

3.3 Bi-LSTM + Attention

The additional feature that the Bi-LSTM network offers is that it considers the input sequence from both the forward and reverse direction. This sequential model has a first embedding layer similar to the previous models. The next layer is a bidirectional LSTM with 256 units in each direction followed by an attention layer and two dense layers with 128 and 2 units. The structure of the attention layer is borrowed from [33].

3.4 HAN

Hierarchical Attention Networks (HAN) is based on LSTM and comprises of four sequential levels - word encoder, word-level attention, sentence encoder, and sentence-level attention [29]. Each data sample is divided into a maximum of 40 sentences and each sentence consists of a maximum of 50 words. The word encoder is a bidirectional LSTM that works on word embeddings of individual sentences to produce hidden representation for each word. The word-level attention helps us to extract important words that contribute to the meaning of the sentence. These informative words which conceive the complete meaning of the sentence are aggregated to form sentence vectors. The sentence vectors are processed by another bidirectional LSTM referred to as sentence encoder. The sentence-level attention layer measures the importance of each sentence and sentences which provide the most significant information for classification are summarized to get a document vector that contains the gist of the entire data sample.

3.5 Transformers

Transformers have outperformed previous sequential models in various NLP tasks [26]. The major component of transformers is self-attention which is a variant of the attention mechanism [9]. Self-attention is used for generating a contextual embedding of any given word in the input sentence with respect to other words in the sentence. The major advantage of transformers over RNNs [24] was that it led to parallelization of the process which made it possible to take advantage of the contemporary hardware.

The Transformer architecture consists of an encoder and a decoder. Transformer blocks consisting of a self-attention layer and a feed-forward neural network are stacked on top of one another where the output of one is passed as input to the next one. In the first layer, the words in the input text are converted to embeddings and positional encoding is added to these embeddings in order to add information about the word's position. The word embeddings generated from the first block are passed to the next block as input. The final encoder generates an embedding for each word in the text. The original transformer architecture consists of a decoder stack which is used for machine translation. However, that is not required for classification tasks as we are only interested in classifying the input text using the embeddings generated by the encoder stack. We used two transformer-based architectures to adapt to the classification task.

BERT. BERT-base [11] is a model that contains 12 transformer blocks, 12 self-attention heads, and a hidden size of 786. The input for BERT contains embeddings for a maximum of 512 words and it outputs a representation for this sequence. The first token of the sequence is always [CLS] which contains the special classification embedding and another special token [SEP] is used for separating segments for other NLP tasks. For the purpose of a classification task, the hidden state of the [CLS] token from the final encoder is considered and a simple softmax classifier is added on top to classify the representation.

DistilBERT. DistilBERT [21] offers a simpler, cheaper, and lighter solution that has the basic transformer architecture similar to that of BERT. Instead of distillation during the fine-tuning phase specific to the task, here the distillation is done during the pre-training phase itself. The number of layers is halved and algebraic operations are optimized. Using a few such changes, DistilBERT provides competitive results even though it is 40% smaller than BERT.

4 Experimental setup

4.1 Dataset details

The Contraint@AAAI 2021 Covid-19 Fake news detection dataset[19] consists of tweets and their corresponding label. The label categorizes tweets as either fake or real. The dataset has a predefined train, test, and validation split. The train data has 6420 samples, test data has 2140 samples and validation data has 2140 samples; making it a total of 10,700 media articles and posts acquired from multiple platforms. Train data contains 3060 fake samples and 3360 real samples while validation and test data contain 1020 fake samples and 1120 real samples each. The fake tweets were collected from fact-checking websites like Politifact, NewsChecker, Boomlive [8,7,1], and from tools like Google fact-check-explorer and IFCN chatbot [5]. For obtaining real tweets verified Twitter handles were used.

After performing the pre-processing steps mentioned in section 4.2 statistics of the dataset are shown in Table 1. It is also observed that 2998 unique tokens from the test data are absent in the training dataset. Similarly, for the validation dataset, 2888 tokens are absent in the train dataset.

Models like BERT are trained on huge text datasets like Wikipedia which comprise of text from a variety of domains. However, re-training such models on the corpus related to the domain under consideration might make the model adapt to a specific domain better. With this aim, an unlabelled corpus of covid

Table 1. Statistics of the dataset.

Feature	Train data	Test data	Validation data
Total words	115244	39056	38021
Total unique tokens	14264	7151	6927
Maximum length of a tweet (in words)	871	968	209
Average length of a tweet (in words)	17.95	18.25	17.76

tweets with the hashtag covid19 was gathered using Twitter API [3]. This corpus was used for further pretraining in BERT and Fast-Text related experiments reported in this paper.

4.2 Preprocessing of the dataset

Following steps of preprocessing are used for sequential models :

- **Removal of HTML tags**: Often in the process of gathering dataset, web or screen scraping leads to the inclusion of HTML tags in the text. These tags are often not paid heed to but it is necessary to get rid of them.
- **Convert Accented Characters to ASCII characters**: To avoid the NLP model from treating accented words like ”résumé”, ”latté”, etc different from their standard spellings, the text has to be passed through this step.
- **Expand Contractions**: Apostrophe is commonly used to shorten the entire word or a group of words. For example, ”don’t means ”do not” and ”it’s” stands for ”it is”. These shortened forms are expanded in this step.
- **Removal of Special Characters**: Special characters are not readable because they are neither alphabets nor numbers. They include characters like ”*”, ”&”, ”$”, etc.
- **Noise Removal**: Noisy text includes unnecessary new lines, white spaces, etc. Filtering of such text is done in this process.
- **Normalization**: The entire text is converted into lowercase characters due to the case sensitive nature of NLP libraries.
- **Removal of stop-words**: English language stop words include words like ‘a’, ‘an’, ‘the’, ‘of’, ‘is’, etc which commonly occur in sentences and usually add less value to the overall meaning of the sentence. To ensure less processing time it is better to remove these stop words and let the model focus on the words that convey the main focus of the sentence.
- **Stemming**: This step reduces the word to its root word after removing the suffixes. But it does not ensure that the resulting word is meaningful. Among many available stemming algorithms, the one used for this paper is Porter’s Stemmer algorithm.

Sequential models were trained using two types of word embeddings namely Glove and Fast-text.

- 100 dimensional pre-trained Glove [20] embeddings
- 300 dimensional Fast-text [13] embeddings which were generated by training on a joint corpus of train data, validation data specific to this task and covid19 corpus [3] of tweets.

The embedding layer is kept trainable and connected to the first layer of the respective network.

4.3 Training Details

All the models were trained using the Tensorflow 2.0 framework. All models were trained for a maximum of 10 epochs and validation loss was used to pick the best epoch.

Transformer-based architectures. The transformer-based models BERT and DistilBERT are used in two different ways:

Fine-tuning strategies: BERT and DistilBERT models which are pre-trained on a general corpus can be used for different classification and generation tasks. We have fine-tuned these two models in order to adapt to the target classification task. Along with this, we have also used two publicly shared BERT-based models pretrained on covid corpus from the huggingface model hub.

- Covid-bert-base : Covid-bert-base [4] is a pretrained model from huggingface which is trained on a covid-19 corpus using the BERT architecture.

- Covid-Twitter-Bert : Covid-Twitter-Bert [18] is pretrained using a large corpus of covid-19 twitter messages on BERT architecture. This model is used from huggingface pretrained models [28] and fine-tuned on the target dataset.

Further pretraining: The pre-trained models of BERT and DistilBERT are based on a general domain corpus from the pre-covid era. They can be further trained on a corpus related to the domain of interest. In this case, we used an accumulated collection of tweets[3] with the hashtag covid19. These models were trained as a language model on the corpus of COVID-19 tweets which is also the target domain. This pre-trained language model was then used as a classification model in order to adapt to the target task. We manually pre-trained BERT and DistilBert models on a covid tweets dataset using huggingface library.

5 Results and Discussion

We analyze the accuracies reported using different types of models on the target dataset in Table 2. The baseline accuracy refers to the best accuracy reported in [19] using SVM model. The BERT and DistilBERT models pretrained on the Covid-19 tweets corpus perform better than the ones which are only fine-tuned on the dataset. The bert-cased model which was trained manually on the

Table 2. Results using five strategies

Strategies	**Model**	**Accuracy**	
		Validation	**Testing**
Finetuning	bert-cased	97.94	98.08
	distilbert-cased	97.94	97.75
	bert-uncased	**98.13**	97.71
	distilbert-uncased	97.94	98.22
	covid-base-bert	97.05	97.05
	covid-twitter-bert	98.22	**98.36**
LM Pretraining	bert-cased	98.04	**98.41**
	distilbert-cased	98.13	98.22
	distilbert-uncased	97.99	98.04
	bert-uncased	**98.27**	98.17
Fast-text	CNN	91.64	94
	LSTM	93.6	94.95
	BiLSTM + Attention	92.71	94.71
	HAN	**95.42**	**95**
GloVe	CNN	93	93.50
	LSTM	92.52	92.62
	BiLSTM + Attention	**94.39**	92.99
	HAN	94.16	**94.25**
Baseline	–	93.46	93.32

covid-19 tweets corpus gives the best results followed by the Covid-Twitter-Bert model. Among the non-transformer models, HAN gives the best results. Overall, the transformer models both pre-trained and fine-tuned, perform much better than the non-transformer models word-based models. The fast text word vectors were trained on target corpus and hence perform slightly better than pre-trained GloVe embeddings. This shows the importance of pre-training on target domain like corpus.

6 Conclusion

Under the shared task of Contraint@AAAI 2021 Covid-19 Fake news detection, we analyzed the efficacy of various deep learning models. We performed thorough experiments on transformer-based models and sequential models. Our experiments involved further pretraining using a covid-19 corpus and fine-tuning the transformer-based models. We show that manually pretraining the model on a subject-related corpus and then adapting the model to the specific task gives the best accuracy. The transformer-based models outperform other basic models with an absolute difference of 3-4% in accuracy. We achieved a maximum accuracy of 98.41% using language model pretraining on BERT over the baseline accuracy of 93.32%. Primarily we demonstrate the importance of pre-training on target domain like corpus.

Acknowledgements

This research was conducted under the guidance of L3Cube, Pune. We would like to express our gratitude towards our mentors at L3Cube for their continuous support and encouragement. We would also like to thank the competition organizers for providing us an opportunity to explore the domain.

References

1. Boom: Coronavirus news, fact checks on fake and viral news, online news updates. https://www.boomlive.in/, (Accessed on 12/25/2020)
2. Coronavirus: Indian man 'died by suicide' after becoming convinced he was infected. https://www.telegraph.co.uk/global-health/science-and-disease/coronavirus-indian-man-died-suicide-becoming-convinced-infected/, (Accessed on 12/25/2020)
3. Covid19 tweets — kaggle. https://www.kaggle.com/gpreda/covid19-tweets, (Accessed on 12/25/2020)
4. deepset/covid_bert_base · hugging face. https://huggingface.co/deepset/covid_bert_base, (Accessed on 12/25/2020)
5. Fact check tools. https://toolbox.google.com/factcheck/explorer, (Accessed on 12/25/2020)
6. Fake news in india - wikipedia. https://en.wikipedia.org/wiki/Fake_news_in_India, (Accessed on 12/25/2020)
7. Home - newschecker. https://newschecker.in/, (Accessed on 12/25/2020)
8. Politifact. https://www.politifact.com/, (Accessed on 12/25/2020)
9. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
10. Castillo, C., Mendoza, M., Poblete, B.: Information credibility on twitter. In: Proceedings of the 20th international conference on World wide web. pp. 675–684 (2011)
11. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
12. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation 9(8), 1735–1780 (1997)
13. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759 (2016)
14. Kaliyar, R.K., Goswami, A., Narang, P., Sinha, S.: Fndnet–a deep convolutional neural network for fake news detection. Cognitive Systems Research 61, 32–44 (2020)
15. Kim, Y.: Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882 (2014)
16. Kwon, S., Cha, M., Jung, K., Chen, W., Wang, Y.: Prominent features of rumor propagation in online social media. In: 2013 IEEE 13th International Conference on Data Mining. pp. 1103–1108. IEEE (2013)
17. Liu, Y., Wu, Y.F.B.: Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In: Thirty-second AAAI conference on artificial intelligence (2018)

18. Müller, M., Salathé, M., Kummervold, P.E.: Covid-twitter-bert: A natural language processing model to analyse covid-19 content on twitter. arXiv preprint arXiv:2005.07503 (2020)
19. Patwa, P., Sharma, S., PYKL, S., Guptha, V., Kumari, G., Akhtar, M.S., Ekbal, A., Das, A., Chakraborty, T.: Fighting an infodemic: Covid-19 fake news dataset. arXiv preprint arXiv:2011.03327 (2020)
20. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
21. Sanh, V., Debut, L., Chaumond, J., Wolf, T.: Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108 (2019)
22. Shu, K., Sliva, A., Wang, S., Tang, J., Liu, H.: Fake news detection on social media: A data mining perspective. ACM SIGKDD explorations newsletter 19(1), 22–36 (2017)
23. Shu, K., Wang, S., Liu, H.: Beyond news contents: The role of social context for fake news detection. In: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining. pp. 312–320 (2019)
24. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. Advances in neural information processing systems 27, 3104–3112 (2014)
25. Tasnim, S., Hossain, M.M., Mazumder, H.: Impact of rumors and misinformation on covid-19 in social media. Journal of preventive medicine and public health 53(3), 171–174 (2020)
26. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
27. Wang, W.Y.: " liar, liar pants on fire": A new benchmark dataset for fake news detection. arXiv preprint arXiv:1705.00648 (2017)
28. Wolf, T., Chaumond, J., Debut, L., Sanh, V., Delangue, C., Moi, A., Cistac, P., Funtowicz, M., Davison, J., Shleifer, S., et al.: Transformers: State-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. pp. 38–45 (2020)
29. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E.: Hierarchical attention networks for document classification. In: Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies. pp. 1480–1489 (2016)
30. Zhang, J., Dong, B., Philip, S.Y.: Fakedetector: Effective fake news detection with deep diffusive neural network. In: 2020 IEEE 36th International Conference on Data Engineering (ICDE). pp. 1826–1829. IEEE (2020)
31. Zhang, X., Ghorbani, A.A.: An overview of online fake news: Characterization, detection, and discussion. Information Processing & Management 57(2), 102025 (2020)
32. Zhao, Z., Resnick, P., Mei, Q.: Enquiring minds: Early detection of rumors in social media from enquiry posts. In: Proceedings of the 24th international conference on world wide web. pp. 1395–1405 (2015)
33. Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., Xu, B.: Attention-based bidirectional long short-term memory networks for relation classification. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). pp. 207–212 (2016)