Fighting Fake News Using Deep Learning

Pre-trained Word Embeddings and the Embedding Layer Investigated

Fantahun Bogale Gereme
Institute of Fundamental and Frontier Sciences, University of Electronic Science and Technology of China, Chengdu, China
fantishb@gmail.com

William Zhu
Institute of Fundamental and Frontier Sciences, University of Electronic Science and Technology of China, Chengdu, China

ABSTRACT

Fake news is progressively becoming a threat to individuals, society, news systems, governments and democracy. The need to fight it is rising accompanied by various researches that showed promising results. Deep learning methods and word embeddings contributed a lot in devising detection mechanisms. However, lack of sufficient datasets and the question "which word embedding best captures content features" have posed challenges to make detection methods adequately accurate. In this work, we prepared a dataset from a scrape of 13 years of continuous data that we believe will narrow the gap. We also proposed a deep learning model for early detection of fake news using convolutional neural networks and long short-term memory networks. We evaluated three pre-trained word embeddings in the context of the fake news problem using different measures. Series of experiments were made on three real world datasets, including ours, using the proposed model. Results showed that the choice of pre-trained embeddings can be arbitrary. However, embeddings purely trained from the fake news dataset and pre-trained embeddings allowed to update showed relatively better performance over static embeddings. High dimensional embeddings showed better results than low dimensional embeddings and this persisted for all the datasets used.

CCS CONCEPTS

• Security and Privacy; **• Human and societal aspects of security and privacy**; **• Social aspects of security and privacy**;

KEYWORDS

fake news, word embedding, social media, deep learning, embedding layer

ACM Reference Format:

Fantahun Bogale Gereme and William Zhu. 2020. Fighting Fake News Using Deep Learning: Pre-trained Word Embeddings and the Embedding Layer Investigated. In *2020 The 3rd International Conference on Computational Intelligence and Intelligent Systems (CIIS 2020), November 13–15, 2020, Tokyo, Japan.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3440840.3440847

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CIIS 2020, November 13–15, 2020, Tokyo, Japan

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-8808-5/20/11…$15.00
https://doi.org/10.1145/3440840.3440847

1 CONTEXT

The advantages offered by online media, specifically social media, such as accessibility, low cost, suitable way of commenting & sharing, more timely nature, and prompt dissemination of information [2], [6], [12] have made it to be a preferred way of communication. However, it also plays a negative role by propagating inauthentic information such as fake news. The extensive spread of fake news is progressively becoming a threat to individuals, governments, freedom of speech, news systems and society as a whole, [8], [12], [15]. It disrupts the authenticity balance of the news system, creates real-life fears in the society, and threatens freedom of speech/press and democracy. To reduce the adverse effects of fake news, combatting fake news has become critical. To this response, some researches have been made using a variety of methods. Deep learning methods using news content and word embeddings have shown encouraging results [7].

For early detection of fake news, representative datasets and content representation techniques, to capture the news features that are important to distinguish fake news, are vital. For this purpose, word embeddings have been used to represent the features for traditional classifiers, or as initializations in deep neural networks. Word embeddings are real valued representations for text by embedding both semantic and syntactic meanings obtained from unlabeled large corpus and are perhaps one of the key advances for the remarkable performance of deep learning methods on challenging NLP (natural language processing) problems such as content based fake news detection. They are widely used in NLP tasks, such as sentiment analysis [21]; dependency parsing [16]; machine translation [5]; and fake news detection [2], [9], [10], [19], [20], [22], [23], [25].

There are several approaches of learning and utilizing word embeddings. (1) Learning a word embedding for a particular problem like fake news detection as used in [7]. Training the embedding can be done either in a standalone fashion, where a model is trained to learn the embedding, which is then saved and used as a part of another model for later task, or learning jointly [7], where the embedding is learned as part of a large task-specific model. (2) Reusing off-the-shelf word embeddings available often under a permissive license that can be used in two ways. Static, where the embedding is kept static and is used as off-the-shelf-component of a model and updated, where the embedding is used to seed the model being updated jointly during the training of the model.

In spite of the above mentioned options to obtain and use word embeddings, the question "what is a good word embedding model" remains an open problem [24]. Specific to the fake news detection problem, to the best of our knowledge, there was no relevant evaluation made to see which pre-trained word embedding is best for fake news detection problem. There was no relevant work made to evaluate whether training a word embedding on the available fake news datasets or using pre-trained embeddings is better for the same. There is no good clue to choose static use of off-the-shelf word embeddings or updating them to get better performance of a model. Moreover, the absence of representative fake news dataset makes content based fake news detection challenging. Our contributions on this work include the following.

- We prepared a multi-domain dataset for the fake news detection problem to fill the gap on the area.
- We introduced a deep learning based model in which we analyzed the performance of the various pretrained embeddings and the embedding layer for the fake news detection problem using news content.
- We further investigated the results of the performance of our model using pre-trained embedings as static and updated, and also evaluated the effect of using different embedding dimensions on model performance.

The rest of this document is organized as follows. In section 2, we present related work. Our research methodology is explained in section 3. Section 4 is dedicated for experiments and evaluation. Results and discussion are presented under section 5, while section 6 concludes our work.

2 RELATED WORK

Word embeddings were used in a multiplicity of NLP applications, including fake news detection [2], [9], [10], [19], [20], [22], [23], [25] either to represent the features for traditional classifiers, or as initializations in deep neural networks. Nevertheless, their use seems random. For example, word2vec embedding was used in [9], [20], [22], [23], [25] while GloVe was used in [2], [10], [19] with different dimensions in static or updated ways. On the other hand, [7] trained the keras embedding layer with publicly available task specific fake news detection datasets. In spite of their seemingly random utilization, word embeddings were not evaluated and used accordingly, in the fake news problem domain, though such works exist in other domains.

Current evaluation approaches can be grouped into two categories: intrinsic evaluation and extrinsic evaluation. Intrinsic evaluations measure the quality of word embeddings by directly computing the correlation between semantically and geometrically related words, typically through query words [18]. These kinds of evaluators test the quality of a representation independent of specific NLP task. They measure syntactic or semantic relationships between words directly [24]. For example, Word2Vec was evaluated on a word similarity task [13]. In extrinsic evaluations, word embeddings are used as input features to a particular task, and the evaluation of the word embeddings is done according to the performance metrics specific to that task [18]. For example, GloVe embeddings were evaluated on parts-of-speech tagging and named-entity recognition tasks [17].

A feasibility study of building “generalized” crisis classifiers through means of word embeddings and sentence encodings by [11] compared three types of word embeddings (Word2Vec, GloVe and FastText). They performed an extrinsic evaluation, where the embeddings were used with four traditional machine learning algorithms. The objective was to learn generalized classifiers for crisis tweet classification tasks. They used both pre-trained and crisis-specific word embeddings. Their results showed that the crisis-specific embeddings are more appropriate for specific crisis-related tasks, while the pre-trained embeddings are more suitable for general classification tasks.

An empirical evaluation of word embeddings trained from four different corpora by [24] showed that applying word embeddings trained from corpora in a general domain, such as Wikipedia and news, is not significantly inferior to applying those obtained from biomedical or clinical domain.

Another study made by [3] compared word2vec, fastText, and GloVe word embeddings to initialize a word-embedding layer of the seq2seq model for the conversation-modeling track of DSTC6. The expectation was that it helps the model to learn a better word representation for generating response utterances. Best result was obtained when FastText was used with the embedding size of 200.

Considering the difficulty of the fake news detection problem, content based methods using deep learning can benefit from suitable word embeddings. Extrinsic evaluation made on word embeddings in the specific task of fake news detection can contribute its share in combating fake news by allowing appropriate utilization of word embeddings.

3 METHODOLOGY

3.1 The Fake News Problem

The phrase “fake news” is used in different perspectives in various researches. Concepts related to fake news are summarized in Table 1 adopted from [26] based on their authenticity, intention and whether they are news or not. In its narrow sense, fake news can be defined, in line with [1], [12], [26], as news articles that are intentionally and verifiably false. This definition carries more weight from the perspective of fighting the bad effects of fake news, because, in spite of its untrue authenticity, the intention element existing to create inauthentic news adds more danger. It is this danger, in mind, which demands us to detect and take measure on fake news before it spreads and infects people and the news ecosystem.

3.2 Pre-trained Word Embeddings and the Embedding Layer in the Fake News Detection Problem

In this work, the Keras embedding layer and three famous pre-trained word embeddings (word2vec, GloVe and FastText) are to be investigated in the context of the fake news detection problem. Table 2 summarizes the word embedding architecture we used in this work.

3.2.1 Embedding Layer. An embedding layer is a word embedding that is learned jointly with a neural network model on a specific NLP task. It requires text to be preprocessed such that each word is one-hot encoded. The size of the vector space is specified as part of the model usually 50, 100, 200 or 300 dimensions, initialized with small random numbers. The embedding layer is used on the

Table 1: Concepts Correlated with Fake News

Concepts	Authenticity	Intention	News?
Malicious false news	False	Bad	Yes
False news	False	Unknown	Yes
Satire news	Unknown	Not bad	Yes
Disinformation	False	Bad	Unknown
Misinformation	False	Unknown	Unknown
Rumor	Unknown	Unknown	Unknown

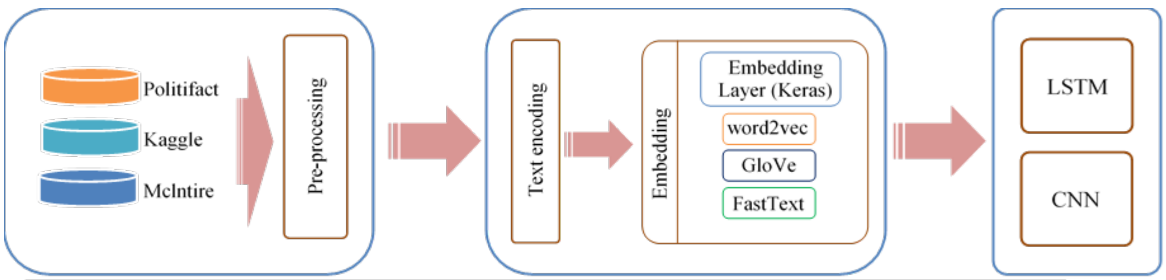

Figure 1: A high-level fake news detection model using deep learning and word-embeddings

Table 2: Word Embedding Architecture Used

Embedding Model	Dim	Vocabulary	Corpus
Embedding Layer (Keras)	300	–	–
GloVe	300	400K	Wikipedia 2014 and Gigaword 5
Word2vec	300	1M	Google News
FastText (with sub-word)	300	1M	wiki-news- Wikipedia 2017, UMBC webbase corpus and statmt.org news dataset

front end of a neural network and is fit in a supervised way using the Backpropagation algorithm. The one-hot encoded words are mapped to the word vectors. This type of embedding is used for fake news detection in [7].

3.2.2 Word2Vec. Introduced by [13], [14] at Google, as a response to make the neural-network-based training of embedding more efficient, word2vec word embedding is based on skip-gram and continuous bag-of-words (CBOW) models. Word2vev computes continuous vector representations of words based on a local window

3.2.3 GloVe. GloVe for Global Vectors, is a global log-bilinear regression model for word representations introduced by [17] at Stanford. It was meant to get the benefit of global matrix factorization and local context window methods together.

3.2.4 FastText. The FastText word embedding, contributed by [4], extends word2vec by introducing subword modeling and is based on the skip-gram model. FastText builds on Word2Vec by learning vector representations for each word and the n-grams found within each word. The values of the representations are then averaged into one vector at each training step.

3.3 Model Building Methods

We used Convolutional Neural Networks (CNN) and variants of Recurrent Neural Networks, Long Short-Term Memory (LSTM), as our model building blocks. Figure 1 depicts the high-level fake news detection model using deep learning and word embeddings.

3.4 Datasets

Early detection of fake news requires the use of the news content without waiting for its proliferation on social media. Detection methods using user engagements including likes or dislikes, shares, comments, etc. of users, are other important options. However, they need to wait until the news spreads across media, which allows unintended consumption of fake news by users. In this work, we use news content and to obtain a better representative dataset, we make use of three datasets, two publicly available datasets and one prepared by us as part of this work. Table 3 summarizes the dataset composition used in this work.

3.4.1 PolitiFact (2007-2019) Dataset. Politifact.com is a nonpartisan fact-checking website to sort out the truth in American politics. It covers a wide range of political topics, and they provide detailed judgments with fine-grained labels. The labels given to statements

Table 3: Composition of the Dataset Used in this Work

Dataset Name	Source	#Fake News	#Real News	Total
PolitiFact (2007-2019)	politifact.com (transcripts, speeches, news, press releases, campaign brochures, TV, social media)	5,241	5,241	10,482
Kaggle	Scraped from 244 websites by BS-Detecter	10,349	10,369	20,718
George Mclntire	The NY Times, WSJ, Bloomberg, NPR, the Guardian and Kaggle	3,151	3,155	6,306

Table 4: Evaluation Result of the Fake News Detection Model on the Three Datasets Using CNN and LSTM. Pre-Trained Word-Embeddings are Used in Two-Ways, As-Is(Static) and Updated.

Dataset	Word Embed-dings	CNN								LSTM							
		Static				Updated				Static				Updated			
		Acc	*Prec*	*Rec*	*F1*	*Acc*	*Rec*	*Prec*	*F1*	*Acc*	*Prec*	*Rec*	*F1*	*Acc*	*Prec*	*Rec*	*F1*
Kaggle	KEL	-	-	-	-	**.991**	**.989**	**.993**	**.991**	-	-	-	-	**.968**	**.973**	.963	**.968**
	W2V	.971	.974	.968	.971	.983	.977	.989	.983	.964	**969**	.960	.964	.963	.969	.957	.963
	GloVe	**.979**	**.9844**	.972	**.978**	.983	.975	.991	.983	**.966**	.966	**.966**	**.966**	.967	.962	**.971**	.967
	FastText	.971	.966	**.976**	.971	.985	.985	.985	.985	.964	.966	.961	.964	.967	.968	.966	.967
George Mclntire	KEL	-	-	-	-	**.973**	**.972**	**.969**	**.971**	-	-	-	-	**.923**	.928	.915	**.922**
	W2V	**.943**	**.943**	**.943**	.938	.947	.946	.949	.947	.915	.928	.899	.913	.918	**.930**	.901	.915
	GloVe	.938	.935	.942	**.943**	.946	.965	.916	.940	.918	.912	**.928**	**.920**	.912	.902	.926	.914
	FastText	.908	.886	.939	.912	.951	.948	.967	.957	**.920**	.928	.908	.918	.921	.928	.907	.918
Politifact (2007-2019)	KEL	-	-	-	-	.664	.668	.689	.678	-	-	-	-	.625	.636	.614	.625
	W2V	.670	**.703**	.655	.678	**.680**	**.697**	.665	.681	.657	.653	.694	.673	**.637**	.635	.668	.651
	GloVe	**.671**	.690	.688	.689	**.680**	.677	**.721**	**.698**	.656	**.656**	.676	.666	.618	**.643**	.558	.598
	FastText	.650	.636	**.797**	**.707**	.659	.658	.698	.678	**.678**	.654	**.778**	**.711**	.635	.631	**.678**	**.654**

Acc=Aaccuracy, Pre=Precision, Rec=Recall

by politifact.com include the following. TRUE: The statement is accurate and there is nothing significant missing. MOSTLY TRUE: The statement is accurate but needs clarification or additional information. HALF TRUE: The statement is partially accurate but leaves out important details or takes things out of context. MOSTLY FALSE: The statement contains an element of truth but ignores critical facts that would give a different impression. FALSE: The statement is not accurate. PANTS ON FIRE: the statement is not accurate and makes a ridiculous claim.

The need of datasets for the fake news detection problem is not yet satisfied. Liar was a dataset created in 2016 by [23] from politifact.com to fill this gap. However, it contains fewer statements as they used data before 2016 and was prepared for five-class classification. Fake news itself has become a big issue after the 2016 US election [1] and we believe data generated after this year has to be included in fake news detection datasets. Moreover, deep learning methods are data hungry and it is known that inclusion of as much data as possible could improve the accuracy of such methods. Motivated by these problems, we have scrapped 13 years of continuous text data from May-2007 to October-2019 from politifact.com, to prepare a dataset called PolitiFact(2007-2019)[1] . To do this, we created a python API that can crawl data from politifact.com. We used only the news content, which is text data. We merged TRUE and MOSTLY TRUE labeled articles together as both contain accurate statements, according to politifact.com to make the real-news-group. On the other hand, FALSE and PANTS ON FIRE are merged together to get the fake-news-group. Finally, we tried to balance the fake and real news groups by dropping extra articles from one group and end up with 5241 fake news statements and 5241 real news statements. The Politifact(2007-2019) data set contains a total of 10,482 articles.

3.4.2 Kaggle Dataset. The Kaggle[2] dataset was released by kaggle.com and is one of the public datasets considered for fake news detection problem. It is composed of more than 20700 articles well divided between fake and real news.

3.4.3 GeorgeMclntire Dataset. Released by George Mclntire[3] , this dataset has been used by many researches made on fake news detection. It contains more than 6300 articles roughly half fake and half real, with good balance between fake and real news.

4 EXPERIMENTS AND EVALUATION

4.1 Evaluation Metrics

We used accuracy, precision, recall and F1 score as evaluation matrices. We made several repeated experiments for each dataset and embeddings. Final results are obtained by averaging few top records.

[1] https://github.com/fanpoliti/politifact-2007-2019-dataset
[2] https://www.kaggle.com/mrisdal/fake-news
[3] https://www.kdnuggets.com/2017/04/machine-learning-fake-news-accuracy.html

Table 5: Evaluation Result Using the Glove Embedding Model and the Various Dimensions

Dataset	Dim	CNN				LSTM			
		Acc	*Pre*	*Rec*	*F1*	*Acc*	*Pre*	*Rec*	*F1*
Kaggle	50	.966	.956	.977	.966	.947	.941	.953	.947
	100	.976	**.984**	.967	.975	.960	.941	**.982**	.961
	200	.980	.978	.983	.980	**.973**	**.978**	.968	**.973**
	300	**.983**	.977	**.989**	**.983**	.967	.962	.973	.967
George Mclntire	50	.887	.874	.890	.882	.806	.836	.764	.798
	100	.899	.857	.945	.899	.826	.871	.767	.816
	200	.914	.879	.950	.913	.865	.834	**.913**	.872
	300	**.921**	**.890**	**.952**	**.920**	**.902**	**.928**	.873	**.900**
Politifact	50	.656	.656	.683	.670	.661	.678	.628	.652
(2007-2019)	100	.676	.653	**.778**	.710	.668	**.682**	.643	.662
	200	.680	.674	.721	.697	.675	.674	.690	.682
	300	**.702**	**.696**	.738	**.716**	**.678**	.654	**.773**	**.708**

4.2 Experimental Setup

We wrote both the main project code and the data scraping script in Python3.7.0, using TensorFlow r1.10, NumPy and Keras library. The general model is shown in Figure 1. We present specific details as follows. After making all the preprocessing on the data, 80% of each dataset was used as training set and the remaining 20% was set aside as validation set. For convolutional networks, we used an embedding dimension of 300, with 10,000 unique words and 5000 sequence length post padded with zeros. The output of the embedding layer is fed into a dense network of 128 neurons with ReLU (Rectified Linear Unit) activation function whose output is then passed into a one dimensional GlobalMaxPooling layer. The output of this layer is again fed into a dense network of 128 neurons with ReLU activation function whose output is finally passed into a one dimensional dense network with sigmoid activation function. We used Rmsprop as optimization technique and binary-crossentropy as the loss function.

For the LSTM, we set the maximum input length to be 500 post padded with zeros. Each input sequence is embedded into 64-dimensional vectors while using the keras embedding layer. The embedding dimension varies as 50, 100, 200 and 300 when pre-trained word embeddings are used. The embedded inputs are then fed into an LSTM network having 100 neural units which in turn feed a one dimensional dense network with sigmoid activation function. Adam optimizer is used as optimization function while binary_crossentropy is used as a loss function.

The Keras embedding layer was set to learn word embeddings directly from the fake news dataset during the training process. On the other hand, for several objectives, the three pre-trained word embeddings, word2vec, GloVe and FastText were used in two modes, static(as-is) and updated (allowed to update during the training). Comparison of the word embeddings among themselves and the Keras embedding layer, analyzing results of static and updated versions of each were among the objectives. To analyze the effect of embedding dimension in the performance of a model in the fake news dataset, the GloVe embedding was used with dimensions 50, 100, 200 and 300. Finally, to get representative results, all sets of experiments were repeated for all the datasets separately and results were recorded. Few best records were averaged to get the final result.

4.3 Results and Discussion

In this section, we discuss the results of the various extensive sets of experiments in detail.

4.3.1 Using Static Vs Updated Embeddings. As detailed in Table 4, experimental results showed that the model has performed well when pre-trained word embeddings are allowed to update with the fake news task specific dataset than using them as static (trained with general corpus). This is consistent on all the three datasets for the CNN based model. However, for the LSTM based model, the difference between static and updated results seems insignificant.

4.3.2 Using Different Dimensions. We showed the experimental results on the three datasets using different dimensions of the Glove pre-trained word embedding in Table 5. Based on the results, we can generally say that models using higher dimension word embeddings performed better than lower dimensions. This is consistent on the three datasets and the difference seems significant. For example, for the George Mclntire dataset, the accuracy difference is up to 4% between each consecutive steps of dimensions and 11% between the highest and the lowest dimensions.

4.3.3 Keras Embedding Layer, GloVe, Word2vec, and FastText Compared. Table 4 shows accuracy, recall, precision and F1 scores for CNN and LSTM on all the datasets using Keras Embedding Layer, GloVe, word2vec and FastText word Embeddings. The Keras embedding layer was purely trained with the task specific corpora. On the other hand, word2vec, GloVe, and fastText were pre-trained on general corpora. However, they were used in two modes static and updated. According to the results, we can say that the performance difference among

pretrained word embeddings does not seem significant and regular. The Keras embedding layer trained directly from the task specific corpus showed better performance in the Kagle and George Mclntire datasets.

5 CONCLUSION

In this paper, we have studied early detection of fake news using deep learning and news content under evaluation of pre trained word embeddings in the context of the fake news detection problem. Series of experiments showed that updating pre-trained embeddings with the task data brings improved results than using them statically (as they are) and the Keras embedding layer which is purely trained by the task specific corpus has given the model superior performance in the Kaggle and George Mclntire datasets. On the other hand, the choice among the various pre-trained embeddings may be taken arbitrarily as their performance difference is not significant and regular. Using higher dimensions of the embeddings is better than lower dimensions. Generally, in the presence of sufficient dataset, learning embeddings directly from the task specific dataset may be preferred over using off-the-shelf embeddings.

REFERENCES

[1] Hunt Allcott and Matthew Gentzkow. 2017. Social media and fake news in the 2016 election. Journal of Economic Perspectives 31 (2), pp.211-36, 2017, DOI: 10.1257/jep.31.2.211.
[2] Samir Bajaj. 2017. The pope has a new baby! fake news detection using deep learning. Stanford University, 2017.
[3] Zhuang Bairong, Wang Wenbo, Li Zhiyu, Zheng Chonghui, and Takahiro Shinozaki, Comparative analysis of word embedding methods for DSTC6 end-to-end Conversation Modeling Track, In Proceedings of the 6th Dialog System Technology Challenges (DSTC6) Workshop, Long Beach, USA, 2017.
[4] Piotr Bojanowski, Edouard Grave, Armand Joulin and Tomas Mikolov. 2017. Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics, Volume 5, pp.135-146, , 2017, DOI: 10.1162/tacl_a_00051.
[5] Kehai Chen, Tiejun Zhao, Muyun Yang, Lemao Liu, Akihiro Tamura, Rui Wang, Masao Utiyama and Eiichiro Sumita .2018. A neural approach to source dependence based context model for statistical machine translation. IEEE/ACM Transactions on Audio, Speech and Language Processing 26(2), pp.266–280, 2018, doi: 10.1109/TASLP.2017.2772846.
[6] William Ferreira and Andreas Vlachos. 2016. Emergent: a novel data-set for stance classification. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, ACL, 2016.
[7] Fantahun Bogale Gereme and William Zhu. 2019. Early detection of fake news, "before it flies high". In Proceedings of the 2nd International Conference on Big Data Technologies (ICBDT2019), pp.142-148, 2019, New York: Association for Computing Machinery, DOI: https://doi.org/10.1145/3358528.3358567.
[8] Minyoung Huh, Andrew Liu, Andrew Owens, and Alexei A. Efros, "Fighting fake news: image splice detection via learned self-consistency," The European Conference on Computer Vision (ECCV), Springer, pp.101-117, 2018.
[9] Hamid Karimi and Jiliang Tanh .2019. Learning hierarchical discourse-level structure for fake news detection. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.1, pp.3432–3442, Minneapolis: Association for Computational Linguistics, 2019, 10.18653/v1/N19-1347.
[10] Junaed Younus Khan, Md. Tawkat Islam Khondaker, Anindya Iqbal and Sadia Afroz. 2019. A benchmark study on machine learning methods for fake news detection. arXiv:1905.04749, 2019.
[11] Hongmin Li, Xukun Li, Doina Caragea and Cornelia Caragea. 2018. Comparison of word embeddings and sentence encodings as generalized representations for crisis tweet classification tasks. Proceedings of the ISCRAM Asian Pacific Conference, Wellington, New Zealand, 2018.
[12] Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J. Jansen, Kam-FaiWong, and Meeyoung Cha .2016. Detecting rumors from microblogs with recurrent neural networks. Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16), pp.3818-3824. New York, Association for the Advancement of Artificial Intelligence, 2016.
[13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean .2013a. Efficient estimation of word representations in vector space. International Conference on Learning Representations, 2013a.
[14] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado and Jeffrey Dean .2013b. Distributed representations of words and phrases and their compositionality. Advances in Neural Information Processing Systems 26, pp.3111–3119, 2013b.
[15] Saif M. Mohammad, Parinaz Sobhani and Svetlana Kiritchenko .2017. Stance and sentiment in tweets. ACM Transactions on Internet Technology 17(3), New York: Association for Computing Machinery, 2017.
[16] Hiroki Ouchi, Kevin Duh, Hiroyuki Shindo and Yuji Matsumoto. 2016. Transition-based dependency parsing exploiting supertags, IEEE/ACM Transactions on Audio, Speech, and Language Processing 24(11), pp.2059–2068, Piscataway, IEEE Press, 2016.
[17] 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), Stroudsburg, Association for Computational Linguistics, 2014, DOI: 10.3115/v1/D14-1162.
[18] Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims .2015. Evaluation methods for unsupervised word embeddings," In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp.298–307, Association for Computational Linguistic, 2015, DOI: 10.18653/v1/D15-1036.
[19] Sneha Singhania, Nigel Fernandez, and Shrisha Rao .2017. 3HAN: a deep neural network for fake news detection, International Conference on Neural Information Processing. Springer, Cham, 2017, DOI: https://doi.org/10.1007/978-3-319-70096-0_59.
[20] Tsubasa Tagami, Hiroki Ouchi, Hiroki Asano, Kazuaki Hanawa, Kaori Uchiyama, Kaito Suzuki, Kentaro Inui, Atsushi Komiya, Atsuo Fujimura Hitofumi Yanai, Ryo Yamashita and Akinori Machino. 2018. Suspicious news detection using micro blog text. arXiv:1810.11663, 2018.
[21] Tang D., Wei F., Yang N., Zhou M., Liu T., and Qin B. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In Proceedings of the 52 nd Annual Meeting of the Association for Computational Linguistics 1, pp.1555-1565, Baltimore, Maryland: Association for Computational Linguistics, 2014.
[22] Aswini Thota, Priyanka Tilak, Simrat Ahluwalia and Nibrat Lohia .2018. Fake news detection: a deep learning approach. SMU Data Science Review 1(3), Article 10, 2018.
[23] William Yang Wang .2017. Liar-Liar pants on fire: a new benchmark dataset for fake news detection. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 422–426, Vancouver, Canada, 2017, DOI: 10.18653/v1/P17-2067.
[24] Bin Wang, Angela Wang, Fenxiao Chen, Yuncheng Wang and C. Jay Kuo. 2019. Evaluating word embedding models: methods and experimental results. ArXiv, abs/1901.09785, 2019.
[25] Yang Y., Zheng L., Zhang J., Cui Q., Li Z., and Yu P. 2018. TI-CNN: convolutional neural networks for fake news detection. arXiv:1806.00749, 2018.
[26] Zhe Zhao, Paul Resnick, and Qiaozhu Mei, Enquir-ing minds: early detection of rumors in social media from enquiry posts. In Proceedings of the 24th International Conference on World Wide Web.1395–1405, New York, Association for Computing Machinery, 2015.