Chapter 37
A Hierarchical Learning Model for Claim Validation

Amar Debnath, Redoan Rahman, Md. Mofijul Islam and Md. Abdur Razzaque

1 Introduction

In recent years, the massive growth of different social media platforms allowed the rapid dissemination of information to the mass people. Moreover, almost all of the well-known news media utilized these social platforms to spread their contents. Hence, people increasingly depend on these social media platforms to seek out and consume news. For example, in 2016, about 62% of U.S. adults depend on social media to get news content.[1] On the other hand, propagandists exploit these platforms which lead to the rapid dissemination of the fake or distorted news. For instance, during United States Election of 2016, there were 156 fake news that was identified later, among which 115 were pro-Trump and 41 of them were pro-Clinton [1]. Thus, fake news intentionally persuades consumers to accept biased or false beliefs and subsequently, it is very much crucial to stop the spreading of distorted contents.

To mitigate the negative effects of fake news, it is critical to develop methods for automatically detecting fake news on social media platforms. However, it is a strenuous task to detect the distorted news, as the propagandists carefully craft the news to present it as validated news. Even one study reveals that human evaluators

[1] www.journalism.org/2016/05/26/news-use-across-social-media-platforms-2016/.

A. Debnath · R. Rahman · Md. Mofijul Islam (✉) · Md. Abdur Razzaque
Green Networking Research Group, Department of Computer Science and Engineering, University of Dhaka, Dhaka, Bangladesh
e-mail: akash.cse.du@gmail.com

A. Debnath
e-mail: amar.csedu@gmail.com

R. Rahman
e-mail: redoanrahman744@gmail.com

Md. Abdur Razzaque
e-mail: razzaque@du.ac.bd

M. S. Uddin and J. C. Bansal (eds.), *Proceedings of International Joint Conference on Computational Intelligence*, Algorithms for Intelligent Systems, https://doi.org/10.1007/978-981-13-7564-4_37

can identify the fake news with only 50–63% accuracy [14]. Moreover, brief contents in the social media platforms make this task more arduous. Therefore, designing an automated fake news detection approach on social media has recently become an emerging research area.

Recently, a couple of content verification systems have been introduced to address the fake news problem. For example, expert-based fact-checking platforms have been introduced (PolitiFact,[2] Snopes[3]). Furthermore, crowd-sourced expertise has been utilized to detect the check worthy statements, for instance, Fiskkit[4] allows the users to annotate news contents. Even though these approaches are able to verify the news articles, the main drawbacks of these systems are the scalability. Hence, in recent years, Natural Language Processing practitioners have been trying to solve the deception detection issue with the help of various statistical machine learning approaches [15]. Nevertheless, the main obstacle to develop and evaluate these learning approaches is the lack of comprehensive benchmark dataset. Additionally, fake news contain various linguistic cues, which enforce a diverse dataset to train and evaluate the state-of-the-art learning models. Recently, a couple of datasets have been introduced in the literature to aid the claim verification problem [3, 16, 17].

In this work, we proposed TLCV: Transfer Learning based Claim Validation model, where we incorporate composite Convolutional Neural Network with deep contextualized word vector representation model, ELMo [12]. Our proposed approach TLCV outperforms the state-of-the-art-works in identifying the check worthy statements. The major contributions of this work are summarized below:

- Design a transfer learning-based claim validation learning model.
- Develop a composite Convolutional Neural Network to achieve the consistent state-of-the-art performance.
- Employ the deep contextual word representation to capture diverse linguistic cues to improve the fake news detection approach.

The rest of the works are designed as follows: in Sect. 2, we have discussed the related works. Thereafter, the proposed claim verification model is presented in Sect. 3. Subsequently, in Sect. 4, the detailed experimental analysis and performance evaluation of the proposed models are discussed. Finally, Sect. 5 has concluded the work with future plans.

2 Related Works

In recent years, deception detection problem has been studied in various domains from different angles. In [8], computational linguistics-based deceptive opinion mining approaches have been proposed and a crowd-sourced dataset has been developed.

[2] https://www.politifact.com.

[3] https://www.snopes.com/.

[4] http://fiskkit.com/.

However, the detection of fake news is challenging, due to the diverse linguistic cues. A few works have been addressed for the validation of news contents, because of the lack of benchmark datasets to aid this fact-checking problem.

In [3], the author proposes a fact-checking dataset, nonetheless, it contains only 221 training examples and thus, this dataset is not enough to design a comprehensive claim validation learning model. In [17], the author established a novel dataset, named LIAR, which consists of 12.8K human-labeled short statements from Politifact platform and each of these statements is categorized into six different labels—*pants-fire, false, barely-true, half-true, mostly-true and true*. LIAR is one of the most comprehensive datasets, which exclusively addressed the claim validation problem. Moreover, LIAR contains textual statements along with the speakers' profile metadata—*party affiliations, state, job and credit history*. Recently, another dataset, FEVER: Fact Extraction and Verification [16], ushered a new way to confront the deception detection problem. The characteristic of FEVER dataset is that it includes the evidences of the claim statements, which are extracted from a knowledge base (Wikipedia).

A number of computational approaches have been proposed to validate the news contents by utilizing the publicly available datasets. In [4], the author analyzed user reviews from Amazon and concluded a few assumptions that fraud reviewers tend to stay with their review style and make a lot of similar comments with various word combinations. Linguistic-based approach has been proposed to detect deceptive text on a crowd-sourced dataset in [11]. Nevertheless, the lexical features are not enough for designing the fraud detection learning model. In [6], the authors showed that speaker profiles are also useful for classifying fake news. Additionally, in the LIAR [17] dataset, the author utilized a hybrid Convolutional Neural Network to develop a computational claim verification learning model. Moreover, in [7], the author proposed a hybrid model with attention mechanism, which utilized Long Short-term Memory learning model as well as attention blocks on speaker and topic attributes. This approach outperforms the baseline approach of LIAR dataset [17].

Unlike the state-of-the-art-approach, in this work, we propose a computational approach, TLCV, to validate a claim statement, where we not only just utilize the textual features and author profile but also linguistic cues retrieved using the composite CNN model. Moreover, we also utilize the transfer learning approach to extract the deep contextual information from the claim statements.

3 Proposed Computational Claim Validation Approaches

In this work, we have designed a computational approach, TLCV, to validate the check worthy statement by leveraging the deep contextual word representation and the ruling comments extracted from the Politifact platform. In addition, we also incorporated the speaker information from LIAR dataset to train our claim validation learning model.

3.1 TLCV Architecture

In TLCV, we employed a composite Convolutional Neural Network (CNN) model to train the claim validation model. The proposed TLCV model can be divided into three modules:

– Retrieve Evidence.
– Vectorized Representation of Composite Features.
– Composite CNN Training Model.

Retrieve Evidence The textual information of the claim statement is not sufficient to check the validity. For this reason, this module is designed to retrieve evidence and augment this information in training the fact-checking model. A couple of state-of-the-art approaches have been proposed to retrieve related information of a given statement. For instance, we can utilize the TF–IDF similarity [13] to extract related information from the large knowledge base. Furthermore, Sent2vec [9] with compositional n-gram features can be utilized to retrieve contextual information. TF–IDF [13] performs fairly well but it focuses on mainly syntactic features. On the other hand, Sent2vec [9] uses n-gram features to represent a sentence into a vector to capture the semantic information. This vectorized format of the sentence can be utilized to retrieve related information. In this work, later approach, Sent2vec, has been incorporated to extract sentence as the compositional n-gram features can help to capture the appropriate textual semantic cues.

The extracted sentence vector representation has been utilized to calculate the cosine similarity between news claim statements and the sentences of the ruling comments, which are extracted from the Politifact, for determining the effective evidence. In this work, we directed our attention to the evidence retrieved from Politifact.com.

Vectorized Representation of Composite Features This module of TLCV is responsible to represent the textual and other meta-features into a vectorized format. As the TLCV has been designed by incorporating the composite CNN model, hence the representation of the textual features into the vectorized format was needed to feed into the CNN model. In this work, the pretrained ELMo [12] model was utilized to represent the textual features into the vectorized format, as ELMo can capture deep contextual word representation. The ELMo model consists of 13 million parameters and it is trained on the 1 billion Word Benchmark [2], which can represent a sentence in 256- dimensional vectors. In the TLCV, the following three groups of features have been incorporated.

– The claim sentences in the LIAR dataset are padded to be 100 long word sequence. Then, the sentences were fed into the ELMo model and the words of the sentences are converted to 256-dimensional vectors making each sentence size to be (100, 256).
– Ruling comments are scrapped from Politifact and sent2vec [9] method has been employed to fetch the appropriate evidence. Similar to claim sentences represen-

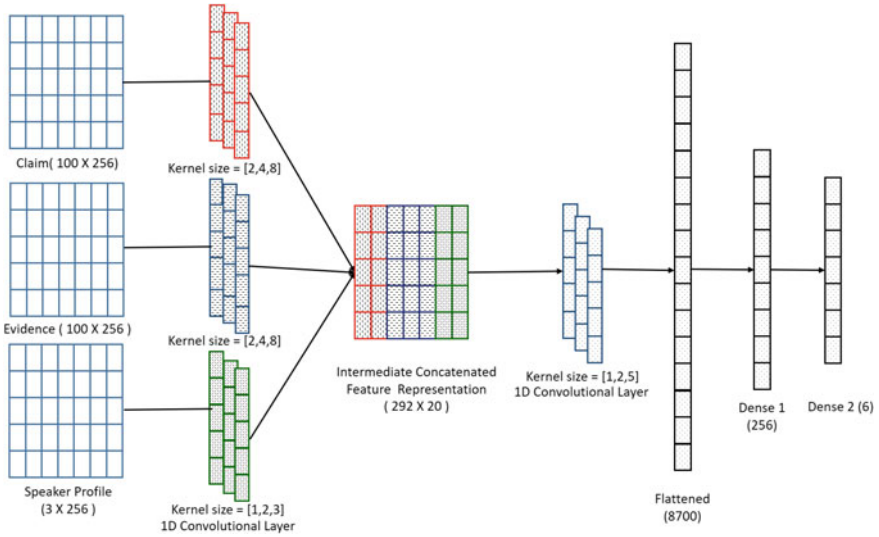

Fig. 1 Training model of TLCV

tation, the evidence sentences are processed and converted to (100, 256) sized matrix.

– In TLCV model, the speaker information (speaker's name, job and party affiliations) have also been incorporated. We concatenated these three pieces of information into one sentence. After that these sentences were converted to 256-sized vector representations using the ELMo model, similar to the previous approach. Hence, each speaker information is represented by (3, 256) sized matrix.

Composite CNN Training Model Convolutional Neural Networks have proven to be efficient when it comes to sentence classification task [5]. Hence, to design the underlying architecture of TLCV, the composite CNN model is utilized, where each of the feature groups is fed into a separate CNN filter set. The complete composite training model architecture is depicted in Fig. 1.

In the first layer of TLCV training model, three input matrices were constructed. Feature embeddings of claim statements, evidence sentences, and author profile information were turned into a vectorized format using ELMo model. Three sets of convolutional filters were initialized and applied into these features embedding. The three kernel sizes of the first two convolutional filter sets are {2, 4, 8} and the kernel sizes for the third convolutional filter set are {1, 2, 3}.

The resulting feature embeddings from the first layer are concatenated and a composite feature embedding is created. After that, another filter set of size {1, 2, 5}, has been applied to this composite feature embedding. The output of the filter is flattened to be passed to a dense neural network of hidden neuron size 256. Subsequently, the output of the first dense layer is again passed to another dense layer consisting of six neurons with softmax activation for producing the final output. Relu activation

and valid padding are used for all the convolutional filters. Standard dropout with discrete probability has been utilized to prevent over-fitting. This architecture gave a consistent result and proved efficient in extracting input features.

3.2 Extended Models of TLCV

The proposed model TLCV has been extended and two computational models are designed. However, the underlying training module architecture of these two extended models is similar to the previous TLCV model. The purpose of these two extended models is to serve as a baseline model to evaluate the performance of the proposed TLCV model.

CVMwoSF: Claim Validation Model without Supplementary Features In this extended learning model, GloVe [10] word vector representation model has been employed to create the feature embeddings. Moreover, in CVMwoSF model the supporting features such as retrieved evidence and speaker profile was not incorporated.

CVMwSF: Claim Validation Model with Supplementary Features In CVMwSF mode, similar to the TLCV, retrieved evidence and speaker profile was incorporated as supporting features. However, in this model, GloVe word vector representation is utilized for feature embeddings instead of ELMo.

4 Experimental Analysis

For evaluating the performance of our proposed claim verification model, TLCV, the LIAR dataset [17] was utilized, as it not only consists of the claim statement but also the speakers' profile information. Each of the claim statements of LIAR dataset is categorized into six labels—*pants-fire, false, barely-true, half-true, mostly-true, and true*. Besides the speaker profile information and claim statements, the LIAR dataset was augmented with ruling comments which serve as an evidence of the claim statements. A python scraper was written which helps to extract the ruling comments from the Politifact platform. The detailed information of LIAR dataset is presented in Table 1.

We conduct the performance evaluation of TLCV model in Google Colaboratory Platform.[5] This platform provides Tesla K80 GPU, having 2496 CUDA cores, 12 GB GDDR5 VRAM. The CPU has a single-core hyper-threaded Xeon Processors with 2.3 GHz clock speed. Additionally, it contains 13 GB RAM and 33 GB available disk space.

[5] https://colab.research.google.com.

Table 1 LIAR dataset

(a) Top 3 speakers

Top-3 Speaker	
Barack Obama	94
Donald Trump	54
Hillary Clinton	48
Top-3 Party Affiliations	
Democrats	4150
Republicans	5687
None (e.g., FB posts)	2185

(b) Count of each label

Label	Pants-true	False	Barely-true	Half-true	Mostly-true	True
Training Set	839	1995	1654	2114	1962	1676
Validation Set	116	263	237	248	251	169
Test Set	92	249	212	265	241	208

4.1 Performance Result

In this section, the detailed performance evaluation of our proposed claim validation model TLCV is presented. The performance of TLCV was compared with two baseline approaches, which are proposed in LIAR dataset [17]. The main difference between these approaches is that one approach utilizes only the claim statements and another model incorporates meta-features of fact, speaker profile, and history, in addition to the textual claims.

In addition to these two state-of-the-art baseline approaches, the performance of TLCV model was compared with the two extensions of TLCV: CVMwSF and CVMwoSF. The detailed performance evaluations of these claim verification models are presented in Table 2.

It can be easily identified that CVMwoSF claim verification model slightly outperforms the baseline approaches, which are proposed in LIAR work [17]. The only reason is that CVMwoSF utilizes the composite CNN training model, where each of the features groups is trained using a separate CNN model to get the feature embeddings. However, in the CVMwSF model, when the speaker profile was incorporated with CVMwoSF, the performance improves with accuracy 28.75%. Because the author of fake news has a distinctive profile. Nonetheless, both CVMwoSF and CVMwSF models incorporate GloVe model to get the vectorized representation of the textual features.

On the other hand, fake news has diverse linguistic features, hence the pretrained GloVe or word2vec model cannot fully capture the semantic textual features. For this reason, in TLCV model, we leverage the pretrained ELMo model to retrieve

Table 2 Performance comparison of different claim verification models

Model	Accuracy	Precision	Recall	F1-score	MCC
LIAR baseline [17]	27.0	–	–	–	–
LIAR baseline with speaker profile [17]	27.4	–	–	–	–
CVMwoSF	27.52	28	27	27	0.1238
CVMwSF	28.75	29	28	28	0.1294
TLCV	29.38	30	29	28	0.1327

the deep contextual word representation to extract the composite feature embeddings. As ELMO model is able to capture the deep contextual word representation, TLCV approach outperforms the state-of-the-art models with accuracy of 29.38%. This performance improvement indicates that the claim verification model can be improved by incorporating the language model and transfer learning approach.

Additionally, in another work [7], the author used a speaker profile and credit history along with the claim statement to detect authenticity. Although in [7], the author achieved a better result by incorporating the speaker credit history and bidirectional long short-term memory model (Bi-LSTM), however, in most general cases, speaker profile containing their truth history may not be available, which makes the pro-

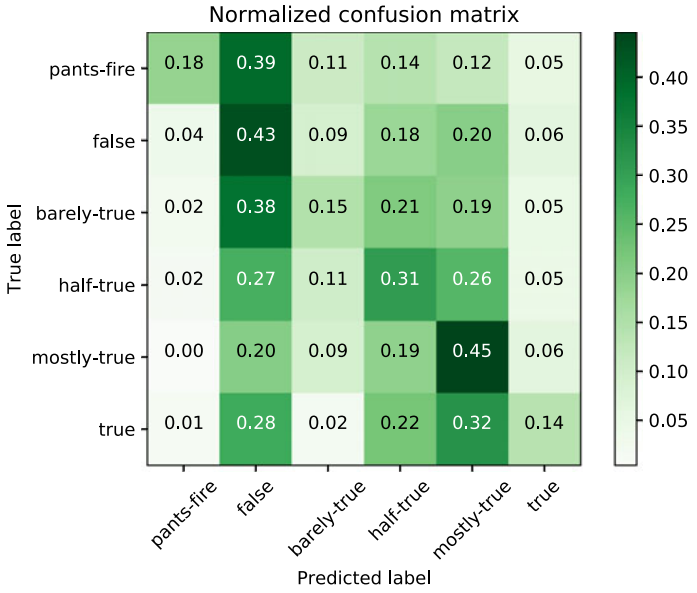

Fig. 2 Confusion matrix of TLCV approach on test data

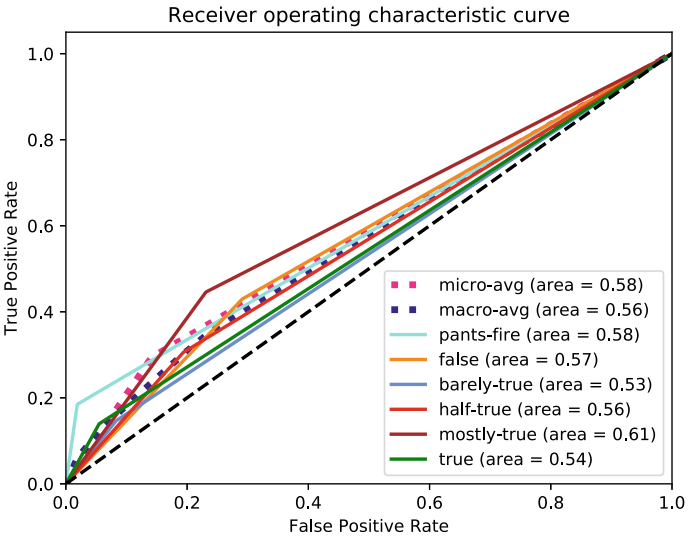

Fig. 3 Receiver operating characteristics (ROC) curve of TLCV

posed claim verification system impractical. For this reason, we did not consider this approach to evaluate the performance of TLCV approach.

The categorical performance of the TLCV approach is depicted in Fig. 2. It can be easily identified from the ROC graph of TLCV in Fig. 3 that the categorical performance of our proposed model is quite consistent. The reason behind this consistent performance is that TLCV model leverages the deep contextual vector representation of ELMo.

5 Conclusion

In this work, we proposed TLCV, a transfer learning-based claim verification approach, where we utilized the composite Convolutional Neural Network to detect the authenticity of political news. Furthermore, to improve our proposed claim verification model, we incorporated evidence retrieval module, which helps to extract evidence of the claim statement by utilizing the sent2vec approach. In the experimental analysis, our proposed model achieved 29.38% accuracy which outperforms the state-of-the-art baseline approach by 8.81%. One of the main reasons behind this performance improvement is that we leveraged ELMo model, to extract the deep contextual feature embedding of the textual information. Moreover, separate CNN models have been applied to develop composite feature group representation. The experimental evaluations of TLCV indicated that there is a lot of scope available to

improve the claim verification model by leveraging the transfer and deep learning approaches.

In future, we have a plan to extend the evidence retrieval model of TLCV to extract evidence from a large and diverse knowledge base. Additionally, transfer learning approach can be employed to fine tune the pretrained model in the target domain.

References

1. Allcott H, Gentzkow M (2017) Social media and fake news in the 2016 election. Working paper 23089, National Bureau of Economic Research. https://doi.org/10.3386/w23089
2. Chelba C, Mikolov T, Schuster M, Ge Q, Brants T, Koehn P, Robinson T (2014) One billion word benchmark for measuring progress in statistical language modeling. In: INTERSPEECH. ISCA, pp 2635–2639
3. Ferreira W, Vlachos A (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, pp 1163–1168
4. Kaghazgaran P, Caverlee J, Alfifi M (2017) Behavioral analysis of review fraud: linking malicious crowdsourcing to Amazon and beyond. In: ICWSM, pp 560–563
5. Kim Y (2014) Convolutional neural networks for sentence classification. In: EMNLP. ACL, pp 1746–1751
6. Long Y, Lu Q, Xiao Y, Li M, Huang CR (2016) Domain-specific user preference prediction based on multiple user activities. In: 2016 IEEE international conference on big data (Big Data). IEEE, pp 3913–3921
7. Long Y, Lu Q, Xiang R, Li M, Huang CR (2017) Fake news detection through multi-perspective speaker profiles. In: Proceedings of the eighth international joint conference on natural language processing (volume 2: short papers), vol 2, pp 252–256
8. Ott M, Choi Y, Cardie C, Hancock JT (2011) Finding deceptive opinion spam by any stretch of the imagination. In: Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies-volume 1. Association for Computational Linguistics, pp 309–319
9. Pagliardini M, Gupta P, Jaggi M (2018) Unsupervised learning of sentence embeddings using compositional n-gram features. In: Proceedings of the 2018 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long papers). Association for Computational Linguistics, pp 528–540. https://doi.org/10.18653/v1/N18-1049
10. Pennington J, Socher R, Manning C (2014) Glove: global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1532–1543
11. Pérez-Rosas V, Mihalcea R (2015) Experiments in open domain deception detection. In: Proceedings of the 2015 conference on empirical methods in natural language processing, pp 1120–1125
12. Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, Zettlemoyer L (2018) Deep contextualized word representations. In: Proceedings of NAACL
13. Ramos J et al (2003) Using TF-IDF to determine word relevance in document queries. In: Proceedings of the first instructional conference on machine learning, vol 242, pp 133–142
14. Rubin VL (2017) Deception detection and rumor debunking for social media. The SAGE handbook of social media research methods, pp 342–363
15. Shu K, Sliva A, Wang S, Tang J, Liu H (2017) Fake news detection on social media: a data mining perspective. ACM SIGKDD Explor Newsl 19(1):22–36

16. Thorne J, Vlachos A, Christodoulopoulos C, Mittal A (2018) FEVER: a large-scale dataset for fact extraction and verification. In: NAACL-HLT
17. Wang WY (2017) Liar, liar pants on fire: a new benchmark dataset for fake news detection. In: Proceedings of the 55th annual meeting of the association for computational linguistics (volume 2: short papers), vol 2, pp 422–426