# HACK: A Hierarchical Model for Fake News Detection

Yanqi Li[1,2], Ke Ji[1,2], Kun Ma[1,2], Zhenxiang Chen[1,2], Jun Wu[3], Yidong Li[3], and Guandong Xu[4]

[1] School of Information Science and Engineering, University of Jinan, Jinan, 250022, China
[2] Shandong Provincial Key Laboratory Of Network Based Intelligent Computing, University of Jinan, Jinan, 250022, China
[3] School of Computer and Information Technology, Beijing Jiaotong University, Beijing, 100044, China
[4] Data Science and Machine Intelligence Lab, Advanced Analytics Institute, University of Technology Sydney, Australia

**Abstract.** Online social media sites have become the most powerful platform to share news nowadays. However, all kinds of unauthenticated news that are released online without strict limits may lead to the spread of fake news, which has become a synonym for social and political threats. The existing solutions to the fake news issue are mostly trying to construct a social graph network by integrating the news content and social context of the news, which may be restricted when lacking social context information. In this paper, we propose a model for text only, regardless of contextual information, and named it HACK (**H**ier**A**rchical dete**C**tion for fa**K**e news), which can construct high-level combined features of spatial capsule vectors from low-level character features and phrase features by fusing a pre-trained language model and convolution network. The experimental results on real-life data show that the classification accuracy is significantly improved by our method comparing with the state-of-the-art methods.

**Keywords:** Fake news · Hierarchical framework · Feature extraction · Pre-trained LM · CapsNet.

## 1 Introduction

With the development of the Internet, people tend to deliver news on online social media sites, which allow anyone to express themselves and convey the news to others. However, most of the sites have no strict limits on user behavior, and so all kinds of unauthenticated news are heavily released, among them there is probably fake news as well.

As social media has widely permeated all aspects of our life, the spread of fake news may mislead popular belief and poses a serious threat to public security. Several approaches have been proposed for solving the problem. They can be broadly divided into three categories: content-based methods and social context-based methods. In the above research work, machine learning methods and deep learning methods are employed to improve classification performance. Though existing work has overcome this issue to a certain extent, there are some restrictions for practical use. For example, many times the publishers of fake news may just be newcomers, only have little user context information, even without social network behavior. In this case, it's impossible to detect fake news only relying on construct propagation network based on social context information.

Many content-based classification methods like [18] have been implemented to the fake news issue, particularly some deep learning frameworks via CNN [7] or RNN [2], which can encode the text to represent content. However, most of the frameworks extract features from an overall perspective, seriously losing structural information about spatial patterns in different positions of the text. In addition, not all content features are useful and isolated features may not do much for recognizing fake news. Clearly, the combination of local features in different positions is more beneficial to distinguish fake news.

## 2 Related Work

### 2.1 Fake News Detection

The existing methods can be divided into two categories, content-based and social context-based methods, and they are often used in combination in actual operation.

News content is the most intuitive discriminant basis. Because fake news often has exaggerated or radical emotional overtones [16], there has been some research work on conflict viewpoints [8] and stance factors [19]. Apart from the content itself, social contextual information related to the news and publishers [1] (e.g., number of posts, age of the account, number of friends/followers) can present a clear description of when the news appears, which is used to give a verified status for the credibility. Furthermore, one of the main hazards of fake news lies in its powerful dissemination ability in a social context. As in [8], the authors propose a credibility propagation network by taking advantage of the conflicting viewpoints of users. In [13], the authors propose a model that can capture changes in user characteristics along the propagation path based on recurrent and convolutional networks. Furthermore, the proposed model in [14] utilizes a four-layer Graph CNN to fuse content, user profile and activity, social graph, and news propagation, which has a better performance than using convolutional networks. However, there are some restrictions for context-based methods when social context information collected from the publishers is sparse or limited.

### 2.2 Text Classification

In essence, fake news detection can be abstracted to the text classification issue (more specifically, binary classification). Therefore, many NLP-based machine learning approaches can be applied to solve the fake news issue.

The core of text classification is how to get a text representation that contains more information. The traditional methods such as K-nearest Neighbor and Naive Bayes are influenced greatly by the sparsity of features. In recent years, the neural network is introduced into NLP, particularly the pre-trained language model, which can construct better-distributed feature representation with contextual information and model complex relationships within sparse data. Representative methods are Word2Vec and GloVe. However, both of them can only generate static word embeddings, which leads to the polysemy problem. To solve this problem, pre-trained language models such as ELMO [15], GPT [17], and BERT [3] are designed to dynamically adjust word embedding according to text context. In addition, due to the powerful effect of CNN and RNN in processing sequence data, many text classification methods [4, 5] are proposed. The existing classifiers with good performance are all based on CNN and RNN. We will use them as the baseline approaches and discuss them in Results Analysis.

## 3 Our Model

### 3.1 Character feature representation

The pre-trained language model can not only integrate contextual information into each character, the smallest Chinese language unit, through the self-attention mechanism but also bring prior knowledge from other corpora into the current task, overcoming the problem of insufficient data. We choose Bert [3] as the first level of our framework to get the embedding representation for the characters.

The input is the sum of $k$-dimensional embeddings: token $E_i^t$, segmentation $E_i^s$, and position $E_i^p$:

$$x_i = E_i^t \oplus E_i^s \oplus E_i^p \tag{1}$$

Given news $x_i$ containing $L$ characters, the pre-training Bert can output the embedding vectors $T = [\mathbf{t_1}, \mathbf{t_2}, \dots, \mathbf{t_L}]$ corresponding to the characters.

### 3.2 Phrase feature representation

Character is the basic language unit, several adjacent characters can compose a local feature of the text, particularly for Chinese text, only the word or phrase with multiple characters has expressive meaning. We set a sliding window across the vectors $\mathbf{t_1}, \mathbf{t_2}, \mathbf{t_3}, \dots$.

Assume window size is set to $K_1$ and $j$-th filter for the convolution operation is $W_j \in \mathbb{R}^{K_1 * k}$. For example, a new feature $c_{i,j}$ is generated from the adjacent characters in the window:

$$c_{i,j} = f(W_j \circ \mathbf{t_{i:i+K_1+1}} + b) \tag{2}$$

, where $\mathbf{t_{i:i+K_1-1}}$ means the characters from $i$ to $i + K_1 - 1$, $f$ is a nonlinear function, $\circ$ is element-wise multiplication and $b$ is a bias term. If we set $B$ filters, the similar operations in the same position can generate $B$ new features that are arranged as a vector:

$$\mathbf{c_i} = (c_{i,1}, c_{i,2}, c_{i,3}, \ldots, c_{i,B}) \tag{3}$$

The sliding window from front to back can produce $L - K_1 + 1$ vectors, which can be arranged as matrix $C \in \mathbb{R}^{(L-K_1+1)\times B}$.

$$C = [\mathbf{c_1}, \mathbf{c_2}, \mathbf{c_3}, \ldots, \mathbf{c_{L-K_1+1}}] \tag{4}$$

### 3.3 Sentence feature representation

Because the kernel size of the convolution filter is usually small and only focuses on the local N-gram, the pooling operation will lead to the loss of position information. The combination of some local features in different positions will be possible for finding significant semantic features or spatial patterns to distinguish them. Inspired by CapsNet [6], we construct high-level combined feature representations based on it. The architecture is shown in Fig. 1, including three layers:

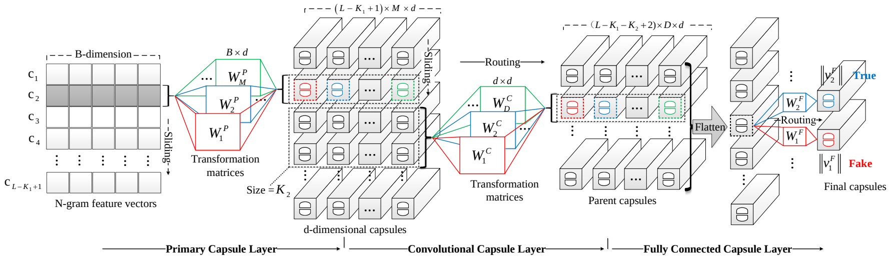

Fig. 1: The third level of our framework, which uses capsule network to construct high-level combined feature representation with capsule vectors.

***Primary Capsule Layer*** converts every vector to some $d$-dimensional capsules by $M$ transformation matrices $W_1^P, W_2^P, \cdots, W_M^P \in \mathbb{R}^{B\times d}$, $j$-th capsule $\mathbf{p_{i,j}}$ derived from vector $\mathbf{c_i}$ is computed as:

$$\mathbf{p_{i,j}} = g\left(W_j^P \mathbf{c_i} + \mathbf{b_l}\right) \tag{5}$$

where $g$ is the nonlinear squash function. The same operation on all vectors of $C$ will collect $(L - K_1 + 1) \times M$ capsules as tensor $P$.

***Convolutional Capsule Layer*** sets a $K_2 \times M$ window on $P$, and learns $D$ parent capsules based on $K_2 \times M$ child capsules in the window. Suppose shared transformation matrices $W_1^C, W_2^C, \cdots, W_D^C \in \mathbb{R}^{d\times d}$. Given a child capsule $\mathbf{u_i}$, the prediction vector $\mathbf{\hat{u}_{j|i}}$ for potential parent capsule $\mathbf{v_j}$ is computed as:

$$\mathbf{\hat{u}_{j|i}} = W_j^F \mathbf{u_i} + \mathbf{\hat{b}_{j|i}} \tag{6}$$

, where $\hat{\mathbf{b}}_{\mathbf{j|i}}$ is the capsule bias term. With the predictions, the dynamic routing algorithm is used to optimize iteratively how each child is sent to an appropriate parent. After the window slides to the end, we can have $(L - K_1 - K_2 + 2) \times D$ parent capsules. See the paper [6] for details of the dynamic routing algorithm.

***Fully Connected Capsule Layer*** flats the parent capsules below into a list $\mathbf{v_1}, \mathbf{v_2}, \cdots, \mathbf{v_{(L-K_1-K_2+2)\times D}}$, which are fed into $H$ final capsules $\mathbf{v_1^F}, \mathbf{v_2^F}, \cdots, \mathbf{v_H^F}$ by a fully connected layer. The process is similar to the steps above approach. Suppose $H$ shared transformation matrices $W_1^F$, $W_2^F$, $\cdots, W_H^F \in \mathbb{R}^{d\times d}$. The dynamic routing algorithm is used to optimize iteratively how each capsule $\mathbf{v_i}$ is sent to the final capsule $\mathbf{v_j^F}$. Since recognizing fake news is binary classification, $H$ is set to 2, corresponding to 2 final capsules. Because of adding squash function, the length of the output capsule is compressed from 0 to 1, representing the probability of a category.

Note that as the squash function is used to compress the capsules into (0,1) interval in dynamic routing, we use the length of the final capsule $\mathbf{v_i^F}$ to represent the probability of a category.

## 4 Experiments

### 4.1 Datasets

We conduct extensive experiments on 2 benchmark news datasets in Chinese: BiendataFake and CHECKED. The basic statistics of the dataset are summarized in Table 1.

**BiendataFake**: This dataset is released by ICT (Institute of Computer Technology, Chinese Academy of Sciences) and BAAI (Beijing Academy of Artificial Intelligence), which are published in the competition platform - biendata[5].

**CHECKED**: This dataset is the first Chinese COVID-19 fake news dataset [20] based on the Weibo platform and we adopt the dataset from on github[6].

Table 1: Analysis of statistical information to the news

| Dataset | Language | #item | | | #category |
|---|---|---|---|---|---|
| BiendataFake | Chinese | fake:19,285 | real:19,378 | total:38,663 | 2 |
| CHECKED | Chinese | fake:344 | real:1,759 | total:2,103 | 2 |

### 4.2 Baselines

In order to comprehensively evaluate the effectiveness of our model, we compare it with the following baselines:

[5] https://www.biendata.xyz/competition/falsenews/

[6] https://github.com/cyang03/checked

**TextCNN** [10] is a CNN-based neural network model, the core point of CNN is that it can capture the local correlation of information.

**TextRNN** [12] is an RNN-based neural network model, due to the ability of memory function of RNN, TextRNN can tackle long text better.

**TextRCNN** [11] replaces the convolutional layer with a bidirectional RNN on the basis of TextCNN.

**DPCNN** [9] uses the structure of deep word-level CNN for text classification.

**Bert** [3] is a language model built on a bi-directional Transformer, the first token of every sequence is used as the feature representation for classification by a fully connected softmax layer.

**BertCNN** takes Bert as an embedder and is followed by CNN as a classifier.

**BertCNN*** is an enhancement we have done based on BertCNN, not only taking account of max-pooling but also average-pooling. It uses a combination of the feature representations derived from two kinds of pooling operations for classification.

### 4.3 Experimental Results and Analysis

In this section, we investigate whether the classification performance can be improved by implementing our method. Table 2 shows the experimental results.

As shown in Table 1, the CHECKED has the problem of data imbalance, so the effect of HACK on CHECKED is not as obvious as that on BiendataFake. Because the amount of data of biendataFake is much larger than that of CHECKED, we divide biendataFake in the ratio of 0.1: 0.6: 1, where the division ratio 0.1 is to compare the experimental effect when the amount of data of biendataFake and CHECKED is similar, 0.6 and 1 are to test whether the model effect will be affected as the size of data increases.

Table 2 shows that CNN-based models perform better than RNN-based models on the whole in the fake news issue. The results show that the performances of RNN-based models are inferior to that of the CNN-based models, which verifies that CNN-based models are better suited for this issue. Bert, as a transformer-based model, which is good at integrating text context by self-attention mechanism, when takes it as an embedder and combines it with CNN, it can perform better. For the structure, BertCNN is similar to TextCNN, but BertCNN makes significant improvements over the latter, proving that Bert can construct better feature representations than the usual word embedding model does. It also performs better than Bert, proving the advantage of N-gram convolution and Max-pooling on feature extraction. BertCNN* further improves BertCNN, illustrating that Average-Max pooling can tie up with Max-pooling to help abstract the features. At last, we observe that HACK outperforms all of the above methods. The comparison results show that instead of the output of a pre-trained language model or CNN-based feature extraction, a capsule network in the third level of our framework can better use a combination of local features in different positions to classify the fake news while preserving spatial information to the maximum.

Table 2: Test metrics results of different models on datasets. The best results are bold and the second-best results are underlined to easily compare.

| Models | BiendataFake | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | p | | | r | | | f1 | | |
| | 10% | 60% | 100% | 10% | 60% | 100% | 10% | 60% | 100% |
| TextCNN | 0.7835 | 0.8757 | 0.9057 | 0.7773 | 0.8751 | 0.9043 | 0.7757 | 0.8751 | 0.9043 |
| TextRNN | 0.8399 | 0.8561 | 0.8860 | 0.8342 | 0.8560 | 0.8816 | 0.8183 | 0.8560 | 0.8814 |
| TextRCNN | 0.8415 | 0.8680 | 0.8863 | 0.8360 | 0.8661 | 0.8863 | 0.8206 | 0.8661 | 0.8863 |
| DPCNN | 0.7992 | 0.8987 | 0.9177 | 0.7962 | 0.8920 | 0.9173 | 0.7959 | 0.8916 | 0.9172 |
| Bert | 0.8569 | 0.8724 | 0.8818 | 0.8553 | 0.8656 | 0.8723 | 0.8560 | 0.8689 | 0.8770 |
| BertCNN | 0.8754 | 0.8809 | 0.8981 | 0.8719 | 0.8755 | 0.8875 | 0.8736 | 0.8781 | 0.8927 |
| BertCNN* | 0.8818 | 0.8992 | 0.9130 | 0.8723 | 0.8890 | 0.9038 | 0.8770 | 0.8940 | 0.9083 |
| HACK | **0.9261** | **0.9310** | **0.9455** | **0.9179** | **0.9228** | **0.9396** | **0.9219** | **0.9268** | **0.9425** |

| Models | CHECKED | | |
|---|---|---|---|
| | p | r | f1 |
| TextRNN | 0.8648 | 0.8385 | 0.7721 |
| TextRCNN | 0.6901 | 0.8307 | 0.7539 |
| DPCNN | 0.6944 | 0.8333 | 0.7575 |
| Bert | 0.8993 | **0.8854** | **0.8598** |
| BertCNN | 0.7206 | 0.7142 | 0.7174 |
| HACK | **0.9117** | 0.8258 | 0.8452 |

## 5 Conclusion

In this paper, we solve the fake news issue by a hierarchical extraction framework, which can construct high-level features from low-level features. The novelty of our method is that with the capsule network, the local features can keep their spatial information by replacing neural nodes with capsule vectors, and the combination of pre-trained language model and capsule network can generate the sentence-level macroscopic feature representations by encoding spatial patterns.

## 6 Acknowledgement

This work is supported by National Science Foundation of China No. 61702216, 61772231, and Higher Educational Science and Technology Program of Jinan City under Grant with No. 2020GXRC057, 2018GXRC002.

## References

1. Chen, Z., Freire, J.: Proactive discovery of fake news domains from real-time social media feeds. In: Companion Proceedings of the Web Conference 2020. pp. 584–592 (2020)

2. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding (2018)
4. Dos Santos, C., Gatti, M.: Deep convolutional neural networks for sentiment analysis of short texts. In: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers. pp. 69–78 (2014)
5. Guo, L., Zhang, D., Wang, L., Wang, H., Cui, B.: Cran: a hybrid cnn-rnn attention-based model for text classification. In: International Conference on Conceptual Modeling. pp. 571–585. Springer (2018)
6. Hinton, G.E., Sabour, S., Frosst, N.: Matrix capsules with em routing (2018)
7. Jacovi, A., Shalom, O.S., Goldberg, Y.: Understanding convolutional neural networks for text classification. arXiv preprint arXiv:1809.08037 (2018)
8. Jin, Z., Cao, J., Zhang, Y., Luo, J.: News verification by exploiting conflicting social viewpoints in microblogs. In: Thirtieth Aaai Conference on Artificial Intelligence (2016)
9. Johnson, R., Zhang, T.: Deep pyramid convolutional neural networks for text categorization. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 562–570 (2017)
10. Kim, Y.: Convolutional neural networks for sentence classification. Eprint Arxiv (2014)
11. Lai, S., Xu, L., Liu, K., Zhao, J.: Recurrent convolutional neural networks for text classification. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 29 (2015)
12. Liu, P., Qiu, X., Huang, X.: Recurrent neural network for text classification with multi-task learning. arXiv preprint arXiv:1605.05101 (2016)
13. Liu, Y., Wu, Y.F.: Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32 (2018)
14. Monti, F., Frasca, F., Eynard, D., Mannion, D., Bronstein, M.M.: Fake news detection on social media using geometric deep learning. arXiv preprint arXiv:1902.06673 (2019)
15. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. arXiv preprint arXiv:1802.05365 (2018)
16. Przybyla, P.: Capturing the style of fake news. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 490–497 (2020)
17. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training. URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf (2018)
18. Rubin, V.L., Conroy, N., Chen, Y., Cornwell, S.: Fake news or truth? using satirical cues to detect potentially misleading news. In: Proceedings of the second workshop on computational approaches to deception detection. pp. 7–17 (2016)
19. Xu, C., Paris, C., Nepal, S., Sparks, R.: Cross-target stance classification with self-attention networks. arXiv preprint arXiv:1805.06593 (2018)
20. Yang, C., Zhou, X., Zafarani, R.: Checked: Chinese covid-19 fake news dataset. arXiv preprint arXiv:2010.09029 (2020)