Applied Intelligence (2023) 53:18971–18988
https://doi.org/10.1007/s10489-023-04455-1

Intra-graph and Inter-graph joint information propagation network with third-order text graph tensor for fake news detection

Benkuan Cui[1] · Kun Ma[1,2] · Leping Li[1,2] · Weijuan Zhang[3] · Ke Ji[1,2] · Zhenxiang Chen[1,2] · Ajith Abraham[4]

Accepted: 3 January 2023 / Published online: 15 February 2023

Abstract

Although the Internet and social media provide people with a range of opportunities and benefits in a variety of ways, the proliferation of fake news has negatively affected society and individuals. Many efforts have been invested to detect the fake news. However, to learn the representation of fake news by context information, it has brought many challenges for fake news detection due to the feature sparsity and ineffectively capturing the non-consecutive and long-range context. In this paper, we have proposed Intra-graph and Inter-graph Joint Information Propagation Network (abbreviated as IIJIPN) with Third-order Text Graph Tensor for fake news detection. Specifically, data augmentation is firstly utilized to solve the data imbalance and strengthen the small corpus. In the stage of feature extraction, Third-order Text Graph Tensor with sequential, syntactic, and semantic features is proposed to describe contextual information at different language properties. After constructing the text graphs for each text feature, Intra-graph and Inter-graph Joint Information Propagation is used for encoding the text: intra-graph information propagation is performed in each graph to realize homogeneous information interaction, and high-order homogeneous information interaction in each graph can be achieved by stacking propagation layer; inter-graph information propagation is performed among text graphs to realize heterogeneous information interaction by connecting the nodes across the graphs. Finally, news representations are generated by attention mechanism consisting of graph-level attention and node-level attention mechanism, and then news representations are fed into a fake news classifier. The experimental results on four public datasets indicate that our model has outperformed state-of-the-art methods. Our source code is available at https://github.com/cuibenkuan/IIJIPN.

Keywords Fake news detection · Text classification · Feature engineering · Information propagation · Graph neural network

1 Introduction

The rise of online social media, such as Twitter, has offered increasingly huge opportunities for fake news production, dissemination and consumption [1, 2]. Fake news refer to those stories that are false: the story itself is fabricated with no verifiable facts, sources or quotes [3]. It often has the aim of misleading public opinion, disturbing social order, and damaging the credibility of social media [4]. With the pandemic of COVID-19, related fake news are spreading all over the sky throughout the social media [5]. Fake news not only threaten the physical health but also affect the mental health of the public with restless anxiety or fear [6]. Therefore, it is urgent to develop effective methods to prevent fake news from causing serious negative effects.

Current researches in the fake news detection aim to determine the truthfulness of a given claim [7]. Among the existing methods of fake news detection, content-based methods verify the validity of news by headlines and body contents [8]. Besides, content-based approaches learn latent textual or visual feature representation of fake news by context information [9]. However, there are two major challenges for learning the representation of fake news by context information:

- **Challenge 1: Feature sparsity of text**. News item, such as tweets, headlines and online news, has become an important form to share information [10]. News text is usually composed of one or

Kun Ma contributed equally to this work.

✉ Kun Ma
ise_mak@ujn.edu.cn

Extended author information available on the last page of the article.

few short sentences, which lack the context of the syntactic structure [11, 12]. Recently, Graph Neural Networks (GNNs) have been widely used to learn text representations to solve the sparsity problem of text features in text classification [13]. Heterogeneous Graph ATtention networks (HGAT) embeds the Heterogeneous Information Network (HIN) for short text classification based on node-level and type-level attentions [14]. Short Text Graph Convolutional Network (STGCN) considers the word and sentence representations generated by the word co-occurrence, document word relations and text topic information as the classification feature [15]. These GNNs can capture semantic and sequential features in local consecutive word sequences well, but ignore global information between words of the news, and also fail to capture the relations of the features in news [16].

- **Challenge 2: Ineffectiveness to capture the non-consecutive and long-range word dependency**. Most of current deep learning methods only capture local features (messages or word-level syntactic) in a small window, which leads to ineffectively capture the non-consecutive and long-range semantic relations among words [13]. These methods ignore global word co-occurrence in a corpus which carries non-consecutive and long-distance semantics [17]. Recently, GNNs have attracted wide attention, they do not consider the text as a sequence but as a set of co-occurrence words [18]. Text Graph Convolutional Network (TextGCN) builds a heterogeneous word document graph for the whole corpus and turns document classification into a node classification [17]. Tensor Graph Convolutional Networks (TensorGCN) proposes text graph tensor constructed by contextual information and two kinds of propagation learning [19]. However, worse performance for GCNs when integrating high-order information via stacking convolutional layers [7].

In order to address the aforementioned challenges, this paper has proposed Intra-graph and Inter-graph Joint Information Propagation Network (abbreviated as IIJIPN) with Third-order Text Graph Tensor for fake news detection, which utilizes variety relations between words and allows information to be disseminated within and among graphs successively. Particularly, data augmentation is utilized to solve the problem of data imbalance and the shortage of labeled data in corpus. We present Third-order Text Graph Tensor (abbreviated as TTGT) with sequential, syntactic, and semantic features to describe contextual information. Then, we propose Intra-graph and Inter-graph Joint Information Propagation (abbreviated as IIJIP) to propagate homogeneous and heterogeneous information respectively. Finally, news representation is generated by attention mechanism which consists of graph-level attention and node-level attention mechanism, and then news representations are fed into a fake news classifier. The main contributions of our work can be summarized as follows:

- **Third-order Text Graph Tensor (abbreviated as TTGT)**. Third-order Text Graph Tensor with three features is proposed to capture contextual information at different language properties. Sequential-based, syntactic-based, and semantic-based text graphs are constructed to form a text graph tensor. Text sequential features are extracted with point-wise mutual information to depict the property of local word co-occurrence language; text syntactic features are described with word dependencies to depict the language property in the rules of a formal grammar; text semantic features are represented with topics and topic-related key words to depict the language property of the meaning of word. Compared with other similar works that only incorporate independent sequential or semantic information for word embedding learning, our method has proposed Third-order Text Graph Tensor with sequential, semantic and syntactic information jointly learning of heterogeneous graphs. Besides, our work also proposes novel constructions of the text graph weight for sequential, syntactic and semantic features, and a novel method of semantic information extraction with topics and topic-related key words.
- **Intra-graph and Inter-graph Joint Information Propagation (abbreviated as IIJIP)**. To encode the heterogeneous information from multi-graphs, two levels of information propagation are proposed in IIJIP: intra-graph information propagation is firstly performed in each graph to realize homogeneous information interaction by building graph for each text property, and high-order homogeneous information interaction in each graph can be achieved by stacking propagation layer; inter-graph information propagation is then performed among text graphs to realize heterogeneous information interaction, and by connecting the nodes across the graphs, the heterogeneous information in one graph can be gradually fused into other graphs. To the best of our knowledge, it is the first time to implement high-order homogeneous information interaction in each graph firstly, and heterogeneous information interaction among graphs successively.

The organization of this paper is as follows. Section 2 provides a brief description of the existing work in the area of fake news detection. Section 3 introduces our method. Firstly, a brief description of data normalization and data augmentation are provided. Secondly, the overall architecture of our method is introduced in detail. Section 4 shows the experimental results on four datasets with detailed analysis. Section 5 summarizes our work and briefly describes future work.

2 Related work

Fake news detection can be considered as a binary text classification that judges a piece of news is true or not [20]. In this section, we first review related work on text classification and then detail the methods for fake news detection.

2.1 Text classification

2.1.1 Machine learning methods

Text classification refers to the process of assigning predefined labels to documents [21]. Numerous works have been proposed for text classification. Traditional classification methods are based on statistical learning models [21]. A Bag-Of-Word(BOW) model is proposed by supplementing rare information with related terms [22]. A Vector Space Model (VSM) improves the term weighting schemes for short text classification [23]. However, these traditional methods disregard the contextual information in textual data and ignore the semantic information of the words.

2.1.2 Deep learning methods

Compared with traditional classification methods, deep learning methods automatically provide semantic representations by text mining. TextCNN utilizes the convolution layers with filters to extract local features from text for text classification [24]. However, it focuses only on the locality of words and thus lacks of the long-distance and non-consecutive word interactions. Bidirectional Gate Recurrent Unit can make up for the deficiency of Convolutional Neural Network (CNN) in extracting semantic information from long text. Self-Attention-Based BiLSTM utilizes Bi-directional Long Short-Term Memory (Bi-LSTM) to obtain context-related semantic features for classifying short texts [25]. Attention mechanism can strengthen the learning of long-distance dependency by assigning different weights to different words. Attention-based bidirectional long short-term memory with convolution layer (AC-BiLSTM) is proposed to address the issues of the high dimensionality and sparsity of text data. It captures both the local feature of phrases and global sentence semantics [26]. Heterogeneous Graph Attention networks (HGAT) is proposed to embed the HIN for short text classification with both node-level and type-level attention mechanism to address the issue of the semantic sparsity [14]. The shortcoming of these deep learning methods is that they cannot distinguish the contribution of each word in an entire sentence.

2.1.3 Graph neural networks

Graph Neural Networks have obtained better performance on text classification by encoding syntactic structure of sentences. TextGCN constructs a graph for the whole corpus with global relations between documents and words [17]. However, the contextual-aware word relations within each document are neglected. As an improvement, TextING [18] builds the individual graph for each document with sequential information and performs text-level word interactions. Apart from TextING, a principled model called hypergraph attention networks (HyperGAT) is proposed for modeling each text document as document-level hypergraphs with node-level attention and edge-level attention to capture the high-order word interactions [27]. Inductive text representation learning in HyperGAT is implemented to eliminate the mandatory access of test documents during training to reduce computational consumption. SHINE is another GNN for short text classification. It models the text dataset as a hierarchical heterogeneous graph, and learns the document graph performing the label propagation among texts dynamically [16]. However, these methods only consider sequential information rather than semantic information.

2.2 Fake news detection

2.2.1 Deep learning methods

Deep learning methods such as Recurrent Neural Networks (RNN), CNN and LSTM have been used for fake news detection. A Deep CNN extracts a number of discriminatory features at each layer for fake news detection [28]. Semantic information in the text are extracted by an improved RNN for fake news detection [29]. Polysemy proliferates in natural language, however, deep learning methods cannot exactly distinguish the concrete meaning in the context.

2.2.2 Pre-trained language models

Pre-trained language models, such as Bidirectional Encoder Representations from Transformers (BERT) [30], can learn the meaning of the word depending on the context accurately. Better performance can be achieved by fine-tuning the pre-trained models on fake news classification. FakeBERT, a BERT-based deep learning approach, combines different parallel blocks of the single-layer deep CNN, which can address the issue of the ambiguity [31]. BERT-BiLSTM-Capsule utilizes BERT to obtain word encoding to solve the problem of sentence-level propaganda classification [32]. Albert, increasing the training speed of BERT, incorporates factorized embedding parameterization and cross-layer parameter sharing to reduce parameters [33]. An ensemble method of three transformer models (BERT, ALBERT, and XLNET) is proposed to analyze and detect COVID-19 fake news [34]. However, pre-trained methods perform poorly for the specific domains because professional phrases are rare in universal corpora.

2.2.3 Graph neural networks

Compared with pre-trained methods, GNNs have successfully described complex relationships and interdependencies with graphs [35]. Propagation patterns and social context are introduced for identifying the fake news [2, 8, 13, 36]. A propagation-based method is proposed to integrate the GNNs to distinguish the propagation patterns between the fake news and real news [8]. Explainable FakE News Detection (dEFEND) on social media exploits both news contents and user comments jointly to capture explainable user comments with a sentence-comment co-attention sub-network [36]. Knowledge-driven Multimodal Graph Convolutional Network (KMGCN) exploits multi-modal content information and additional background knowledge for fake news detection [13]. Temporally Evolving Graph Neural Network for Fake news detection (TGNF) captures dynamic evolution patterns of news propagation from the perspective of continuous time [2]. However, these methods mainly focus on multi-modal posts, and they are not appropriate for plain text news in social media.

3 Method

Giving a tweet about a news, the main goal of our approach is to infer whether it is a fake news or not based only on the text. In this section, we propose Intra-graph and Inter-graph Joint Information Propagation Network (abbreviated as IIJIPN) with Third-order Text Graph Tensor for fake news detection. IIJIPN jointly explores text feature extraction, information propagation and attention mechanism. The overall architecture of IIJIPN is shown in Fig. 1. Architecture of IIJIPN includes four parts:

1. **Third-order Text Graph Tensor (abbreviated as TTGT)**. Sequential, syntactic, and semantic features are utilized to describe contextual information.
2. **Intra-graph and Inter-graph Joint Information Propagation (abbreviated as IIJIP)**. It is used for realizing homogeneous and heterogeneous information interaction respectively.
3. **Graph-level and Node-level attention mechanism**. It captures both the importance of text graphs and nodes respectively.
4. **Classifier**. It is used to generate the tag of news.

Giving a news $x = (e_1, e_2, \cdots, e_n)$, it contains n words. Firstly, text sequential feature, syntactic feature, and semantic feature are extracted in x. Then, sequential graph G_{seq}, syntactic graph G_{syn}, and semantic graph G_{sem} are built. Secondly, the information propagation within the graph and between the graphs are performed, the context interaction of homogeneous and heterogeneous information are realized, and the graph representation is updated to get

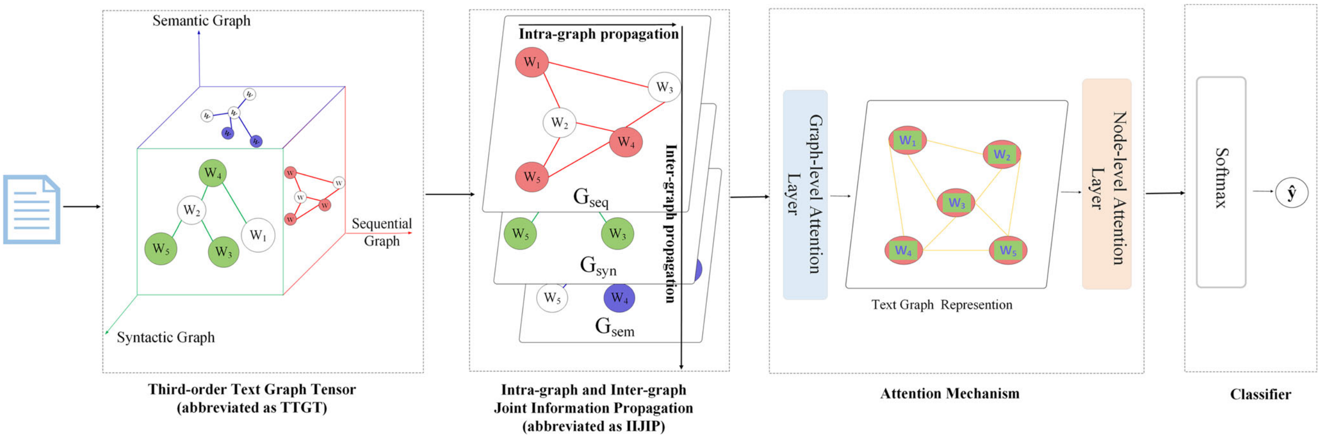

Semantic Graph
Sequential Graph
Syntactic Graph
W₁
W₂
W₃
W₄
W₅
Third-order Text Graph Tensor (abbreviated as TTGT)
Intra-graph propagation
Inter-graph propagation
G_seq
G_syn
G_sem
Intra-graph and Inter-graph Joint Information Propagation (abbreviated as IIJIP)
Graph-level Attention Layer
Text Graph Represention
Node-level Attention Layer
Attention Mechanism
Softmax
ŷ
Classifier

Fig. 1 Architecture of Intra-graph and Inter-graph Joint Information Propagation Network (abbreviated as IIJIPN) with Third-order Text Graph Tensor for fake news detection

G''_{seq}, G''_{syn}, and G''_{sem}. Then, the graph attention mechanism calculates the final graph representation G_f, and sends G_f to the node attention mechanism obtaining the node representation w_f. Finally, w_f is sent to the classifier to predict the classification label $\hat{y}$ of x.

3.1 Text normalization

Since there is noisy content including punctuation and irrelevant characters in the dataset, we firstly normalize text as follows:

1. **Text cleaning**: Removing HTML, punctuation, special symbols and any other characters.
2. **Tokenization**: Turning a meaningful piece of the whole text into a number of words.
3. **Remove stop words**: Removing the prepositions, pronouns, conjunctions, and auxiliary words that are used frequently but have nothing to do for the text.
4. **Stemming and Lemmatization**: Identifying the common root form of a word by removing or replacing word suffixes (e.g. "played" is stemmed as "play", and "apples" is stemmed as "apple").
5. **To lowercase**: Converting all characters to lowercase.

3.2 Data augmentation

In this work, experiments and analysis are based on four datasets including Liar, Constraint, Twitter15, and Twitter16. During the process of training model, one problem in Liar is data imbalance where at least one of the target variable values has a smaller number of instances when compared with the other values [37], and there is a ratio of 2.6:1 between true news and fake news in Liar. Another problem in Twitter15 and Twitter16 is smaller corpus where there is the shortage of training data. Twitter15 has 742 news, and Twitter16 only has 412 news. The performance of model is affected by the size and quality of the data [38]. We concentrate on adjusting the number of news to address above two issues: Increasing the number of fake news to make the number of fake news equal to the true news can solve the problem of data imbalance in Liar; Increasing the number of news can solve the problem of small corpus.

A method for data augmentation is proposed. It consists of two text editing operations: synonym replacement (SR, replacing word with one of its synonyms chosen at random), random insertion (RI, find a random synonym of a random word and then insert that synonym into a random position in the sentence). Giving a sentence, each operation can generate a variant sentence. Examples of augmented sentences are shown in Table 1. In the process of model training, SR and RI are used to generate two variant sentences for each sentence, aiming to by adjusting the number of training sets to solve the problem of data imbalance in Liar and small corpus in Twitter15 and Twitter16.

Table 1 The original sentence and generated sentences in data augmentation

Operation	Sentence
None	The economy is creating jobs at the fastest pace.
SR	The economy is creating jobs at the **quick** pace.
RI	The economy is creating **libertine** jobs at the fastest pace.

The bold parts are the optimal results of different experiments

3.3 Third-order text graph tensor

Tensor refers to a multi-dimensional array of numbers, called an $n - way$ or $n - mode$ array [39]. Vectors and matrices are first-order and second-order tensors respectively. $A \in R^{n_1 \times n_2 \times n_3}$ is a third-order tensor where *order* is the number of ways or modes of the tensor.

A series of graphs can be utilized to depict text from different properties. We focus on three features within the text: sequential feature, syntactic feature, and semantic feature. Sequential-based, syntactic-based, and semantic-based text graphs are constructed to create a Third-order Text Graph Tensor T, denoted as $T = (G_1, G_2, G_3)$. $G_i(i = 1, 2, 3)$ is the ith text graph in the Third-order Text Graph Tensor T. Given $G_i = (V_i, E_i, A_i, X_i)$, where $V_i(|V_i| = n)$ is the set of the ith graph nodes, E_i is the set of the ith graph edges, A_i is the ith graph adjacency matrix, and X_i is the ith graph feature matrix. All graph adjacency matrices are packed into the graph adjacency tensor $\Lambda = (A_1, A_2, A_3) \in R^{3 \times n \times n}$. All graph feature matrices are packed into a graph feature tensor $\Upsilon = (X_1, X_2, X_3) \in R^{3 \times n \times m}$, where m is the dimension of the feature vector of each node.

All graph feature matrices X_i are uniformly initialized with *Glove*. Following subsections are focus on how to build E_i and A_i for each G_i.

3.3.1 Sequential-based graph

Sequential information depicts the local co-occurrence language property among words, which has been widely used for text representation learning. In this study, Pointwise Mutual Information (PMI) [19] is utilized to describe the sequential context using sliding window strategy. In Sequential-based Graph G_{seq}, the edge weight of each pair of words is estimated by

$$d_{sequential}(e_i, e_j) = log \frac{p(e_i, e_j)}{p(e_i)p(e_j)} \tag{1}$$

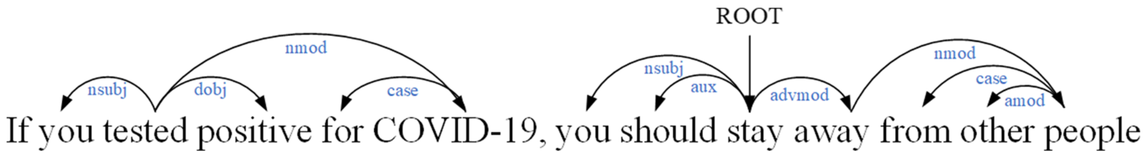

ROOT
nsubj
dobj
nmod
case
nsubj
aux
advmod
nmod
case
amod
If you tested positive for COVID-19, you should stay away from other people

Fig. 2 An example of word dependency in one sentence. Each word dependency is a directed triplets consisting of the name of the relation, governor and dependent. In our work, we mainly focus on the word pair (governor, dependent). In Syntactic-based Graph, we define the weight of each pair of words as: E(governor,dependent) = 1. For example, E(tested,COVID-19)=1, E(stay,away) = 1 and so on

where $p(e_i, e_j)$ shows the probability of occurrence of the word pair (e_i, e_j) in a news record x, which is calculated by $\frac{N_{co-occurence}(e_i,e_j)}{N_{windows}}$, where $N_{windows}$ is the total number of sliding windows in x and $N_{co-occurence}(e_i, e_j)$ is the number of sliding windows that the word pair (e_i, e_j) co-occurs in x. $p(e_i) = \frac{N_{occurrence}(e_i)}{N_{windows}}$ is the probability that the word e_i occurs in x, and $N_{occurrence}(e_i)$ is the total number of sliding windows in the text that contains e_i.

3.3.2 Syntactic-based graph

Syntactic analysis, defined as syntax analysis or parsing, is the process of analyzing natural language with the rules of a formal grammar. We utilize word dependency produced by Stanford CoreNLP [40] to describe the syntactic feature. The word dependency provides a representation of grammatical relations between words in a sentence. Figure 2 shows an example of word dependency in one sentence. Different word dependency can be extracted in one sentence, each word dependency is a directed triplets consisting of the name of the relation, governor and dependent. In our model, we mainly focus on the word pair $(governor, dependent)$. In Syntactic-based Graph G_{syn}, we define the weight of each pair of words as: $E(governor, dependent) = 1$.

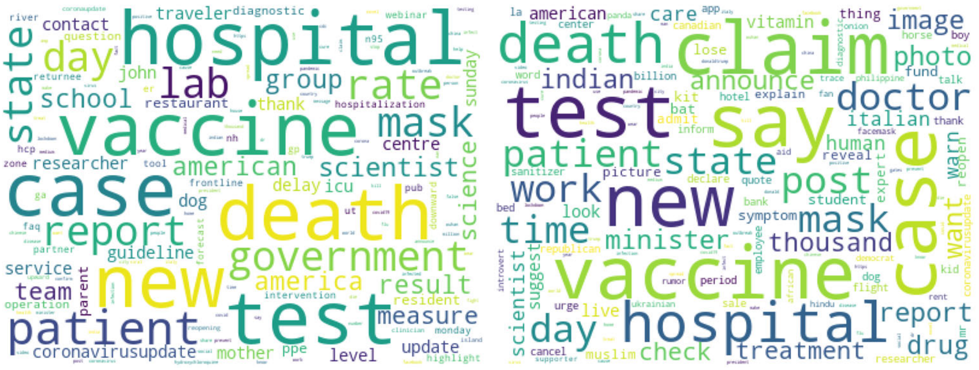
3.3.3 Semantic-based graph

Furthermore, in order to enrich the context for each word, we extract semantic feature by capturing topic-related correlations between words. The implementation steps are as follows:

1. We mine the latent topics $\tau = (t_1, t_2, \cdots, t_k)$ (k is the total number of topics) from the whole corpus using Latent Dirichlet Allocation (LDA) [41].
2. For each topic t_i, it can be represented with the probability distribution over the words $t_i = (e_1^i, e_2^i, e_3^i, \cdots)$.
3. To further represent the topic t_i, we consider relating top M key words with the largest probabilities in the corpus $t_i = (e_1^i, e_2^i, \cdots, e_M^i)$.

We apply extracting the topic and related key words into each text to numerically represent the semantic relations. Giving a news $x = (e_1, e_2, \cdots, e_h, \cdots, e_n)$, corresponding to topic $t_j = (e_1^j, e_2^j, \cdots, e_M^j)$. When constructing the Semantic-based Graph G_{sem}, if word $e_h \in (e_1^j, e_2^j, \cdots, e_M^j)$, the weight of word pair (e_i, e_j) is defined as:

$$E_{ij} = \begin{cases} 1 \ \text{i} = \text{h}, \text{j} \in (1, 2, \cdots, n)\text{andj} \neq \text{i} \\ 0 \ \text{else} \end{cases} \tag{2}$$



(a) The keyword distribution of topic of true news in Constraint dataset. (b) The keyword distribution of topic of fake news in Constraint dataset.

Fig. 3 The keywords distribution of topics in Constraint dataset. We select a word (e.g. "death") belonging to the topics in Fig. 3(a) from the sentence "Suffering COVID-19 may leads to death". Then, we construct the relationship between "death" and other words in this sentence. When building semantic graph for this news, the weight of edges in the semantic feature graph includes $E_{(death,Suffering)} = 1$, $E_{(death,COVID-19)} = 1$, $E_{(death,may)} = 1$, $E_{(death,leads)} = 1$, $E_{(death,to)} = 1$

Figure 3 shows an example about how to build semantic feature graph for a news.

3.4 Intra-graph and inter-graph joint information propagation

Third-order Text Graph Tensor T is utilized to perform Intra-graph and Inter-graph Joint Information Propagation. Formula (3) indicates the learning process:

$$T \xrightarrow{f_{intra}} T_{intra} \xrightarrow{f_{iner}} T_{inter} \tag{3}$$

where $T = (G_{seq}, G_{syn}, G_{sem})$, f_{intra} and f_{inter} denote the intra-graph propagation and inter-graph propagation, respectively. The learning details of the two kinds of propagation are described next.

3.4.1 Intra-graph information propagation

Intra-graph information propagation means that nodes interact message with other nodes and update their representation within graph (see Fig. 4). Gated Graph Neural Networks [42] is firstly used to learn the word feature vector. Nodes can receive the information from its neighbours and merge it into own vector to update itself. As the propagation layer only works on the first-order neighbours, such layer can be stacked t times to achieve high-order feature interaction, where a node can receive another nodes information which far from t hops away. The formulas of the interaction are:

$$a^t = \Lambda w^{t-1} W_a \tag{4}$$

$$z^t = \sigma(W_z a^t + U_z w^{t-1} + b_z) \tag{5}$$

$$r^t = \sigma(W_r a^t + U_r w^{t-1} + b_r) \tag{6}$$

$$l^t = tanh(W_h a^t + U_h(r^t \odot w^{t-1}) + b_h) \tag{7}$$

$$w^t = l^t \odot z^t + w^{t-1} \odot (1 - z^t) \tag{8}$$

where $w^{(0)}$ is Υ, σ is the sigmoid function, and all W, U and b are trainable weights and biases. z and r function as the update gate and reset gate respectively to determine to what degree the neighbour information contributes to the current node embedding.

After the Intra-graph information propagation, the $T = (G_{seq}, G_{syn}, G_{sem})$ is transformed into $T_{intra} = (G'_{seq}, G'_{syn}, G'_{sem})$.

3.4.2 Inter-graph information propagation

Different graphs maintain different properties depicting the given data. To comprehensively describe the data, it is necessary to perform the learning among different graphs. The inter-graph information propagation is to aggregate information between different graphs. The input of Inter-graph information propagation is the T_{intra}. The information from one graph can be gradually fused into other graphs (see Fig. 5). Specifically, after we perform intra-graph information propagation, we choose any two updated graphs to achieve inter-graph word interaction. The formulas of the interaction are as follows:

$$G''_{seq} = G'_{seq} + G'_{syn} \tag{9}$$

$$G''_{syn} = G'_{syn} + G'_{sem} \tag{10}$$

$$G''_{sem} = G'_{sem} + G'_{seq} \tag{11}$$

After the Inter-graph information propagation, the $T_{intra} = (G'_{seq}, G'_{syn}, G'_{sem})$ is transformed into $T_{inter} = (G''_{seq}, G''_{syn}, G''_{sem})$.

3.5 Attention mechanism

We also propose the graph attention mechanism and node attention mechanism to measure the importance of text

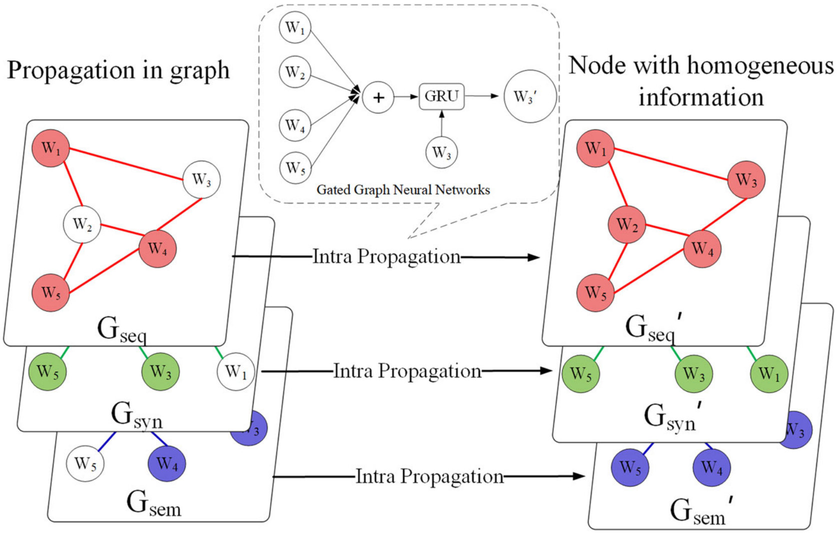


Fig. 4 Intra-graph information propagation is firstly conducted in each graph to interact homogeneous information, every word nodes updates itself from its neighbours and then merge with its own representation to update itself, and high-order neighbours information interaction in each text graph can be achieved by stacking propagation layer

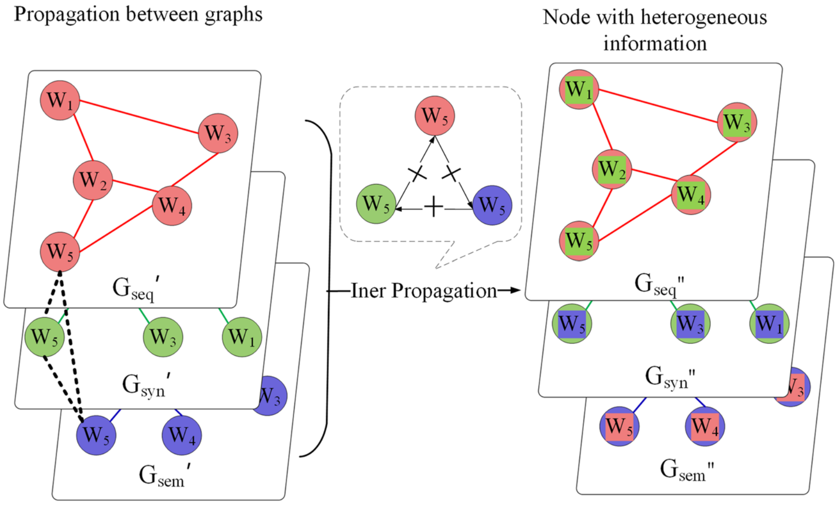


Fig. 5 Inter-graph information propagation harmonize heterogeneous information using the output of intra-graph propagation. By connecting the nodes across the graphs, the heterogeneous information in one graph can be gradually fused into other graphs

graphs and nodes. The details of the two kinds of attention mechanisms are described next.

3.5.1 Graph attention mechanism

Graph attention mechanism is introduced to learn the weights to measure the importance of each graph $G_t \in T_{inter} = (G''_{seq}, G''_{syn}, G''_{sem})$, and the final graph G_f can be computed as follows:

$$G_f = \sum_{t=1}^{3} \beta_t G_t \tag{12}$$

Where β_t measures the importance of t-th graph for the final graph G_f, and β_t is calculated as follows:

$$u_t = tanh(W_t G_t + b_t) \tag{13}$$

$$\beta_t = \frac{exp(u_t)}{\sum_{k=1}^{3} exp(u_k)} \tag{14}$$

Where u_t is the hidden representation of G_t obtained by feeding the G_t to a fully embedding layer, and W_t, b_t are trainable weights and biases.

3.5.2 Node attention mechanism

When we get the final text graph G_f, the word nodes are sufficient updated, which can be used to produce the final prediction. Because not all nodes play an equal important role in classification, we hope the key words should contribute to more explicitly. We define the node attention mechanism as:

$$e_i = \sigma(f_1(e_i)) \odot tanh(f_2(e_i)) \tag{15}$$

$$w_f = \frac{1}{|n|} \Sigma_{i \in n}^{|V|} e_i + Maxpooling(e_1, e_2, \cdots, e_n) \tag{16}$$

Where f_1 and f_2 are two multilayer perceptions. The former is used as a soft attention weight and the latter as a non-linear feature transformation. In addition to average the weighted word features, we also apply a Max-pooling function for the word representation w_f.

3.6 Final prediction

Finally, the label of text is predicted by feeding the word representation w_f into a softmax layer. We minimize the loss through the cross-entropy function:

$$\hat{y} = softmax(W w_f + b) \tag{17}$$

$$\mathcal{L} = -\Sigma_i y_i log(\hat{y}_i) \tag{18}$$

Where W and b are weights and bias, and y_i is the i-th the element of the one-hot label.

4 Experiments

In this section, we aim at evaluating the performance of our model.

4.1 Datasets

We use four datasets including Liar, Constraint, Twitter15, and Twitter16 (see Table 2) to perform experiments and analysis.

Liar [43]: It is a public dataset which includes over 12000 labeled short news. The news in the Liar dataset is classified into six categories for the truthfulness ratings: pants-fire, false, barely-true, half-true, mostly-true, and true. In our work, we mainly focus on classifying news as real and fake. For the binary classification, we divide these types into two types. Pants-fire and false are categorized as fake, barely-true, half-true, mostly-true, and true are as true. Although the dataset also provides some additional meta-data like the

Table 2 Summary statics of datasets

Dataset	Category	True/Fake number	Training/Validation/Testing number	Average length
Liar	2	9275/3552	10269/1284/1283	17.95
Constraint	2	5600/5100	8560/1070/1070	28.64
Twitter15	2	372/370	594/74/74	15.62
Twitter16	2	207/205	330/40/42	14.96

subject, speaker, job, et al., only textual content is used in our work.

Constraint [5]: The dataset for Constraint@AAAI2021 COVID-19 Fake New Detection in English provided by the organizers on the competition websites. The Constraint dataset contains fake and real news about COVID-19 in the form of social media posts such as Twitter, Facebook, Instagram, etc. As for the 10,700 social media posts in dataset, 8,560 posts are preserved for training, 1,070 posts are used in the validation phase, and the remain 1,070 posts are kept for testing.

Twitter15 [44]: The news in dataset is crawled from the two rumor tracking websites, namely, $snopes.com$ and $emergent.info$. The dataset consists of total 942 events, but in our work, we only use the verified 370 fake news and 372 true news.

Twitter16 [45]: The news is divided into rumors and non-rumors from $www.snopes.com$, an online rumor debunking service. They obtain 778 reported events during March-December 2015, of which 64% are rumors. At the same as Twitter15, we also use verified 205 fake news and 207 true news.

4.2 Baselines

To evaluate the performance of proposed methods, we compare some state-of-art methods with our model:

1) Machine learning methods:

TF-IDF+SVM: Term frequency–inverse document frequency (TF-IDF) is used to feature extraction, and Support Vector Machine (SVM) as the classifier.

2) Deep learning Methods:

TextCNN [24]: Applying Convolution Neural Networks into Natural Language Processing, and processing text classification.

Bi-LSTM: Bi-directional Long Short-Term Memory, is one of the variants of Recurrent Neural Networks, which does well in processing Long-Term Dependencies.

3) Pre-trained Methods:

Albert [33]: Albert, one of the variations of BERT, incorporates factorized embedding parameterization and cross-layer parameter sharing to reduce parameters.

Roberta [46]: A Robustly Optimized BERT Pretraining Approach (Roberta), one of the variations of BERT, proposes a 'pre-training'+'post-training'+'fine-tuning' three-stage paradigm.

4) Topic Methods:

TMN [47]: Topic Model Networks for text classification, which can extract the topic and find the topic words with the support of Neural Topic Model and Topic Memory Mechanism.

5) Graph-based Methods:

HGAT [14]: Based on a dual-level attention level, Heterogeneous Graph Attention network(HGAT) can embed the heterogeneous information network for text classification.

TextGCN [17]: A Graph Convolution Networks for Text Classification, which models the whole corpus as a heterogeneous graph and learns the word and document embedding.

HyperGAT [27]: Hypergraph Attention Networks for Inductive Text Classification, sequential hyperedges and semantic hyperedges are used to describe Hypergraphs.

TextING [18]:Graph Neural Networks are used to inductive text classification, where building graph for each document and text word interactions can be performed in it.

4.3 Experiments sets

Due to different structure of above methods, we tune the hyperparameter by the performance on the validation set. Table 3 shows the hyperparameters of IIJIPN. We also show the differences of the configuration of IIJIPN and baseline models in Table 4.

Table 3 The hyperparameters of IIJIPN

Model	Hyperparameters

IIJIPN	Learning_rate = 0.001
	max_epoch = 200
	batch_size = 128
	input_dim = 300
	early_stopping_epoch = 20

Table 4 Differences of the configuration of IIJIPN and baseline models

Model	Hyperparameters
TextCNN	Kernel_size = [3,4,5]
	Hidden_layers = 2
Bi-LSTM	Hidden_size = 32
	Bidirectional = True
Albert	Version = albert-base-v2
Roberta	Version = roberta-base
TMN	Topic_num = 2
HyperGAT	Hidden_size = 100
TextING	Hidden_size = 64

4.4 Evaluation indicators

In this work, Accuracy(Acc), Precision(P), Recall(R) and F1-score($F1$) are used to evaluate the performance of the model. Liar is an imbalanced dataset, we also use BalancedAccuracy($BAcc$) to measure the performance of the models on the Liar datasets.

4.5 Results

We conduct comprehensive experiments to evaluate the performance on fake news classification. Overall, our model achieve state-of-the-art results, which effectively demonstrate the superior capability of detecting fake news.

Experimental results of liar dataset As shown in Table 5, IIJIPN achieves better performance than baseline models. In addition, it can be seen from the Table 5 that IIJIPN and other Graph-based models, such as TextING, have achieved better results than non-graph models, such as TextCNN. For Graph-based models, a single graph with nodes and edges is firstly built for each news. Each node represents different word which carries various information. Each node can capture information from whole graph instead only capturing the information from its neighbours. Information propagation is conducted in each graph, realizing long-distance and discontinuous word interaction, obtaining the associated information between words, and transferring the text semantics to each node to achieve the continuity of context information. However, graph-based methods such as TextING and TextGCN only consider the propagation of information within the graph. In addition to the propagation of information within the graph, we also consider the propagation of information between the graphs. Information propagation between graphs realizes the interaction of heterogeneous information. The experimental results show that when both intra-graph and inter-graph information propagation are simultaneously considered, the obtained experimental results are better than the model that only considers intra-graph information propagation.

Table 5 The performance on liar dataset

	Acc	BAcc	P	R	F1
SVM	0.7387	0.5148	0.7379	**0.9957**	0.8477
TextCNN	0.7040	0.5013	0.4286	0.0079	0.0155
Bi-LSTM	0.5342	0.5157	0.2990	0.4300	0.3528
Albert	0.7155	0.5585	0.7599	0.8948	0.8219
Roberta	0.7350	0.5495	0.7544	0.9469	0.8398
TMN	0.6698	0.5243	0.7553	0.8016	0.6605
HGAT	0.7227	0.5136	0.3614	0.5000	0.4195
TextGCN	0.7405	0.5381	0.7456	0.9809	0.8472
HyperGAT	0.6485	0.5722	0.7634	0.7545	0.7590
TextING	0.7256	0.5213	0.7481	0.9437	0.8346
IIJIPN	**0.8683**	**0.8587**	**0.8433**	0.9181	**0.8791**

The bold parts are the optimal results of different experiments

Experimental results of constraint dataset To avoid that the proposed model showing good performance only for Liar, we also conduct the second experiment using Constraint dataset. Table 6 shows the performance on Constraint. Although IIJIPN has not achieved the best performance, it still has the better performance than most models. Albert and Roberta get better performance. The pre-trained knowledge has significance on fake news detection, especially for Constraint (composed of COVID-19 events) and Liar dataset (composed of political news). Pre-trained language models contain large-scale annotated data and sufficient semantic understanding of domain-specific knowledge. In our model, both pre-trained word embedding *Glove* and random initialization are used to represent the word, which is not only fully expressing the familiar vocabularies but also describing new words without pre-trained tokens.

Table 6 The performance on Constraint Dataset

	Acc	P	R	F1
SVM	0.9116	0.9053	0.9285	0.9168
TextCNN	0.9075	0.8983	0.9088	0.9035
Bi-LSTM	0.8668	0.8777	0.8373	0.8570
Albert	0.9570	0.9439	**0.9759**	0.9596
Roberta	**0.9682**	0.9655	0.9741	**0.9698**
TMN	0.5322	0.6321	0.9480	0.4465
HGAT	0.6564	0.7180	0.6680	0.6394
TextGCN	0.5023	0.5230	0.5580	0.5400
HyperGAT	0.9425	**0.9677**	0.9098	0.9378
TextING	0.9393	0.9195	0.9688	0.9435
IIJIPN	0.9181	0.9232	0.9033	0.9131

The bold parts are the optimal results of different experiments

Table 7 The performance on Twitter15 dataset

	Acc	P	R	F1
SVM	0.9329	**0.9857**	0.8846	0.9324
TextCNN	0.8378	0.7812	0.8333	0.8065
Bi-LSTM	0.5541	0.4681	0.7333	0.5714
Albert	0.7973	0.8537	0.7955	0.8235
Roberta	0.8784	0.9268	0.8636	0.8941
TMN	0.9030	0.9385	0.8714	0.9030
HGAT	0.5886	0.5932	0.5886	0.5834
TextGCN	0.9155	0.9111	**0.9762**	0.9425
HyperGAT	0.9014	0.9231	0.8276	0.8727
TextING	0.8873	0.8387	0.8966	0.8667
IIJIPN	**0.9459**	0.9762	0.9318	**0.9535**

The bold parts are the optimal results of different experiments

Table 8 The performance on Twitter16 dataset

	Acc	P	R	F1
SVM	0.9286	0.9545	0.9130	0.9333
TextCNN	0.8500	0.9375	0.7500	0.8333
Bi-LSTM	0.6500	0.6667	0.6000	0.6316
Albert	0.8750	0.8947	0.8500	0.8718
Roberta	0.9250	0.9474	0.9000	0.9231
TMN	0.4392	0.2684	0.1131	0.3821
HGAT	0.7054	0.7345	0.7063	0.6966
TextGCN	0.8992	0.9767	0.8936	0.9333
HyperGAT	0.8992	0.6970	0.9200	0.7931
TextING	0.8403	0.9630	0.8298	0.8914
IIJIPN	**0.9683**	**0.9844**	**0.9545**	**0.9692**

The bold parts are the optimal results of different experiments

Experimental results of Twitter15 dataset To verify the performance of our model on none particular event datasets, we further conduct experiment using Twitter15 and Twitter16 Datasets. The experimental results of Twitter15 Dataset are as shown in Table 7, IIJIPN has the highest values of *Acc*. TMN has the better *Acc*, which exceeding some Graph-based models, such as TextING and HGAT. Both TMN and our model consider the extraction of semantic information when obtaining text features. And our model also considers the extraction of additional information such as sequential information and syntactic information. From experimental results, our model achieves better results than TMN. It indicates that the better performance cannot be achieved if only considering the semantic information. Different methods describe text documents at different language properties. TextGCN and TextING describe sequence context information, and HyperGAT describes both sequence and semantic context information. But our model describes all the sequence, semantic, and syntactic context information. Short news is composed of few words. For the fake new classification, which is insufficient if we only consider a single kind of information.

Experimental results of Twitter16 dataset Table 8 shows the performance on Twitter16 dataset, it can be seen that IIJIPN achieves the highest value in all evaluation indicators. Deep learning methods TextCNN has a good effect on fake news detection, and its *Precision* value is 0.9375. TextCNN has not achieved better results. The potential reason is TextCNN fails to solve the problem of global feature loss caused by pooling layer. At the same time, long-term dependence cannot be learned under the limitation of convolution kernel size. Stacking the convolution layer multiple times can increase the receptive field to increase the feature learning ability. However, excessive convolution layers will lead to a sharp increase in computation. To solve the problem of global feature loss, two measures are proposed in this work. Firstly, for each graph with single information, nodes can receive the information from its neighbours and merge it into own representation to update itself by propagation layer. As the propagation layer only works on the first-order neighbours, such layer has been stacked many times to achieve high-order feature interaction. Secondly, to comprehensively describe the data, we perform the learning among different graphs to aggregate information between different graphs.

There is the shortage of news in Twitter15 and Twitter16. Twitter15 has 742 news, and Twitter16 has only 412 news. These news are too small to effectively train the deep learning models. Better performance cannot be achieved when these deep learning models without sufficient training. Different from deep learning models, SVM is a powerful supervised algorithm that works well on smaller datasets. We introduce a simple text classification pipeline that contains stop words removal, features extraction with TF-IDF, SVM as classifier. In Tables 7 and 8, SVM achieves the excellent performance respectively. Particular words in datasets can be used in SVM to identify fake news.

4.6 Ablation analysis

Ablation analysis is conducted to investigate the contribution of components in our model.

4.6.1 Data augmentation analysis

We firstly examine and perform an analysis of the effectiveness of data augmentation. The results of without data augmentation and with data augmentation are presented in Table 9. It is clearly that data augmentation has helped tremendously for the classification on Liar, Twitter15, and Twitter16, especially for Liar.

Table 9 The results in data augmentation analysis IIJIPN(-Aug) indicates that IIJIPN without considering data augmentation

Model	Liar	Constraint	Twitter15	Twitter16
IIJIPN(-Aug)	0.7303	0.9173	0.8919	0.9524
IIJIPN	0.8683	0.9181	0.9459	0.9683

Data augmentation for liar The Liar dataset is imbalanced. $BAcc$ is an indicator to measure the performance of the model on the imbalanced dataset. For baseline models, the obtained value of $BAcc$ is within the range of [0.50,0.57]. The obtained value for IIJIPN is 0.8587, which is larger than the baseline models. And the value of $BAcc$ is close to Acc in IIJIPN.

Because the true news is more than the fake news in Liar dataset, it is easy for IIJIPN to learn the features of the true news sufficiently, but difficult to learn the features of the fake news. It is seen from Fig. 6 that numerous news are predicted to be true news. Data augmentation can solve the data imbalance in Liar. After applying data augmentation in Liar, the number of fake news is roughly as the same as true news. IIJIPN can sufficiently learn the characteristics of true news and fake news. In Fig. 7, the number of news which are predicted to be the correct label has been significantly increased, especially for fake news.

Data augmentation for twitter15 and twitter16 As for Twitter15 and Twitter16, the shortage of training news as the main bottleneck which leads to the poor results for IIJIPN(-Aug). After performing the data augmentation in Twitter15 and Twitter16, the number of total news has increased. Compared with the small corpus, IIJIPN can learn sufficient text features from large corpus which generated by data augmentation.

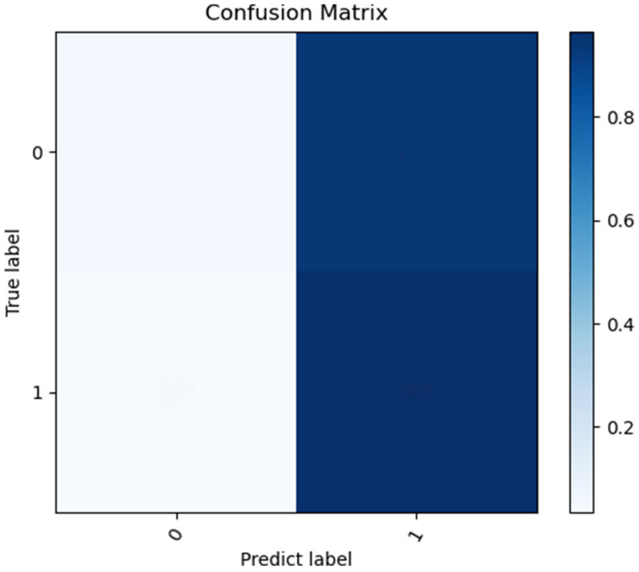


Fig. 6 The confusion matrix describes the results of Liar, which is generated by IIJIPN(-Aug). In the figure, 0 represents Fake label and 1 represents True label in the horizontal and vertical coordinates. Darker color represents higher value

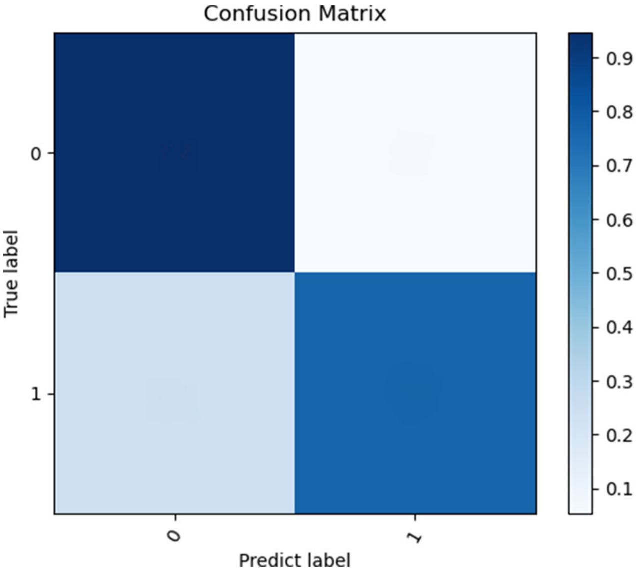


Fig. 7 The confusion matrix describes the results of Liar, which is generated by IIJIPN

Data augmentation for constraint Data augmentation is also applied in the Constraint Dataset. It is shown from Table 9 that the difference value on Acc between the IIJIPN (-Aug) and IIJIPN is only 0.0008. The improvement of data augmentation is not obvious because the data with 10,700 posts in the Constraint dataset is balanced. According to the results of Constraint dataset, it can be concluded that data augmentation can effectively solve the problems of data imbalance or the shortage of training data in the datasets.

4.6.2 Analysis of third-order text graph tensor

We also examine and perform an analysis of the effectiveness of Third-order Text Graph Tensor with sequential information, syntactic information and semantic information. We generate three variant models, each of which does not consider one text feature.

Sequential information From Fig. 8, when the sequential information is not considered in IIJIPN, the value of Acc will decrease at different degrees. Among all datasets, the impact on Twitter15 and Twitter16 are greater than Liar and Constraint. Twitter15 and Twitter16 are small corpus. The frequency of word co-occurrence in these two datasets are small. The word co-occurrence means

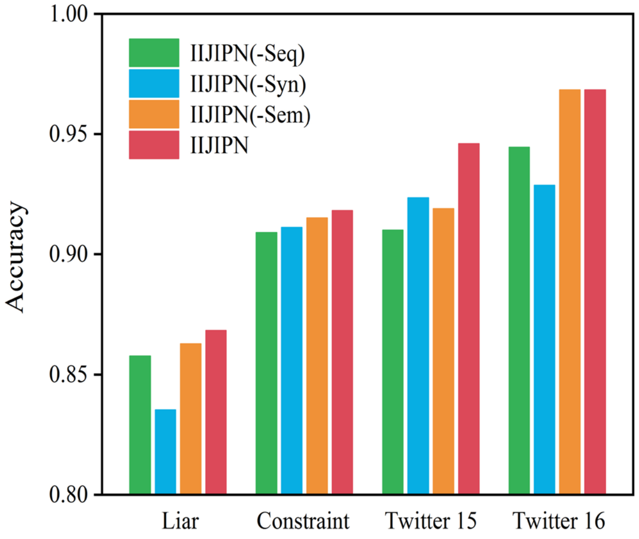


Fig. 8 Analysis of the effectiveness of sequential, syntactic, and semantic features, and three variants are produced. For example, IIJIPN(-Seq) means IIJIPN only considering feature syntactic and semantic feature rathan than sequential feature

that words have associated semantics, which can be used to capture the contexts information. Therefore, when the sequential information is removed, there is no context information can be used to learn the word representation. Liar and Constraint are large corpus, even if the sequential information is not considered, the word representation can be learned by obtaining the syntactic information and semantic information.

Syntactic information From Fig. 8, When the syntactic information is not considered, the *Acc* on Liar, Twitter15 and Twitter16 is greatly decreased, but Constraint is not. For Liar and Twitter16, the variant model without syntactic information achieves the worst result among three variant models. It can be seen that syntactic information has a great impact on IIJIPN. Syntactic information is obtained by extracting the grammar dependency between words. Both sequential and syntactic information are used to look for the relationship between words. The difference is that in the extraction of sequential information, we are extracting the sequence relationship between words and their neighbors in a limited range(a sliding window); In the extraction of syntactic information, we can find the grammar dependency between two words that are far away in the text.

Semantic information From Fig. 8, When the semantic information is not considered, the *Acc* of Twitter15 is greatly affected, other three datasets are not. When acquiring semantic information, the topic and topic-related words are extracted in the datasets. It means that each word has different relevance to each topic. When distinguishing the true news from the false news, a certain kind of news usually related to specific topics and words. Therefore, the semantic information of a sentence can be obtained by integrating the words with topic distribution.

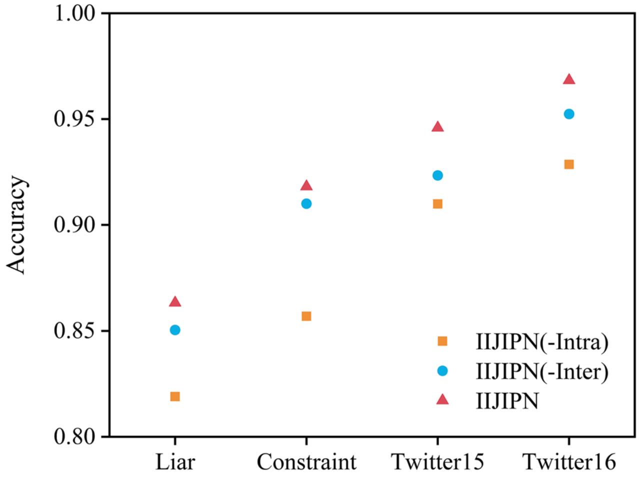


Fig. 9 Analysis of the effectiveness of information propagation methods in IIJIPN. IIJIPN(-Intra) means IIJIPN with only inter-graph propagation rather than intra-graph propagation

As shown in Fig. 8, IIJIPN is always performing better than the other three models. Only considering two kinds of text features, the performance on the fake news detection is weakened. It illustrates that all the sequential, syntactic and semantic features are complementary elements for fake news detection. Besides, different feature of text information will affect the performance for datasets at different degree. For example, the *Acc* on Liar and Twitter16 datasets are worse without considering syntactic information. It means that the syntactic dependency plays an important role in these two datasets. Similarly, not considering the sequential information will dramatically weaken the classification performance of the Constraint dataset. The *Acc* of Twitter15 dataset is worse without considering semantic information. It can be concluded that each text information will influence the classification performance respectively, and it is best when considering all features.

4.6.3 Analysis of Intra-graph and Inter-graph information propagation

Finally, we examine the effectiveness of Intra-graph and Inter-graph information propagation respectively. We develop two variants for IIJIPN, IIJIPN (-intra) and IIJIPN (-inter), which is IIJIPN but without using intra-graph information propagation and inter-graph information propagation respectively.

Intra-graph information propagation The comparison results presented in Fig. 9 illustrate that IIJIPN has the best the performance, and IIJIPN (-inter) always show better performance than IIJIPN (-intra). The worse results in IIJIPN (-intra) can be attributed to which the word can not exchange information from neighbours within a graph. The lacking of information interaction leads to difficulty transfer the text information of each word to others, resulting in discontinuous information of the entire text.

Inter-graph information propagation From Fig. 9, it can be seen that IIJIPN (-inter) achieves better performance than IIJIPN (-intra), but poorly than IIJIPN. As for the inter-graph information propagation, IIJIPN cannot interact heterogeneous information without it, which has a bad influence on updating the word representation. Both Intra-graph and Inter-graph information propagation are considered simultaneously, and the homogeneous and heterogeneous information can be sufficiently learned.

5 Conclusion

The spread of fake news on social media has become a serious social problem. In this work, we propose Intra-graph and Inter-graph Joint Information Propagation Network (abbreviated as IIJIPN) with Third-order Text Graph Tensor for fake news detection. Particularly, we firstly propose data augmentation for corpus solving data imbalance and the shortage of label data. To comprehensively describe contextual information, Third-order Text Graph Tensor with sequential, syntactic, and semantic features is proposed. After constructing graphs for each news, intra-graph and inter-graph information propagation are used for realizing homogeneous and heterogeneous information interaction respectively. The experimental results demonstrate that the method we proposed is a powerful tool to effectively detect fake news.

During the experiments, huge cost of memory and time are existed in our model, so we will pay attention to how to deal with this problem in future. Moreover, additional information about news such as user profile, comments, or propagation paths will be merged in our model.

Acknowledgements This work was supported by the National Natural Science Foundation of China (61772231), the Shandong Provincial Natural Science Foundation (ZR2022LZH016 & ZR2017MF025), the Project of Shandong Provincial Social Science Program (18CHLJ39), and the Shandong Provincial Key R&D Program of China (2021CXGC010103).

Author Contributions All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Benkuan Cui, Kun Ma, Leping Li, Weijuan Zhang, Ke Ji, Zhenxiang Chen, and Ajith Abraham. Kun Ma contributed equally to this work and should be considered co-first authors. The first draft of the manuscript was written by Benkuan Cui and Kun Ma, and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Code Availability The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of Interests The authors declare that they have no conflict of interest.

References

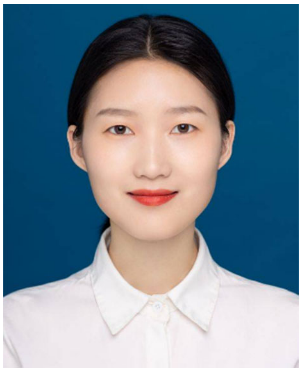
1. Shu K, Sliva A, Wang S, Tang J, Liu H (2017) Fake news detection on social media: a data mining perspective. ACM SIGKDD Explor Newsl 19(1):22–36
2. Song C, Shu K, Wu B (2021) Temporally evolving graph neural network for fake news detection. Inf Process Manag 58(6):102712
3. Allcott H, Gentzkow M (2017) Social media and fake news in the 2016 election. J Econ Perspect 31(2):211–36
4. Vosoughi S, Roy D, Aral S (2018) The spread of true and false news online. Science 359(6380):1146–1151
5. Patwa P, Sharma S, Pykl S, Guptha V, Kumari G, Akhtar MS, Ekbal A, Das A, Chakraborty T (2021) Fighting an infodemic: Covid-19 fake news dataset. In: International workshop on combating online hostile posts in regional languages during emergency situation, Springer, pp 21–29
6. Xiong J, Lipsitz O, Nasri F, Lui LM, Gill H, Phan L, Chen-Li D, Iacobucci M, Ho R, Majeed A et al (2020) Impact of covid-19 pandemic on mental health in the general population: a systematic review. Journal of affective disorders
7. Hu G, Ding Y, Qi S, Wang X, Liao Q (2019) Multi-depth graph convolutional networks for fake news detection. In: CCF international conference on natural language processing and chinese computing, Springer, pp 698–710
8. Han Y, Karunasekera S, Leckie C (2020)
9. Goldani MH, Momtazi S, Safabakhsh R (2021) Detecting fake news with capsule neural networks. Appl Soft Comput 101:106991
10. Chen Y, Zhang H, Liu R, Ye Z, Lin J (2019) Experimental explorations on short text topic mining between lda and nmf based schemes. Knowl-Based Syst 163:1–13
11. Tang J, Qu M, Mei Q (2015) Pte: Predictive text embedding through large-scale heterogeneous text networks. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, pp 1165–1174
12. Wang J, Wang Z, Zhang D, Yan J (2017) Combining knowledge with deep convolutional neural networks for short text classification. In: IJCAI, vol 350
13. Wang Y, Qian S, Hu J, Fang Q, Xu C (2020) Fake news detection via knowledge-driven multimodal graph convolutional networks. In: Proceedings of the 2020 international conference on multimedia retrieval, pp 540–547
14. Linmei H, Yang T, Shi C, Ji H, Li X (2019) Heterogeneous graph attention networks for semi-supervised short text classification. In: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP), pp 4821–4830
15. Ye Z, Jiang G, Liu Y, Li Z, Yuan J (2020) Document and word representations generated by graph convolutional network and bert for short text classification. In: ECAI 2020, pp 2275–2281

16. Wang Y, Wang S, Yao Q, Dou D (2021) Hierarchical heterogeneous graph representation learning for short text classification. In: Proceedings of the 2021 conference on empirical methods in natural language processing, pp 3091–3101
17. Yao L, Mao C, Luo Y (2019) Graph convolutional networks for text classification. In: Proceedings of the AAAI conference on artificial intelligence, vol 33. pp 7370–7377
18. Zhang Y, Yu X, Cui Z, Wu S, Wen Z, Wang L (2020) Every document owns its structure: Inductive text classification via graph neural networks. In: Proceedings of the 58th annual meeting of the association for computational linguistics, pp 334–339
19. Liu X, You X, Zhang X, Wu J, Lv P (2020) Tensor graph convolutional networks for text classification. In: Proceedings of the AAAI conference on artificial intelligence, vol 34. pp 8409–8416
20. Benamira A, Devillers B, Lesot E, Ray AK, Saadi M, Malliaros FD (2019) Semi-supervised learning and graph neural networks for fake news detection. In: 2019 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM), IEEE, pp 568–569
21. Li Q, Peng H, Li J, Xia C, Yang R, Sun L, Philip SY, He L (2020) A survey on text classification: from shallow to deep learning
22. Heap B, Bain M, Wobcke W, Krzywicki A, Schmeidl S (2017) Word vector enrichment of low frequency words in the bag-of-words model for short text multi-class classification problems. arXiv:1709.05778
23. Samant SS, Murthy NB, Malapati A (2019) Improving term weighting schemes for short text classification in vector space model. IEEE Access 7:166578–166592
24. Chen Y (2015) Convolutional neural network for sentence classification. University of Waterloo, Master's thesis
25. Xie J, Chen B, Gu X, Liang F, Xu X (2019) Self-attention-based bilstm model for short text fine-grained sentiment classification. IEEE Access 7:180558–180570
26. Liu G, Guo J (2019) Bidirectional lstm with attention mechanism and convolutional layer for text classification. Neurocomputing 337:325–338
27. Ding K, Wang J, Li J, Li D, Liu H (2020) Be more with less: Hypergraph attention networks for inductive text classification. In: Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP), pp 4927–4936
28. Kaliyar RK, Goswami A, Narang P, Sinha S (2020) Fndnet–a deep convolutional neural network for fake news detection. Cogn Syst Res 61:32–44
29. Jadhav SS, Thepade SD (2019) Fake news identification and classification using dssm and improved recurrent neural network classifier. Appl Artif Intell 33(12):1058–1068
30. Devlin J, Chang M-W, Lee K, Toutanova K (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805
31. Kaliyar RK, Goswami A, Narang P (2021) Fakebert: Fake news detection in social media with a bert-based deep learning approach. Multimed Tools Appl 80(8):11765–11788
32. Vlad G-A, Tanase M-A, Onose C, Cercel D-C (2019) Sentence-level propaganda detection in news articles with transfer learning and bert-bilstm-capsule model. In: Proceedings of the second workshop on natural language processing for internet freedom: censorship, disinformation, and propaganda, pp 148–154
33. Lan Z, Chen M, Goodman S, Gimpel K, Sharma P, Soricut R (2020) ALBERT: A lite BERT for self-supervised learning of language representations. In: 8Th international conference on learning representations, ICLR 2020, Addis ababa, Ethiopia, April 26-30
34. Gundapu S, Mamidi R (2021) Transformer based automatic covid-19 fake news detection system. arXiv:2101.00180
35. Han Y, Huang G, Song S, Yang L, Wang H, Wang Y (2021) Dynamic neural networks: a survey. IEEE Transactions on Pattern Analysis and Machine Intelligence
36. Shu K, Cui L, Wang S, Lee D, Liu H (2019) Defend: explainable fake news detection. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, pp 395–405
37. Thabtah F, Hammoud S, Kamalov F, Gonsalves A (2020) Data imbalance in classification: experimental evaluation. Inf Sci 513:429–441
38. Wei J, Zou K (2019) Eda: easy data augmentation techniques for boosting performance on text classification tasks. In: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP), pp 6382–6388
39. Kilmer ME, Braman K, Hao N, Hoover RC (2013) Third-order tensors as operators on matrices: a theoretical and computational framework with applications in imaging. SIAM J Matrix Anal Appl 34(1):148–172
40. Manning CD, Surdeanu M, Bauer J, Finkel JR, Bethard S, McClosky D (2014) The stanford corenlp natural language processing toolkit. In: Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations, pp 55–60
41. Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. J Mach Learn Res 3:993–1022
42. Li Y, Zemel R, Brockschmidt M, Tarlow D (2016) Gated graph sequence neural networks. In: Proceedings of ICLR'16
43. 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 (vol 2. Short Papers), pp 422–426
44. Liu X, Nourbakhsh A, Li Q, Fang R, Shah S (2015) Real-time rumor debunking on twitter. In: Proceedings of the 24th ACM international on conference on information and knowledge management, pp 1867–1870
45. Ma J, Gao W, Mitra P, Kwon S, Jansen BJ, Wong K-F, Cha M (2016) Detecting rumors from microblogs with recurrent neural networks
46. Zhuang L, Wayne L, Ya S, Jun Z (2021) A robustly optimized bert pre-training approach with post-training. In: Proceedings of the 20th Chinese national conference on computational linguistics, pp 1218–1227
47. Zeng J, Li J, Song Y, Gao C, Lyu MR, King I (2018) Topic memory networks for short text classification. In: Proceedings of the 2018 conference on empirical methods in natural language processing, pp 3120–3131

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

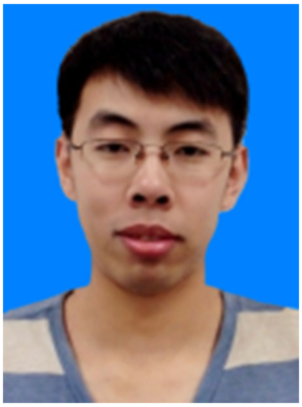
Benkuan Cui is a postgraduate in University of Jinan, China from 2020. His research interests include natural language processing and distributed computing.

Kun Ma received his Ph.D degree in Computer Software and Theory from Shandong University, Jinan, Shandong, China, in 2011. He is an Associate Professor in School of Information Science and Engineering, University of Jinan, China. He presides 1 National Natural Science Foundation of China (NSFC), and 6 provincial scientific research projects and 6 teaching research projects. He has authored and coauthored over 100 research publications in peer-reviewed reputed journals and conference proceedings. His entire publications have been cited over 760 times (Google Scholar). 9 of his papers have been cited over 20 times. The latest Google h-index of his publications is 15. He has served as the program committee member of various international conferences and reviewer for various international journals. He is the Co-Editor-in-Chief of International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM). He is the managing editor of Journal of Information Assurance and Security (JIAS). He is the editorial board member of Engineering Applications of Artificial Intelligence and International Journal of Grid and Utility Computing. His research interests include stream data processing, data intensive computing, natural language processing, and big data management. He has obtained 23 patents for inventions.



Leping Li is a postgraduate in University of Jinan, China from 2020. Her research direction is natural language processing.

Weijuan Zhang received the Master's degree in computer application technology from University of Jinan in 2009. She is an associate professor at Shandong College of Electronic Technology. Her research interest is natural language processing.



Ke Ji was born in 1989. He received the Ph.D. degree in computer science and technology from Beijing Jiaotong University in 2016. He is an associate professor at University of Jinan. His research interests include machine learning, recommendation system.

Zhenxiang Chen has been a professor of School of Information Science and Engineering at University of Jinan, China since 2015. He received his PhD degree in from Shandong University in 2007. From 2015 to 2016, He had been a postdoctoral fellow in the Department of Electrical and Computer Engineering at Virginia Polytechnic Institute and State University, USA. He research interest is data intensive computing.

Ajith Abraham (Senior Member, IEEE) received the Master of Science degree from Nanyang Technological University, Singapore, in 1998, and the Ph.D. degree in computer science from Monash University, Melbourne, Australia, in 2001. He is currently the Director of the Machine Intelligence Research Laboratories (MIR Laboratories), a Not-for-Profit Scientific Network for Innovation and Research Excellence Connecting Industry and Academia. The Network with HQ in Seattle, USA, is currently more than 1,500 scientific members from over 105 countries. As an Investigator/a Co-Investigator, he has won research grants worth over more than 100 million U.S. Dollar. He currently holds two university professorial appointments. He works as a professor in artificial intelligence with Innopolis University, Russia, and the Yayasan Tun Ismail Mohamed Ali Professorial Chair of Artificial Intelligence with UCSI, Malaysia. He works in a multi-disciplinary environment and he has authored/co-authored more than 1,400 research publications out of which there are more than 100 books covering various aspects of computer science. One of his books was translated to Japanese and a few other articles were translated to Russian and Chinese. He has more than 46,000 academic citations (Hindex of more than 102 as Per Google Scholar). He has given more than 150 plenary lectures and conference tutorials (in more than 20 countries). He was the Chair of IEEE Systems Man and Cybernetics Society Technical Committee on Soft Computing (which has over more than 200 members), from 2008 to 2021, and served as a Distinguished Lecturer for IEEE Computer Society representing Europe, from 2011 to 2013. He was the Editor-in-Chief of Engineering Applications of Artificial Intelligence (EAAI), from 2016 to 2021 and serves/served the editorial board of over 15 international journals indexed by Thomson ISI.

Affiliations

Benkuan Cui[1] · Kun Ma[1,2] · Leping Li[1,2] · Weijuan Zhang[3] · Ke Ji[1,2] · Zhenxiang Chen[1,2] · Ajith Abraham[4]

Benkuan Cui
cuibenkuan@stu.ujn.edu.cn

Leping Li
202021100417@mail.ujn.edu.cn

Weijuan Zhang
zhangweijuan@sdcet.cn

Ke Ji
ise_jik@ujn.edu.cn

Zhenxiang Chen
czx@ujn.edu.cn

Ajith Abraham
ajith.abraham@ieee.org

[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] Department of Computer and Software Engineering,
Shandong College of Electronic Technology,
Jinan, 250200, China

[4] Machine Intelligence Research Labs, Scientific Network
for Innovation and Research Excellence,
Auburn, 98071-2259, USA