DTN: Deep Triple Network for Topic Specific Fake News Detection

Jinshuo Liu[a], Chenyang Wang[a], Chenxi Li[a], Ningxi Li[a], Juan Deng[a], Jeff Z. Pan[b,c,*]

[a] *Wuhan University, China*
[b] *University of Aberdeen, UK*
[c] *University of Edinburgh, UK*

Abstract

Detection of fake news has spurred widespread interests in areas such as healthcare and Internet societies, in order to prevent propagating misleading information for commercial and political purposes. However, efforts to study a general framework for exploiting knowledge, for judging the trustworthiness of given news based on their content, have been limited. Indeed, the existing works rarely consider incorporating knowledge graphs (KGs), which could provide rich structured knowledge for better language understanding.

In this work, we propose a deep triple network (DTN) that leverages knowledge graphs to facilitate fake news detection with triple-enhanced explanations. In the DTN, background knowledge graphs, such as open knowledge graphs and extracted graphs from news bases, are applied for both low-level and high-level feature extraction to classify the input news article and provide explanations for the classification.

The performance of the proposed method is evaluated by demonstrating abundant convincing comparative experiments. Obtained results show that DTN outperforms conventional fake news detection methods from different aspects, including the provision of factual evidence supporting the decision of fake news detection.

Keywords: knowledge graph, knowledge graph embedding, multi-channel, deep learning, fake news detection

1. Introduction

Motivation: Fake news refers to news that is intentionally and veritably false so that could mislead readers [1]. With the expansion of social networks in recent years, fake news has become one of the most hotly-debated socio-political topics and has received considerable attention. This is especially more pronounced in the wake of the *2016 US Election* and *Brexit*. Considering the intrinsic characteristics of social media, including low cost, easy access, and rapid dissemination, they can easily mislead public opinion, disturb the social order, and even damage the credibility of the media itself. Therefore, it is of significant importance to investigate detection methods for fake news.

In order to detect fake news, a variety of methods have been developed so far. Some of these methods take the detection of fake news as a normal classification problem [2, 3, 4], while others consider it as a fact-checking problem [5, 6, 7]. In the former methods, the main purpose is to prepare an effective evaluation for each news. However, the importance of content might be ignored occasionally so that a good explanation may not be prepared. On the other hand, triples or claims are applied in the latter category as the input. However, it is almost impossible to present all fake and true news in predefined forms.

Reviewing the literature indicates that many investigations have been performed so far on fake news detection and specific topics. Many of them rely on unique features of specific topics such as political, health and disaster. Consequently, these methods are less applicable in other topics [8, 9, 10, 11]. Unlike these works, in the present study, it is intended to use external topic-related knowledge, such as those in knowledge graphs [12, 13], to detect fake news.

Example:

"Hillary Clinton and her State department were actively arming Islamic jihadists, which includes ISIS..."

This is an example sentence of a fake news that was issued in the midst of the 2016 US Election, from which one could extract a triple (*Hillary Clinton*, *actively arm*, *Islamic jihadists*). The main purpose of the present article is to judge such extracted triples whether they are true or fake. A challenging key in this regard is that the existing open knowledge graphs mainly focus on the attributes such as (*Hillary Clinton*, *spouse*, *Bill Clinton*) and (*London*, *is the capital of*, *UK*) rather than other events. Accordingly, they cannot provide an appropriate support for fake news detection. In order to resolve this shortcoming, one idea is to collect correct triples from related true

*Corresponding author
Email addresses: `dengjuan@whu.edu.cn` (Juan Deng), `j.z.pan@ed.ac.uk` (Jeff Z. Pan)
URL: `http://knowledge-representation.org/j.z.pan/` (Jeff Z. Pan)

news. For instance, the following sentence can be mentioned in this regard:

"Hillary Clinton-led State Department had approved weapon shipments to Libya during the intervention in 2011, and that those weapons had later ended up in the hands of jihadists..."

In this short article, several triples could be extracted. More specifically, (*Hillary State Department*, *had approved*, *weapon shipments*), (*weapon shipments*, *is to*, *Libya*) and (*weapons*, *had ended up in*, *hands of jihadists*) can be extracted in this regard. Although these triples can provide more useful information, the quality of these extracted triples from news might not be as good as the triples from open knowledge graphs. On the other hand, although extracted knowledge graphs could provide better coverage and freshness than available open knowledge graphs such as DBpedia, they may be incomplete and imprecise, which cannot be justified.

Problem: A wide variety of methods have been developed so far for text classification. However, most of them only focus on semantic features [2, 3], care more about topics [14, 15, 16] and mainly rely on training data for machine learning [4]. Different from text classification problems, fake news detection has the following characteristics that make it a challenging problem:

- **In a given fake news article, features cannot separate the fake part.** Although many fake news may share some features, including the short length and negative tone, the authenticity of contents should be considered as the basis to judge, and explain why it is true or fake.

- **Only entities can not judge the news.** In most cases, entities alone are not enough to confirm whether a sample news is true or fake. Indeed, a fake news article can be created from a number of true news by simply replacing a correct correlation with the wrong one. Consequently, it is essential to consider the whole triples rather than entities alone.

- **Article bases might not be enough to cover sufficient knowledge.** Although article bases can be useful in some cases, they might not cover some knowledge shared in a certain domain. In this case, it is necessary to take additional background knowledge graphs such as DBpedia [1] and Wikidata [2] into consideration for detecting fake news.

- **It is of great importance and value to have a reusable methodology for detecting fake news on certain topics.** Most of the existing methods for detecting fake news either do not consider the topic in the detection, or rely on some specific features in certain topics. This issue will be investigated in detail in the following sections. Meanwhile, a new method is proposed in the present study, which is capable of detecting different topics. To this end, the proposed method does not rely on specific features but to leverage general knowledge graphs such as DBpedia and Wikidata.

In order to resolve the above mentioned shortcomings, it is of significant importance to construct a novel framework for integrating knowledge graphs into machine learning frameworks for detecting fake news in specific topics.

Approach: In the present study, it is intended to propose a novel framework that takes triples into the consideration. Moreover, it takes the advantage of background knowledge, namely the deep triple network (DTN), to combine the knowledge graph and deep learning network for detecting fake news. For a given input article, DTN provides the output classification on whether the input article is fake or true. Meanwhile, it provides additional factual evidence supporting the classification. Since the training data sets might not provide sufficient knowledge, embedding models should be pre-trained through the knowledge graph extracted from open knowledge graphs on the same topic.

DTN provides a triple extraction model that transforms an unstructured text into structured triples. These triples are extracted from the input article and articles in the news base to enrich background knowledge from open knowledge graphs. Then a background knowledge-level representation is calculated from the knowledge embedding model and a semantic-level representation through a word embedding model. In order words, DTN is able to produce representations of the same triple in different channels. In one channel, a LSTM model is designed to extract low-level features. In other channels, a transformer encoding structure is set to extract low-level features and take the triples as input. Moreover, an attention model is embedded in the LSTM model to focus on important entities in the triple. Accordingly, a CNN model is chosen to combine low-level features of the two channels to extract high-level features and give classification results. Based on this framework, about 5% improvement is achieved compared with machine learning baselines. Moreover, almost the same performance is achieved compared with other deep learning baselines, with additional factual evidence supporting fake news detection.

Contribution: Main contributions of this article can be summarized as follows:

- Knowledge graphs are introduced to conduct fake news detection while considering both entity and relation information to overcome the limitations originating from lacking extra knowledge for plain texts.

- Semantic-level and background knowledge-level embedding are compromised via the triple-based multi-channel encoding model, which largely enhances the representation with equilibrium.

[1]`https://wiki.dbpedia.org/`
[2]`https://www.wikidata.org/`

- A comprehensive explanation is provided on the proposed fake news classifier by pointing out the interpretable contents in articles with extracted triples supporting the decision.

- The proposed DTN is the first network for detecting fake news on a specific topic, which can also be applied for different topics. Furthermore, experiments show that DTN outperforms all existing triple-based approaches for fake news detection.

2. Preliminaries

2.1. Knowledge graph and knowledge graph embedding

The research of Knowledge Graph is rooted in early KR work on Semantic Networks, the limitations of which have been addressed by many KR researchers. These efforts lead to the set up of the international knowledge graph standard RDF[3], as well as the OWL[4] standard for Knowledge Graph schemas and the standard Knowledge Graph querying language SPARQL[5]. The use of knowledge graph [12, 13] has become popular in knowledge representation and knowledge management applications widely across search [17, 18, 19], question answering [20, 21, 22, 23], dialogues [24], recommendation [25, 26], medical informatics [27, 28], finance [29], science [30, 31, 32, 33], media [34], software engineering [35, 36, 37, 38] environmental science [39] and industrial domains [12, 40]. Knowledge Graph is an important sub-field of Artificial Intelligence, helping to reduce the need of large, labelled datasets in Machine Learning and Deep Learning. For example, knowledge graphs have been shown to be effective in transfer learning [41, 42] and zero-shot learning [43, 44].

A knowledge graph $G = T \cup A$ consists of a data sub-graph A (or ABox) and a schema sub-graph T (or TBox). The size of sub-graph T is often much smaller than that of A. It should be indicated that the TBox T includes type axioms and relation axioms defined in the W3C OWL standard ontology language. Facts in the data sub-graph A are represented as triples in the following two forms:

- *Relation assertion* (h, r, t) where h, r and t denote the head-, relation- and tail-entity, respectively. For example, the triple (*Ivanka_Trump*, *child of*, *Donald_Trump*) is a relation assertion, where "*Ivanka_Trump*" is the head, "*child_of*" is the relation and "*Donald_Trump*" is the tail.

- *Type assertion* $(e, rdf: type, C)$ where e is an entity, $rdf: type$ is the instance-of relation from the standard W3C RDF specification and C is a type. The triple (*Donald_Trump, rdf:type, President*) is an example of type assertion.

[3]https://www.w3.org/TR/2014/REC-rdf11-mt-20140225/

[4]https://www.w3.org/TR/owl2-overview/

[5]https://www.w3.org/TR/2013/REC-sparql11-overview-20130321/

The main purpose of the knowledge graph completion (KGC) [45] is to produce an ABox extension of Ap so that the triples use only entities from A and types/relations from T. One of the most important tasks of KGC is to predict the missing head h or the missing tail t of a triple. In the present study, the knowledge graph embedding (KGE) approach is employed to embed entities and relations into a low-dimensional continuous vector space and simplify operations on the KG. The main idea behind the embedding is to represent an entity as a k-dimensional vector $\mathbf{h}$ (or $\mathbf{t}$) and define a scoring function $f_r(h, t)$ to measure the plausibility of the triple (h, r, t) in the embedding space. Representations of entities and relations are obtained by minimizing a global loss function involving all entities and relations. It is worth noting that different KGE algorithms often differ in their scoring functions, transformations, and loss functions. There are also efforts on combining KGC solutions to come up with some ensemble solutions, such as [46, 47].

2.1.1. TransE model

TransE model [48] represents both entities and relations as vectors in the same space. In this regard, the relation $\mathbf{r}$ is presented as a translation vector, which connects the entity vectors $\mathbf{h}$ and $\mathbf{t}$ with error as low as possible. In other words, if the triple (h, r, t) holds, then $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$. Accordingly, the scoring function in the TransE model can be expressed in the form below:

$$f_r(h, t) = ||\mathbf{h} + \mathbf{r} - \mathbf{t}||_{\ell_{1/2}} \tag{1}$$

When (h, r, t) holds, the output of equation (Eq-1) is a small value; otherwise, it results in relatively big values. Since the TransE model learns only one low-dimensional vector for each entity and relation, it relies on a reduced set of parameters. Moreover, although it works well with 1-to-1 relations, it has issues for N-to-1, 1-to-N and N-to-N relations.

2.1.2. TransH model

In order to resolve the shortcomings of the TransE model for processing 1-to-N, N-to-1 and N-to-N relations, TransH model [49] was proposed. In this model, each entity can have distributed representations when involved in different relations. For a given relation r, the relation-specific translation vector $\mathbf{d}_r$ is placed in the relation-specific hyperplane $\mathbf{w}_r$. Moreover, for a triple (h, r, t), the embedding $\mathbf{h}$ and $\mathbf{t}$ are projected to $\mathbf{w}_r$ so that they are presented as $\mathbf{h}_\perp$ and $\mathbf{t}_\perp$, respectively. The scoring function of the TransH model can be defined as:

$$f_r(h, t) = ||\mathbf{h}_\perp + \mathbf{r} - \mathbf{t}_\perp||_2^2 \tag{2}$$

2.1.3. TransR model

Both previously discussed TransE and TransH models project entities and relations to the same vector space. However, relations and entities are different objects so that they should not be projected to the same space. In order to resolve this problem, TransR model [50] was proposed to set a mapping matrix $\mathbf{M}_r$ for each relation r and map entity embedding vectors into the relation vector space. The scoring function for TransR model is defined as the following:

$$f_r(h,t) = ||\mathbf{h}\mathbf{M}_r + \mathbf{r} - \mathbf{t}\mathbf{M}_r||_2^2 \tag{3}$$

where $\mathbf{M}_r \in \mathbb{R}^{m\times n}$, $\mathbf{h}, \mathbf{t} \in \mathbb{R}^n$ and $\mathbf{r} \in \mathbb{R}^m$.

2.1.4. TransD model

In the TransD model [51], some simplifying assumptions are applied to the TransR model to decompose the mapping matrix into two mapping matrices, $\mathbf{M}_r^1$ and $\mathbf{M}_r^2$. In this model, entity embedding vectors are mapped into the relation vector space through two matrices. Accordingly, the scoring function is defined as:

$$f_r(h,t) = ||\mathbf{h}\mathbf{M}_r^1 + \mathbf{r} - \mathbf{t}\mathbf{M}_r^2||_2^2 \tag{4}$$

Further investigations showed that the TransD model is more efficient and has lower time complexity when the comparison is made with the TransR model.

2.2. Deep learning networks

2.2.1. CNN

Convolutional neural network (CNN) [52, 53, 54] is a deep feed-forward artificial neural networks, which was proposed for image processing. Suppose that $\mathbf{x}$ is a $N \times K$ vector and $\mathbf{x_i} \in \mathcal{R}^K$ is the ith K-dimensional input vector. Convolutional layers apply a convolution operation to the input $\mathbf{x}$, passing the result $\mathbf{a}$ to the next layer,

$$a_j = f(W_j^T \times x_{j:j+h-1} + b) \tag{5}$$

Where $b \in \mathcal{R}$ is a bias term, h is the size of the filter and f is a non-linear function such as hyperbolic tangent function. Pooling layers usually use the maximum value from each cluster of neurons at the prior layer, which can filter the zero-padding.

$$\hat{a} = max\{\mathbf{a}\} \tag{6}$$

Then all univariate vectors are concatenated and a single feature vector is formed. Finally, a fully connected layer is obtained with dropout and softmax output.

2.2.2. RNN

Recurrent neural network (RNN) [55, 56] is an artificial recursive neural network, in which connections between nodes form a directed graph along a sequence. Unlike other neural networks, RNN stores the neuron state of the previous step that affects the next successive layer. Therefore, the time series problem is transformed into the feed-forward problem so that it is usually used in the NLP task. Fig. 1 shows the basic structure, which is unfolded into a full network [57]: In Fig. 1, x_t is the input at step t, which can be a one-hot vector of the ith word in a sentence. Moreover, S_t is the hidden state at step t, which can be defined in the form below:

$$S_t = f(U_{x_t} + W_{s_{t-1}}) \tag{7}$$

S_{-1} is always initialized to all zeros. Furthermore, $f(x)$ and O_t denote an activation function and the output at step t, respectively. It should be indicated that RNN shares the same parameters across all steps.

2.2.3. Attention network

Attention network in neural networks, also called attention post or neural attention, is an effective scheme in text classifier tasks [58]. Studies show that not all words contribute equally to the representation of the sentence meaning in text classifier tasks. Therefore, a neural attention mechanism is necessary to quantify the importance of each word separately. Moreover, an attention network equips a neural network with the ability to focus on a subset of its inputs (or features). With a given input vector as $x \in \mathbb{R}^d$ and attention weight vector as $a \in [0,1]^k$, the corresponding attention network with parameters ϕ can be defined in the form below:

$$a = f_\phi(x) \tag{8}$$

3. Related work

3.1. FND methods based on the machine learning

The main idea of basic machine learning methods on fake news detection is trying to find some linguistic common features rather than evidence and applying different types of machine learning algorithms to do the classification. Mykhailo Granik et al. [2] found some similarities between fake news and spam emails. Both of them normally contain a lot of grammatical mistakes, affect the reader's opinion on certain topics in a manipulative way, and use a similar limited set of words in this regard. Considering these similarities, researchers applied a naive Bayes classifier as a simple approach to detect fake news [2]. Gilda [3] applied term frequency-inverse document frequency (TF-IDF) of bi-grams and probabilistic context-free grammar (PCFG) detection methods, and tested multiple classification algorithms on the dataset. It is worth noting that machine learning algorithms often use classifiers such as Naive Bayes (NB) and support vector machine (SVM). However, these methods have the problem of data sparsity.

3.2. FND methods based on the deep learning

Reviewing the literature indicates that different feature extraction-based methods have been proposed so far through deep learning. In this section, some of the most common methods in this regard are introduced:
DeClarE [59]: In order to debunk fake news and false claims through evidence-aware deep learning, the DeClarE

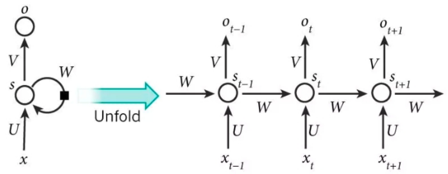

Figure 1: A recurrent neural network and unfolding the forward computation.

method was proposed as a neural network model that judiciously aggregates signals from external evidence articles, and evaluates the trustworthiness of sources. However, the DeClarE method mainly relies on certain claims, while a topic normally covers more contents rather than claims.

Many other methods apply deep learning to classify the text and detect fake news. For example, in the FastText method [60], a normalized bag of features and the Softmax function are applied to compute probabilities and explore a simple and efficient baseline for text classification. Moreover, in the TextCNN [54], a CNN is utilized to the top of the word2vec algorithm to classify sentences. In the RCNN [61], a bi-directional recurrent structure is applied to capture contextual information as far as possible for learning word representations. Furthermore, a max-pooling layer is employed that automatically judges important features to capture the key component in the text. However, the text content does not matter in these methods. Therefore, it is an enormous challenge to do an explanation through these methods.

3.3. Knowledge-based FND methods

The most straightforward way to detect fake news is to check the authenticity of the statements in the news content. Knowledge-based approaches are also known as fact-checking. Generally, these approaches can be categorized as expert-oriented and computational-oriented approaches.

The expert-oriented approaches such as Snopes[6], mainly rely on human experts working in specific fields to help the decision-making process. On the other hand, in the crowdsourcing-oriented approaches such as Fiskkit[7], where normal people can annotate the accuracy of the news content, the crowd wisdom is utilized to check the accuracy of the news articles. However, this approach is time-consuming and labor-intensive.

The computational-oriented approaches can automatically check whether the given claims have reachable paths or can be inferred from existing knowledge graphs. Ciampaglia et al. [6] considered the fact-checking as a problem of finding the shortest paths between concepts in a knowledge graph and proposed an effective indicator to assess the validity of the statement by analyzing path lengths between concepts in question. Shiralkar et al. [7] proposed a novel method called "knowledge stream" and a fact-checking algorithm called "relational knowledge linker" to verify a claim based on the single shortest, semantically related path in KG. Moreover, Shi et al. [5] transformed the fake news detection task into a link prediction task and then presented a discriminative path-based method that incorporates connectivity, type information, and predicate interactions. However, when the whole news is used as the input, none of the abovementioned approaches can be used.

3.4. Combining KGs with the deep network

The idea of combining knowledge graphs with deep neural networks for detecting fake news has not ever been proposed before. But some similar ideas such as infusing external knowledge for natural language processing have been proposed in the past two years.

For instance, the KBLSTM neural model [14] leverages embedding of concepts in knowledge bases to improve the learning of recurrent neural networks for machine-reading. It adopts an attention mechanism to decide whether to use knowledge or not and determine what information in knowledge bases is useful. However, the KBLSTM only refers to the synsets from WordNet and concept categories from NELL as the knowledge base concepts, while all other relations between different concepts (entities) are ignored.

Wang et al. [15] combined knowledge with different deep convolutional neural networks for classifying short texts and proposed a framework in this regard to combine explicit and implicit representations of short texts based on convolutional neural networks. The proposed scheme associates relevant concepts with short texts by leveraging explicit knowledge and generating the implicit representation. However, this scheme only generates the relevant concepts by conceptualizing a short text through a large taxonomy knowledge base, regardless of relations.

Wang et al. [16] incorporated a knowledge graph representation into the news recommendation and proposed the deep knowledge-aware network as a deep recommendation framework for predicting the click-through rate. Its knowledge-aware convolutional neural network fuses the semantic-level and knowledge-level representations of the news title. However, this framework maps both of the two-level representations into each word of the title, indicating that the framework does not care about the relations between words in the same title.

In order to improve learning models with knowledge graphs, Annervaz et al. [62] proposed the knowledge graph

[6] http://www.snopes.com/
[7] http://fiskkit.com

augmented neural networks for natural language processing. The main hypothesis in this method is that the general world knowledge is being infused into the learning model for any given natural language processing task. However, k-means clustering is used in the KG retrieval part to cluster similar entity/relation vectors, which operates on entities and relations separately.

Since none of the abovementioned approaches are proposed for fake news detection, it may be a feasible idea that there is no need to value relations between a pair of entities. However, as part of the knowledge graph, relations between entities are as important as entities so that they should be considered in the long run.

4. Problem statement

Considering a news article a with the topic p and some background knowledge graph G (which can be either obtained from open knowledge graphs or extracted from background article base B with the same topic p), the main task of the fake news detection is to classify a into the fake or true article with respect to G. In addition to detecting fake news articles, it is intended to provide some explanations of the judgment with the help of extracted triples T from the article a. Considering the triples T generated from a, embedding models M_w and M_k can be used to represent T in the form below:

$$T \overset{word2vec}{\longrightarrow} M_w \tag{9}$$

$$T \overset{KGEmbedding}{\longrightarrow} M_k \tag{10}$$

The KG embedding is based on B. The feature vector c of M_w and M_k originates from the deep neural networks. Another goal of the proposed framework is to generate some explanations based on the DTN results for all single triples. In order to enhance the generality of the proposed framework, DTN is equipped with flexible configurations, which makes the problems different from other approaches in the following aspects:

- DTN receives the whole news article as input, while it can also receive a news claim as the input.
- DTN does not need any other unique features of the specific topic so that the methodology can be reused for different topics.
- DTN can extract the core knowledge graph from any kind of the background knowledge and it is not limited to open knowledge graphs or related article bases only.
- DTN not only predicts whether the given news article is true or fake but also provides explanations by highlighting the important parts of news articles.

5. Methodology

In this section, the deep triple network is presented in detail. Firstly, the overview of DTN is given. Then, the triple generation component, the design of triple-based RNN and the final classification component are described.

5.1. DTN framework

Fig. 2 shows the DTN framework. It is observed that DTN contains three components, including 1) triple generation, 2) extracting low-level and high-level features, and 3) classification.

In the first component, triples are extracted from the news article under evaluation. Moreover, triples are generated based on the background information to construct the background knowledge triple set. The background information can be either the fake news-based, true news-based, or the open knowledge graphs. Each of them should correspond to an independent triple set. Triple extraction has the following advantages: It is an information extraction model that simulates human behavior, converts unstructured text to structured triples, replaces conventional feature extraction models, and preserves the semantic information of the text. The first component will be discussed in detail in section 5.2.

For the news under evaluation, the second component aims to represent the input triples reasonably and extract valid features. More specifically, it is focused on extracting features in two different level channels, as the following:

1. the semantic-level channel
2. the KG-level channel

The semantic-level channel aims to extract semantic level features from original information of the news content. In the semantic-level channel, word embedding is utilized to represent words of input triples. Then, the self-attention mechanism [63] is applied to obtain semantic-level features, where the self-attention model will automatically focus on important words. For each input triple, word representations are concatenated to maintain the triple structure.

Unlike the semantic-level channel, the KG-level channel aims to introduce external knowledge information to improve the representation of input triples. Moreover, different from the word embedding that captures the semantic representation of the input text, the knowledge graph embedding captures the background knowledge level representation of the triple. First, a knowledge graph embedding model based on the triple set of the background knowledge is pre-trained. Then, a matrix is generated by applying the triple set to the embedding model and the feature map is obtained through inputting the matrix to a BiLSTM model [64]. Corresponding to the self-attention mechanism in the semantic-level channel, an entity-based attention network is applied, which allows the proposed model to consider the important entity so that it represents the triples more reasonably. Furthermore, the entity-based

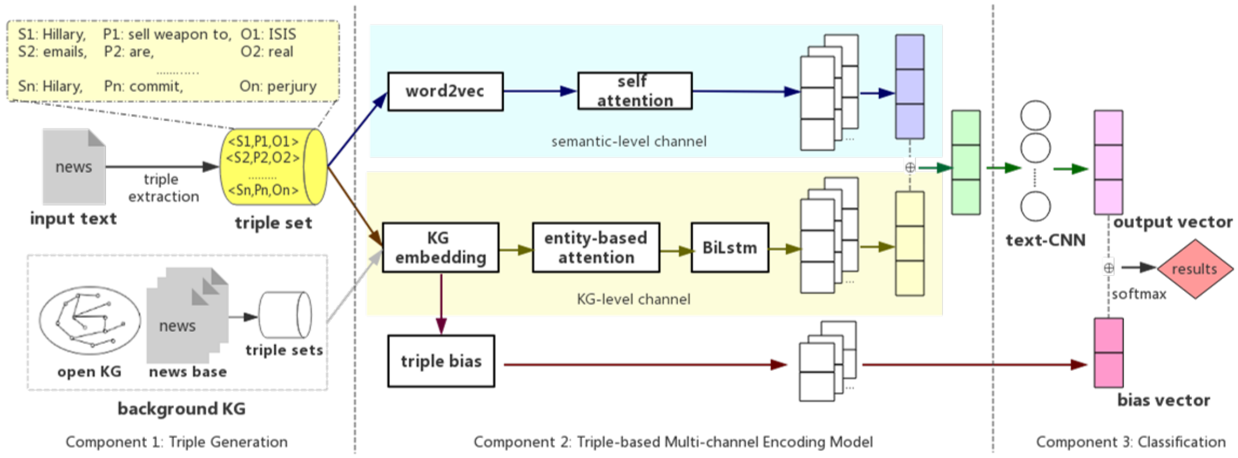

Figure 2: DTN Framework. DTN contains three components, including 1) Triple Generation, 2) Triple-based Multi-channel Encoding Model, and 3) Classification

attention network calculates the attention weight for each input triple, which supports the proposed model to give some explanation. As the triple bias represents the credibility of each triple, the bias of each triple is considered as another feature and it is used as an input to the final classifier.

The third component takes the feature maps of the semantic-level and KG-level channels as the input, and accomplishes the final decision whether the news article is true or fake. As the semantic-level features and KG-level features are in two different vector spaces, the multi-channel features fusion method is proposed to fuse the text semantic information and knowledge information. This part will be discussed in section 5.4.

5.2. Triple generation

Fig. 2 shows the triple generation in the DTN framework. It generates triple sets from the news article and the background information. Moreover, the background information can be one of the fake news article base, true news article base and open knowledge graph.

News articles are always long texts so that the triple set is considered a more concentrated and brief representation in a certain form. In this way, some unimportant descriptions in the articles are ignored and more attention is paid to what the news content wants to emphasize. This is an information extraction model, which can replace the conventional feature extraction model. Moreover, the form of triples is easier to manipulate than texts, which provides the possibility of the KG construction.

5.2.1. Generating triples from news articles

It is considered that the news correlated to the same topic can provide relevant background knowledge. Therefore, triples from the news articles of the same topic are generated and a knowledge graph is constructed. In order to generate triples from news articles, two triples extraction models are considered.

The first one uses a joint learning model for the entity extraction and correlation classification to perform the triple extraction. More specifically, a shared transformer encoding model layer is used to perform the encoding. Then, a LSTM layer is utilized to conduct the named entity recognition (NER) and a CNN model is used to perform the relation classification (RC). Compared with the current mainstream BiLSTM-CRF model, it embeds the previous prediction tag and puts it into the current decoding layer, which replaces the CRF layer to solve the tag dependency problem in NER models. When performing the correlation classification, the model pairs the entities based on the results of the NER predication and then a CNN is used to classify the correlation between the entities by using the entity pair and the text encoding results as the input. The model is mainly shared by the underlying model parameters. During training, both tasks update the shared parameters through the backward propagation algorithm to achieve the dependency between the two subtasks.

The other one starts with a set of news articles and uses openIE [65] to initially extract triples. However, openIE does not perform well enough in the triple extraction of news, which cannot meet our needs. Therefore, Stanford NER [66] is proposed to improve the quality of the triples, which can extract entities from news articles. Then, the entity alignment is done and the triples are obtained.

Finally, the two models are compared. The first model, the joint learning model with Bi-LSTM and CNN, should label the text, which is not scalable. Moreover, the joint learning model can only extract the correlation of a specific category, and cannot describe the content of the news very well. Therefore, the OpenIE and Stanford NER combined models are selected as the news triple extraction model.

5.2.2. Generating triples from an open knowledge graph

The principle of generating triples from the open KG is to discover some facts (triples) correlated to the known topic p. It is worth noting that this is not a problem of the domain-specific sub-graph extraction, but a problem of collecting enough triples on the topic p. Therefore, the simple hop-based approach is adopted to generate the sub-graph G from the open KG. Given a predefined hop-number g, the process of generating triples can be expressed as the following:

$$G = \{T_G\} \tag{11}$$

$$T_G = \{T_1, T_2, ..., T_g\} \tag{12}$$

Where T_1 and T_x denote all triples that have the formulas as $\{h_0, r, t\}$ and $\{h_x, r, t\}$ $(x \in \{2, 3, ..., g\})$, respectively. Moreover, h_0 and h_x are a head entity in T_x that refers to the topic entity p and a tail entity in T_{x-1}, respectively.

5.3. Deep triple network

Conventional neural language representation models such as transformer [63], aim to automatically learn the text representation and semantic patterns from the large-scale text. However, these models rarely consider the external knowledge. It is observed that knowledge graphs contain rich structured knowledge facts, which can help the model extract features from the text better. In this study, the deep triple network (DTN) is proposed, which considers using the knowledge graph embedding models to introduce the external knowledge, and extracts triple features from multiple different channels to improve the detection effect of fake news. Different from the conventional language representation models that only use text embedding, in this study, two embedding models are utilized to embed triples to extract features in two channels. In the semantic-level channel, the word embedding model and self-attention mechanism are used to extract the semantic-level features of input triples. In the KG-level channel, input triples are embedded through the pre-trained knowledge graph embedding model to introduce the knowledge information. Then, the entity-based attention network is applied to make the model focus more on important entities to represent the triple better. Furthermore, the triple bias is calculated as another feature, which indicates the credibility of triples.

5.3.1. Semantic-level channel encoder

As input triples contain semantic information of the original text, the semantic-level channel encoder (SCE) is proposed to extract the semantic-level feature. More specifically, the word embedding is initially used to represent each word of triples. With an input triple (s, p, o), the word sequence is denoted as:

$$\{w_1^s, ..., w_{n_s}^s, w_1^p, ..., w_{n_p}^p, w_1^o, ..., w_{n_o}^o\}, \tag{13}$$

Where w_i^s and $n_s/n_p/n_o$ denote the $i-th$ word of subject entity and the number of word of entity/relation, respectively. Moreover, the embedding result of the word sequence is denoted as the matrix:

$$[\mathbf{x}_1^s, ..., \mathbf{x}_{n_s}^s, \mathbf{x}_1^p, ..., \mathbf{x}_{n_p}^p, \mathbf{x}_1^o, ..., \mathbf{x}_{n_o}^o], \tag{14}$$

Where $\mathbf{x}_i^s, \mathbf{x}_i^p, \mathbf{x}_i^o \in \mathbb{R}^{d_{model}}$, where d_{model} is the word embedding dimension. Since the embedding layer and the encoding layer do not use any recursive structure or convolution structure, in order to utilize the order information of the input sequence, it is necessary to introduce some information to express the absolute or relative position of each word. More specifically, the absolute position is utilized and a position embedding layer is applied to represent the position information of each word in the input sequence. The absolute position embedding vector of each word is denoted as $\mathbf{pos}_i^s/\mathbf{pos}_i^p/\mathbf{pos}_i^o \in \mathbb{R}^{d_{pos}}$, where d_{pos} refers to the position embedding dimension. In order to consider the position information, the word embedding result and position embedding result are concatenated as the following:

$$\begin{aligned} \widetilde{\mathbf{x}}_i^s &= [\mathbf{x}_i^s, \mathbf{pos}_i^s], \\ \widetilde{\mathbf{x}}_i^p &= [\mathbf{x}_i^p, \mathbf{pos}_i^p], \\ \widetilde{\mathbf{x}}_i^o &= [\mathbf{x}_i^o, \mathbf{pos}_i^o], \end{aligned} \tag{15}$$

The better word embedding is obtained, where $\widetilde{\mathbf{x}}_i^s, \widetilde{\mathbf{x}}_i^p, \widetilde{\mathbf{x}}_i^o \in \mathbb{R}^{d_{model}+d_{pos}}$. Then, the self-attention mechanism is applied to obtain the semantic-level feature. The self-attention mechanism is mathematically expressed as follows:

$$Att(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}})\mathbf{V}, \tag{16}$$

Where $\{\mathbf{Q}, \mathbf{K}, \mathbf{V}\}$ is the input matrices, the concatenation of the word embedding and the position embedding, which is defined as:

$$\mathbf{Q} = \mathbf{K} = \mathbf{V} = [\widetilde{\mathbf{x}}_1^s, ..., \widetilde{\mathbf{x}}_{n_s}^s, \widetilde{\mathbf{x}}_1^p, ..., \widetilde{\mathbf{x}}_{n_p}^p, \widetilde{\mathbf{x}}_1^o, ..., \widetilde{\mathbf{x}}_{n_o}^o]. \tag{17}$$

In order to capture more abundant features and obtain better word representation, the multi-head attention mechanism is applied, which uses different, learned linear projections to linearly project the query, key and value h times to the d_k, d_k and d_v dimensions. The multi-head attention mechanism is formalized as:

$$MultiHead(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = Concat(head_1, .., head_h)\mathbf{W}^O, \tag{18}$$

Where $head_i$, $Concat(x)$ and W^O denote the result of $i-th$ self-attention module, the concatenate operation and the matrix of the trainable parameter, respectively. The equation of $head_i$ is defined as:

$$head_i = Att(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V), \tag{19}$$

Where $\mathbf{W}_i^Q, \mathbf{W}_i^K, \mathbf{W}_i^V$ refer to the different matrix of the trainable parameter, which transfer input vectors into different sub-spaces. Finally, a feed-forward network is applied to obtain the semantic-level feature, which is a simple and position-wise fully connected layer. The forward network is defined as:

$$FC(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = MultiHead(\mathbf{Q}, \mathbf{K}, \mathbf{V}) \cdot \mathbf{W}_{FC} + b_{FC}, \tag{20}$$

Where $\mathbf{W}_{FC}$, b_{FC} and $FC(\mathbf{Q},\mathbf{K},\mathbf{V})$ denote the trainable matrix, the trainable bias, and the high-level representation of the input word sequence, respectively, which is denoted in the form below:

$$[\hat{\mathbf{x}}_1^s,...,\hat{\mathbf{x}}_{n_s}^s,\hat{\mathbf{x}}_1^p,...,\hat{\mathbf{x}}_{n_p}^p,\hat{\mathbf{x}}_1^o,...,\hat{\mathbf{x}}_{n_o}^o] = FC(\mathbf{Q},\mathbf{K},\mathbf{V}). \quad (21)$$

In order to maintain the triple structure, mean-pooling operators are employed to aggregate information from these words of each entity/relation:

$$\begin{aligned}\mathbf{h}^s &= MEAN([\hat{\mathbf{x}}_1^s,...,\hat{\mathbf{x}}_{n_s}^s]),\\ \mathbf{r}^s &= MEAN([\hat{\mathbf{x}}_1^p,...,\hat{\mathbf{x}}_{n_p}^p]),\\ \mathbf{t}^s &= MEAN([\hat{\mathbf{x}}_1^o,...,\hat{\mathbf{x}}_{n_o}^o]),\end{aligned} \quad (22)$$

Where $(\mathbf{h}^s,\mathbf{r}^s,\mathbf{t}^s)$ refers the semantic-level features of input triple (s,p,o). Therefore, with a given triple (s_i,p_i,o_i), the SCE is defined as follows:

$$[\mathbf{h}_i^s,\mathbf{r}_i^s,\mathbf{t}_i^s] = SCE(s_i,p_i,o_i). \quad (23)$$

5.3.2. KG-level channel encoder

Different from the semantic-level channel encoder, which focuses on the semantic information of input triples, the KG-level channel encoder (KGCE) is proposed to extract the background knowledge feature of input triples. In order to consider the external knowledge, a pre-trained knowledge graph embedding model is applied to embed a triple as a matrix. The knowledge graph embedding model aims to embed the entity and relation into a low-dim vector space while preserving the structural and knowledge information. With a given triple $(s,p,o\}$, a knowledge graph embedding model is used $f_{KGE}(\cdot)$ to embed it as the following:

$$[\mathbf{h},\mathbf{r},\mathbf{t}] = f_{KGE}(s,p,o), \quad (24)$$

Where $\mathbf{h},\mathbf{r},\mathbf{t} \in \mathbb{R}^{d_{kg}}$ and $\mathbb{R}^{d_{kg}}$ are embedding result vectors of input triple (s,p,o) and the knowledge graph embedding dimension, respectively. Although the knowledge graph embedding model preserves the original structural information of the input triple, it is found that only the single input triple does not provide enough knowledge information to support fake news detection. In order to introduce more knowledge information to represent input triples better, the contextual information is extracted from neighbor entities for each entity. The neighbor entities are denoted as:

$$\begin{aligned}Neighbor(s) &= \{s_i|(s,p,s_i) \in \mathcal{G} or (s_i,p,s) \in \mathcal{G}\},\\ Neighbor(o) &= \{o_i|(o,p,o_i) \in \mathcal{G} or (o_i,p,o) \in \mathcal{G}\},\end{aligned} \quad (25)$$

Where $Neighbor(s), Neighbor(o)$ and $\mathcal{G}$ denote the neighbor entities of entity s and o, and the knowledge graph, respectively. Then, the embedding vectors of neighbor entities are concatenated into a matrix:

$$\begin{aligned}Neighbor(\mathbf{h}) &= [\mathbf{h}_1,\mathbf{h}_2,...,\mathbf{h}_{n_h}],\\ Neighbor(\mathbf{t}) &= [\mathbf{t}_1,\mathbf{t}_2,...,\mathbf{t}_{n_t}],\end{aligned} \quad (26)$$

Where $\mathbf{h}_i \in \mathbb{R}^{d_{kg}}$ refers to the embedding vector of the entity s_i. In order to effectively integrate the knowledge information of all neighbor entities, a mean-pooling operator is applied to calculate the neighbor embedding. Considering the neighbor entities $Neighbor(s)$ and $Neighbor(o)$, the neighbor embedding can be calculated from the following expression:

$$\begin{aligned}\overline{\mathbf{h}} &= MEAN([\mathbf{h}_1,\mathbf{h}_2,...,\mathbf{h}_{n_h}]),\\ \overline{\mathbf{t}} &= MEAN([\mathbf{t}_1,\mathbf{t}_2,...,\mathbf{t}_{n_t}]),\end{aligned} \quad (27)$$

Where $\overline{\mathbf{h}},\overline{\mathbf{t}} \in \mathbb{R}^{d_{kg}}$ refers to the neighbor embedding. Considering triple embeddings $[\mathbf{h},\mathbf{r},\mathbf{t}]$ and corresponding neighbor embedding $\overline{\mathbf{h}}$ and $\overline{\mathbf{t}}$, it is intended to apply a strategy to combine triple embedding and corresponding neighbor embedding. A straightforward strategy simply concatenates them, which is formalized as:

$$\mathbf{T} = [\mathbf{h},\mathbf{r},\mathbf{t},\overline{\mathbf{h}},\overline{\mathbf{t}}]. \quad (28)$$

This simple concatenate strategy has two shortcomings: 1) It breaks up the order structure of the input triple. 2) The concatenate strategy does not fully consider the implicit connection between the entity and the corresponding neighbor entity.

In order to overcome the above mentioned shortcomings, an embedding fusion strategy is proposed. More specifically, the entity embedding and the neighbor embedding are initially concatenated as a matrix, $[\mathbf{h},\widetilde{\mathbf{h}}]$ and $[\mathbf{t},\widetilde{\mathbf{t}}]$. Then, a convolution operator is applied to combine the entity embedding and the corresponding neighbor entity embedding. Considering the input matrix, the convolution operator is formalized as:

$$\begin{aligned}\hat{\mathbf{h}} &= \sigma(\mathbf{W} * [\mathbf{h},\widetilde{\mathbf{h}}] + \mathbf{b}),\\ \hat{\mathbf{t}} &= \sigma(\mathbf{W} * [\mathbf{t},\widetilde{\mathbf{t}}] + \mathbf{b}),\end{aligned} \quad (29)$$

Where $\mathbf{W} \in \mathbb{R}^{d_{kg}\times 2}$, $\mathbf{b}$ and $\sigma(\cdot)$ denote the filter window, the bias and the activate function, respectively. Therefore, the triple is represented as $[\hat{\mathbf{h}},\mathbf{r},\hat{\mathbf{t}}]$. Then, a BiLSTM model is applied to extract the KG-level channel feature. For Bi-LSTM, there are two LSTM models [67]: one is the forward LSTM model $\overrightarrow{LSTM}(\cdot)$ and the other is the backward LSTM model $\overleftarrow{LSTM}(\cdot)$. There is a triple representation for each LSTM model. The triple representation of the forward LSTM model is formalized as:

$$\begin{aligned}\overrightarrow{\mathbf{h}} &= \overrightarrow{LSTM}(\hat{\mathbf{h}}),\\ \overrightarrow{\mathbf{r}} &= \overrightarrow{LSTM}(\mathbf{r}),\\ \overrightarrow{\mathbf{t}} &= \overrightarrow{LSTM}(\hat{\mathbf{t}}).\end{aligned} \quad (30)$$

Similarly, the triple representation of the backward LSTM model is formalized as:

$$\begin{aligned}\overleftarrow{\mathbf{h}} &= \overleftarrow{LSTM}(\hat{\mathbf{h}}),\\ \overleftarrow{\mathbf{r}} &= \overleftarrow{LSTM}(\mathbf{r}),\\ \overleftarrow{\mathbf{t}} &= \overleftarrow{LSTM}(\hat{\mathbf{t}}).\end{aligned} \quad (31)$$

Considering the two kinds of representations of the input triple, the two representations are concatenated as the representation of the BiLSTM model, which is formalized as

$$\begin{aligned}\mathbf{h}^{kg} &= [\overrightarrow{\mathbf{h}}, \overleftarrow{\mathbf{h}}],\\ \mathbf{r}^{kg} &= [\overrightarrow{\mathbf{r}}, \overleftarrow{\mathbf{r}}],\\ \mathbf{t}^{kg} &= [\overrightarrow{\mathbf{t}}, \overleftarrow{\mathbf{t}}].\end{aligned} \tag{32}$$

In summary, considering the triple s_i, p_i, o_i as the input, the KGCE formula is defined as:

$$[\mathbf{h}_i^{kg}, \mathbf{r}_i^{kg}, \mathbf{t}_i^{kg}] = KGCE(s_i, p_i, o_i). \tag{33}$$

5.3.3. Entity-based attention network

The KG-level channel encoder treats each entity or triple equally and considers all entities or triples to be equally important. In fact, not all entities or triples are equal in the fake news detection task. More attention should be paid to more important entities or triples. Therefore, an entity-based attention network is proposed to quantify the importance of each entity, which can give greater weight to important entities in the process of fake news detection.

The input is the entity embedding set of one news, $\mathbf{E} = \{\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_N\}$. Where $\mathbf{e}_i \in \mathbb{R}^{d_{kg}}$, d_{kg} and N are the embedding result of the ith entity, the embedding dimension and the number of the entities, respectively. A shared layer is initially used to compute attention coefficients a_{ij} that indicates the importance of u_j to u_i.

$$a_{ij} = \sigma(\mathbf{W}_v \cdot [\mathbf{e}_i^T, \mathbf{e}_j^T] + b_v), \tag{34}$$

Where $\mathbf{W}_v \in \mathbb{R}^{2 \cdot d_{kg}}, b_v \in \mathbb{R}$ are the trainable matrix and bias. Then, the attention coefficients between entity $\mathbf{e}_i$ and all other entities are computed. Moreover, the sum of the attention coefficients are utilized to measure the importance of the entity $\mathbf{e}_i$. Therefore, the attention weight of entity $\mathbf{e}_i$ is calculated as:

$$a_i = \sum_{j=1}^{N} a_{ij}. \tag{35}$$

In order to easily compare coefficients across different entities, the attention weight across all input entities is normalized as the following:

$$\tilde{a}_i = \frac{exp(a_i)}{\sum_{j=1}^{N} exp(a_j)}. \tag{36}$$

Where $\tilde{a}_i \in [0, 1]$ is the attention weight of the entity $\mathbf{e}_i$. With given a triple set of one news article, $\{s_1, p_1, o_1, ..., o_T\}$. The knowledge graph embedding model is used to obtain the entity representation, $[\mathbf{h}_1^{kg}, \mathbf{r}_1^{kg}, \mathbf{t}_1^{kg}, ..., \mathbf{t}_T^{kg}]$, where $\mathbf{h}_i^{kg}, \mathbf{r}_i^{kg}, \mathbf{t}_i^{kg} \in \mathbb{R}^{d_{kg}}$. Moreover, the entity-based attention network is utilized to add the attention weight to entities. The attention weight of $\mathbf{h}_i^{kg}$ is defined as $a_{\mathbf{h}_i^{kg}}$ and $\mathbf{t}_i^{kg}$ is defined as $a_{\mathbf{t}_i^{kg}}$. Therefore, the entity representation is updated to:

$$\begin{aligned}\mathbf{h}_i^{kg} &= a_{\mathbf{h}_i^{kg}} \cdot \mathbf{h}_i^{kg},\\ \mathbf{t}_i^{kg} &= a_{\mathbf{t}_i^{kg}} \cdot \mathbf{t}_i^{kg}.\end{aligned} \tag{37}$$

5.4. Multi-channel features fusion network

Since the semantic-level channel features and the KG-level channel features are in different vector spaces, a multi-channel feature fusion network is proposed to fuse the features of two low-level channels and extract high-level features. More specifically, a CNN is applied to fuse the two channel features $[\mathbf{h}^s, \mathbf{r}^s, \mathbf{t}^s]$ and $[\mathbf{h}^{kg}, \mathbf{r}^{kg}, \mathbf{t}^{kg}]$.

A non-linear transformation function is initially used, which transfers two-channel features into the same dimension. The non-linear transformation function is defined as the following:

$$\widetilde{\mathbf{h}}^s = \sigma(\mathbf{W}^s \mathbf{h}^s + \mathbf{b}^s), \tag{38}$$

Where $\mathbf{W}^s \in \mathbb{R}^{d_s}$, $\mathbf{b}^s \in \mathbb{R}^d$ and $\widetilde{\mathbf{h}}^s$ are the trainable matrix, the trainable bias vector and the transfer representation, respectively. Moreover, the transformation function for KG-level channel features is defined as the following:

$$\widetilde{\mathbf{h}}^{kg} = \sigma(\mathbf{W}^{kg} \mathbf{h}^{kg} + \mathbf{b}^{kg}), \tag{39}$$

Where $\mathbf{W}^{kg} \in \mathbb{R}^{d_{kg}}$ and $\mathbf{b}^{kg} \in \mathbb{R}^d$. After the non-linear transformation functions, we obtain the new representation of $[\widetilde{\mathbf{h}}^s, \widetilde{\mathbf{r}}^s, \widetilde{\mathbf{t}}^s] \in \mathbb{R}^{3 \times d}$ and $[\widetilde{\mathbf{h}}^{kg}, \widetilde{\mathbf{r}}^{kg}, \widetilde{\mathbf{t}}^{kg}] \in \mathbb{R}^{3 \times d}$. Then, the two-channel embedding matrices of triple (s_i, p_i, o_i) are aligned and stacked as:

$$\mathbf{X}_i = [[\widetilde{\mathbf{h}}_i^s, \widetilde{\mathbf{h}}_i^{kg}], [\widetilde{\mathbf{r}}_i^s, \widetilde{\mathbf{r}}_i^{kg}], [\widetilde{\mathbf{t}}_i^s, \widetilde{\mathbf{t}}_i^{kg}]] \in \mathbb{R}^{3 \times d \times 2}. \tag{40}$$

Moreover, all triples are concatenated as:

$$\mathbf{X} = \mathbf{X}_1 \oplus \mathbf{X}_2 ... \oplus \mathbf{X}_N \in \mathbb{R}^{3N \times d \times 2}, \tag{41}$$

Where $\oplus$ refers the concatenate operation. Hold two-channel features, a convolution filter is applied to fuse them and extract high-level features. The convolution operation is defined as:

$$\mathbf{c}_i^l = \tanh(\mathbf{W}^l \cdot \mathbf{X}_{i:i+l-1} + b^l), \tag{42}$$

Where $\mathbf{W}^l \in \mathbb{R}^{l \times d \times 2}$, $b^l \in \mathbb{R}$ and $\tanh(\cdot)$ denote the filter kernel, a bias and the activation functions, respectively. Then, the max pooling operation is used to obtain the most important value in $\mathbf{c}_i^l$. The max pooling operation is formalized as:

$$\hat{\mathbf{c}}^l = max\{\mathbf{c}_1^l, \mathbf{c}_2^l, ..., \mathbf{c}_N^l\}. \tag{43}$$

Inspired by TextCNN [54], multi filters with different lengths of two extract features are utilized. Moreover, all results of each filter are concatenated as:

$$\mathbf{C} = [\hat{\mathbf{c}}^{l_1}, \hat{\mathbf{c}}^{l_2}, ..., \hat{\mathbf{c}}^{l_k}], \tag{44}$$

Where l_i refers to the ith filter. Furthermore, since it is studied that the bias of the triples refers the credibility of triple, the bias of triples are considered. The bias is concatenated with the max pooling layer outputs:

$$\mathbf{C} = [B_{max}, B_{avg}, \hat{\mathbf{c}}^{l_1}, \hat{\mathbf{c}}^{l_2}, ..., \hat{\mathbf{c}}^{l_k}], \tag{45}$$

Where $B_{max}, B_{avg} \in \mathbb{R}$ are the max bias and average bias of the triple set of the news. The calculation formula of the bias of one triple is defined as:

$$B_i = ||\mathbf{h}_i + \mathbf{r}_i - \mathbf{t}_i||_2^2. \tag{46}$$

The max bias and the average bias of one news is defined as:

$$B_{max} = max\{B_1, B_2, ..B_T\}, \tag{47}$$

$$B_{avg} = \frac{\sum_{i=1}^{T_{B_i}}}{T}. \tag{48}$$

Then, a full connection layer and a softmax layer are used to do the last classification. The full connection layer and the softmax layer are defined as the following:

$$\mathbf{Y} = [y_1, y_2] = \sigma(\mathbf{W}^y \cdot \mathbf{C} + \mathbf{b}^y) \tag{49}$$

$$\hat{y}_i = \frac{exp(y_i)}{\sum_{j=1}^{2} exp(y_j)} \tag{50}$$

Therefore, with a given news article, the predicted label of DTN is defined as:

$$\hat{y} = max\{\hat{y}_1, \hat{y}_2\}. \tag{51}$$

6. Evaluation

6.1. Dataset

Two popular topics are selected in the experiment: the first one is *2016 US Election* and the second one is *Brexit*.

6.1.1. True and fake news

A true and fake article base is created for each topic. The true news article base (TAB) is extracted from the BBC News, Sky News and The Independent. The fake news article base (FAB) is extracted from the fake news websites that are rated as fake news websites by IsItFakeNews [8], including InfoWars [9], Yournewswire.com [10], BeforeItsNews [11]. Each topic consists of near 1.4K articles in its corresponding article bases. In each topic, the training set includes 1000 true news and 1000 fake news. Moreover, the validation set has 200 true news and 200 fake news and the test set is about 200 true news and 200 fake news.

6.1.2. Knowledge graphs

Four knowledge graphs are proposed for each topic, which are described as the following: 1) FKG based on FAB 2) NKG based on TAB 3) DP4 (DBpedia 4-hop) from DBpedia 4) WD4 (Wikidata 4-hop) from Wikidata. Table 1 shows the basic statistics of the extracted knowledge graphs. The test datasets are available online: `http://knowledge-representation.org/j.z.pan/data/DTN_data-JoWS2020.zip`.

[8] `https://isitfakenews.com/`
[9] `https://www.infowars.com/`
[10] `https://yournewswire.com/`
[11] `https://beforeitsnews.com/v3/`

Table 1: The basic statistics of knowledge graphs on the two selected topics.

KG	*2016 US Election*			*Brexit*		
	Triples	Entities	Relations	Triples	Entities	Relations
FKG	8K	4K	12K	9K	3K	11K
NKG	15K	9K	19K	11K	4K	19K
DP4	132K	5K	312K	74K	4K	217K
WD4	117K	6K	287K	63K	3K	194K

6.2. Experiment setup

The knowledge graph embedding model and word embedding model are used to do triple embedding, and the dimension m is set to 32. For the knowledge graph embedding, TransE [48], TransH [49], TransR [50] and TransD [51] models trained by FKG, NKG, DB4 and WD4 are used to initialize $\mathbf{W}_k$. Moreover, for the word embedding, the Word2Vec model is used to initialize $\mathbf{W}_w$. For the training DTN model, the dropout rate and the learning rate are set to 0.7 and $1e-3$, respectively. Moreover, the filter window h and the mini-batch size are set to 2, 4, 6 and 128.

Firstly, the DTN model with different KG embedding models is tested to study if the KG embedding model will affect the results. In this group of comparative experiments, DB4 is used as the KG to train the KG embedding model. Secondly, TransD is used as the KG embedding model and test if different KG will affect the performance of DTN. Thirdly, based on TransD and DB4, an attention network is added on DTN to test the effectiveness of the attention mechanism. Finally, it is tested to see if the triple bias can improve the results based on TransD as the embedding model, DB4 as the KG.

6.3. Baseline methods

In this section, the proposed model is tested on various baseline methods: For the machine learning-based method, the term frequency-inverse document frequency (TF-IDF) is selected as the feature, which can reflect how important a word is to an article in the data set. Moreover, three kinds of classifications are selected: support vector machine (SVM) [68] can construct a hyperplane or set of hyperplanes in a high feature space for classification, while the input vectors are non-linear. Naive Bayes (NB) classifier [69] is a simple probabilistic classifier based on applying Bayes' theorem by assuming that all features are independent of each other. K-nearest neighbors (KNN) [70]

Table 2: Performance of different models on two selected topics.

Approach	*2016 US Election*			*Brexit*		
	Precision	Recall	F1	Precision	Recall	F1
TF-IDF+NB(articles)	0.851	0.814	0.830	**0.855**	0.840	0.847
TF-IDF+KNN(articles)	0.743	**0.903**	0.822	0.782	0.557	0.651
TF-IDF+SVM(articles)	**0.865**	0.837	**0.851**	0.848	**0.927**	**0.887**
TF-IDF+SVM(triples)	0.673	0.840	0.747	0.788	0.743	0.764
DTC(triples)	0.813	0.789	0.801	0.814	0.831	0.824
SVM-TS(triples)	0.822	0.813	0.817	0.847	0.872	0.859
textRNN(articles)	**0.927**	0.921	**0.924**	**0.915**	0.907	0.911
textCNN(articles)	0.895	**0.923**	0.909	0.910	**0.918**	**0.914**
textRNN(triples)	0.897	0.902	0.899	0.874	0.881	0.877
textCNN(triples)	0.883	0.870	0.876	0.868	0.877	0.872
DTN(triples)	**0.948**	**0.934**	**0.941**	**0.930**	**0.926**	**0.928**

is a non-parametric method used for the classification. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors.

Furthermore, two state-of-the-art models of rumor detection are selected as baselines. DTC [71] uses various handcrafted features and constructs a decision tree to detect the rumor. SVM-TS [72] applies a linear SVM classifier that leverages handcrafted features to detect the rumor. For the deep learning-based method, TextCNN [54] and TextRNN [61] are used.

6.4. Results

In this section, the results of baseline methods are compared with DTN on two datasets reported in table 1 and the comparison among different configurations of DTN is carried out. The metrics include the precision, recall rate, and F1 score.

6.4.1. Comparison of different models

Table 2 presents the performance of different models on the two topics, *2016 US Election* and *Brexit*. The obtained results are as the following:

1. Firstly, conventional text classification methods classify the articles with good classification results. However, for TF-IDF+SVM with triple inputs, its performance decreases significantly. More specifically, with triples as input, TF-IDF+SVM just achieves an F1 score of 74.7%, which is 15.5 percent lower than TF-IDF+SVM (articles). This is a reasonable result, since triples contain fewer word than whole articles, TF-IDF based models cannot capture enough information. This demonstrates that conventional models cannot extract features from the refined information.
2. Secondly, it is observed that deep learning models, such as textCNN and textRNN, perform significantly better than conventional text classification methods, such as TF-IDF and SVM. Deep learning models represent the input news better to extract better features. However, as with conventional text classification methods, deep learning models perform worse when the input is a triple set. This is due to the lack of an effective way to represent triples.
3. Unlike TextRNN and TextCNN treat all triples equally, entity-based attention networks make the DTN model pay more attention to important triples, which helps improve the DTN's performance.
4. In summary, DTN outperforms other existing models with the advantage of being able to provide triple based explanations. Compared with baseline models, the DTN model introduces knowledge graph embedding to obtain an effective representation of triples.

6.4.2. Comparison among different configurations of DTN

Table 3 presents the performance of different configurations on the topic *2016 US Election*. Firstly, experiments with different KG embedding models are carried out. The first observation is that using different KG embedding models does affect the results of the experiments. Among them: (1) TransD is the best model, slightly better than the TransR for the used data sets. The results show that using a better KG embedding model has some limited improvement in the experimental results, compared to the basic TransE model. (2) Secondly, comparative experiments with different knowledge graphs are performed, which train the KG embedding models with the TransD model, since the knowledge graph is another factor that affects the results. The results show that DB4 is the best

Table 3: Performance of different configurations of DTN on the topic of *2016 US Election*

Configurations	Precision	Recall	F1
DTN+DB4(TransE)	0.882	0.876	0.879
DTN+DB4(TransH)	0.893	0.887	0.890
DTN+DB4(TransR)	0.905	**0.913**	0.909
DTN+DB4(TransD)	**0.917**	0.910	**0.913**
DTN+FKG(TransD)	0.884	0.881	0.882
DTN+NKG(TransD)	0.898	0.892	0.895
DTN+DB4(TransD)	**0.917**	0.910	**0913**
DTN+WD4(TransD)	0.914	**0.911**	0.912
DTN+EAN(TransD+DB4)	**0.927**	**0.921**	**0.924**
DTN+Bias(TransD+DB4)	**0.929**	**0.915**	**0.922**
DTN+EAN+Bias(TransD+DB4)	**0.948**	**0.934**	**0.941**

among all of the tested knowledge graphs. A possible explanation of this result is that the numbers of entities and relations and the quality of the KG will affect the results. (3) Thirdly, it is intended to use the entity-based attention network (EAN) with the DTN model. The experiment result shows that the entity-based attention network is useful. Therefore, it is necessary to quantify the importance of each entity and give more weight to the important entity/triple. (4) Finally, the application of the triple bias is tested, which is an extra input to the fully connected layer to provide more information, and the result shows it can make further improvement.

6.5. Case study

Figures 3 and 4 show two test examples on real news articles on the topic of *2016 US Election* to display the ability of DTN in detecting fake news and providing explanations. The "fake" or "true" is the DTN result and the "×"(wrong) or "✓"(right) of every triple is done by manual checking. The green and red parts represent the true and fake based on the proposed DTN. Moreover, the attention weight refers to the sum value of two entity attention weights.

Figure 3 is an example of fake news. It is worth noting that not every sentence is fake in a fake news. Out of the 8 triples here, only 2 of them are true. The fake parts with high probability include (*Hillary*, *sell weapons to*, *ISIS*), (*Hillary Clinton*, *arm*, *Islamic jihadists*). Thus, even though DTN has two false positives here that include (*email*, *are*, *real*), DTN is still able to reach the correct judgement (fake, in this case), due to the fact that the fake triples, such as (*Hillary Clinton*, *arm*, *Islamic jihadists*) and (*Hillary Clinton*, *commit*, *perjury*), have higher attention weight. Figure 4 is an example of true news. Theoretically, every triple in the true news should be true. DTN makes a correct overall judgement (true, in this case) despite that two triples are misjudged here, including (*agents*, *examine*, *emails*) Since DTN has higher attention weight to the important entity and triple.

7. Conclusion

Motivated by the need to evaluate the reliability of news articles rather than short texts accurately and convincingly, in the present study, a novel multi-channel deep triple network is proposed for fake news detection on specific topics. In the proposed model, knowledge graphs contribute to fake news detection in two ways: One is to involve background knowledge by introducing entity and relation via triple extraction from input texts on the same topic, with semantic-level representation and background knowledge-level representation integrating as the triple-based multi-channel encoding model. The other is the knowledge-aware enhanced explanation of DTN, which improves the trustworthiness of the proposed model, and gets insights into predictions. The quantitative result shows the effectiveness of the proposed DTN and the qualitative results on the triple attention weight validate the interpretability of DTN in the sentence interaction within contents. Furthermore, DTN can investigate the KG scheme to improve triple extraction and contributing to the ontology correctness.

In the future, we will investigate how to exploit better quality knowledge graph embedding and completion techniques, such as the RotatE [73] and schema aware knowledge graph completion [47].

Acknowledgement

This paper is supported by Aberdeen-Wuhan Joint Research Institute, National Key R&D Plan of China

Title	WikiLeaks Confirms Hillary Sold Weapons To ISIS	
Content	Julian Assange, the founder of WikiLeaks, is a controversial character. But there's no denying the emails he has picked up from inside the Democrat Party are real, and he's willing to expose Hillary Clinton. Now, he's announcing that Hillary Clinton and her State Department were actively arming Islamic jihadists, which includes the Islamic State (ISIS) in Syria. Clinton has repeatedly denied these claims, including during multiple statements while under oath in front of the United States Senate. WikiLeaks is about to prove Hillary Clinton deserves to be arrested. It appears that Hillary Clinton committed perjury, just like her husband was caught doing as President.	
Ground truth	Fake	
DTN prediction	Fake √	
Triples	**Attention weight**	**DTN prediction**
(WikiLeaks, confirm, Hillary sell weapons)	0.129	true ×
(Hillary, sell weapons to, ISIS)	0.045	fake √
(Julian Assange, founder of, WikiLeaks)	0.153	true √
(emails, from, the Democrat Party)	0.218	true √
(emails, are, real)	0.037	true ×
(Hillary Clinton, arm, Islamic jihadists)	0.302	fake √
(Hillary Clinton, deserve to, be arrested)	0.304	fake √
(Hillary Clinton, commit, perjury)	0.308	fake √

Figure 3: Case Study 1. An example of **fake** news on the topic of *2016 US Election*

Title	FBI Examining Hillary Clinton Emails Found in Anthony Weiner Probe	
Content	The FBI is examining additional emails related to Hillary Clinton's use of a private email server four months after closing its investigation, the bureau's director said in a letter to lawmakers on Friday, a surprise twist in a turbulent campaign just 11 days before election day. In a note to congressional committee chairs, FBI director James Comey said that the FBI had discovered additional emails relevant to the investigation into Clinton's server and agents were examining the emails to determine whether they contain classified information. The newly discovered emails were found on at least one device belonging to longtime Clinton aide Huma Abedin and her husband, former congressman Anthony Weiner, as part of an investigation into Weiner's sexting scandal, multiple news outlets reported citing law enforcement officials……	
Ground truth	True	
DTN prediction	True √	
Triples	**Attention weight**	**DTN prediction**
(FBI, examine, Hillary Clinton emails)	0.027	true √
(email server, is of, private)	0.780	true √
(FBI, discover, additional emails)	0.032	true √
(emails, is, relevant to investigation into Clinton 's server)	0.259	fake ×
(agents, examine, emails)	0.057	fake ×
(Huma Abedin's husband, former congressman, Anthony Weiner)	0.067	true √
(congressman Anthony Weiner, is, former)	0.235	true √
(investigation, is into, Weiner 's sexting scandal)	0.123	true √

Figure 4: Case Study 2. An example of **true** news on the topic of *2016 US Election*

2020YFA0607902, National Natural Science Foundation of China U1936107, and the EU K-Drive project (286348).

References

[1] G. A.Hunt, Social media and fake news in the 2016 election, Journal of Economic Perspectives 31 (2017) 211–36.

[2] M. Granik, V. Mesyura, Fake news detection using naive bayes classifier, in: Electrical and Computer Engineering (UKRCON), 2017 IEEE First Ukraine Conference on, IEEE, 2017, pp. 900–903.

[3] S. Gilda, Evaluating machine learning algorithms for fake news detection, in: Research and Development (SCOReD), 2017 IEEE 15th Student Conference on, IEEE, 2017, pp. 110–115.

[4] W. Y. Wang, "liar, liar pants on fire": A new benchmark dataset for fake news detection, arXiv preprint arXiv:1705.00648 (2017).

[5] B. Shi, T. Weninger, Fact checking in heterogeneous information networks, in: Proceedings of the 25th International Conference Companion on World Wide Web, International World Wide Web Conferences Steering Committee, 2016, pp. 101–102.

[6] G. L. Ciampaglia, P. Shiralkar, L. M. Rocha, J. Bollen, F. Menczer, A. Flammini, Computational fact checking from knowledge networks, PloS one 10 (2015) e0128193.

[7] P. Shiralkar, A. Flammini, F. Menczer, G. L. Ciampaglia, Finding streams in knowledge graphs to support fact checking, in: Data Mining (ICDM), 2017 IEEE International Conference on, IEEE, 2017, pp. 859–864.

[8] F. C. Fernández-Reyes, S. Shinde, Evaluating deep neural networks for automatic fake news detection in political domain, in: Ibero-American Conference on Artificial Intelligence, Springer, 2018, pp. 206–216.

[9] Y. Liu, K. Yu, X. Wu, L. Qing, Y. Peng, Analysis and detection of health-related misinformation on chinese social media, IEEE Access 7 (2019) 154480–154489.

[10] H. Samuel, O. Zaiane, Medfact: Towards improving veracity of medical information in social media using applied machine learning, in: Canadian Conference on Artificial Intelligence, Springer, 2018, pp. 108–120.

[11] M. Rajdev, K. Lee, Fake and spam messages: Detecting misinformation during natural disasters on social media, in: 2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), volume 1, IEEE, 2015, pp. 17–20.

[12] J. Pan, G. Vetere, J. Gomez-Perez, H. Wu (Eds.), Exploiting Linked Data and Knowledge Graphs for Large Organisations, Springer, 2017.

[13] J. Pan, D. Calvanese, T. Eiter, I. Horrocks, M. Kifer, F. Lin, Y. Zhao (Eds.), Reasoning Web: Logical Foundation of Knowledge Graph Construction and Querying Answering, Springer, 2017.

[14] B. Yang, T. Mitchell, Leveraging knowledge bases in lstms for improving machine reading, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), volume 1, 2017, pp. 1436–1446.

[15] J. Wang, Z. Wang, D. Zhang, J. Yan, Combining knowledge with deep convolutional neural networks for short text classification, in: Proceedings of IJCAI, volume 350, 2017.

[16] H. Wang, F. Zhang, X. Xie, M. Guo, Dkn: Deep knowledge-aware network for news recommendation, arXiv preprint arXiv:1801.08284 (2018).

[17] J. Z. Pan, S. Taylor, E. Thomas, Reducing Ambiguity in Tagging Systems with Folksonomy Search Expansion, in: the Proc. of the 6th European Semantic Web Conference (ESWC2009), 2009.

[18] L. Dietz, A. Kotov, E. Meij, Utilizing knowledge graphs for text-centric information retrieval, in: Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2018), 2018, pp. 1387–1390.

[19] Y. Gu, T. Zhou, G. Cheng, Z. Li, J. Z. Pan, Y. Qu, Relevance Search over Schema-Rich Knowledge Graphs, in: Proc. of the 12th ACM International WSDM Conference (WSDM2019), 2019, pp. 114–122.

[20] X. Huang, J. Zhang, D. Li, P. Li, Knowledge Graph Embedding Based Question Answering, in: Proc. of 18th International Semantic Web Conference (ISWC 2019), 2019, p. 105–113.

[21] J. Z. Pan, M. Zhang, K. Singh, F. V. Harmelen, J. Gu, Z. Zhang, Entity Enabled Relation Linking, in: Proc. of 18th International Semantic Web Conference (ISWC 2019), 2019, pp. 523–538.

[22] K. Singh, M. Saleem, A. Nadgeri, F. Conrads, A.-C. N. N. Jeff Z. Pan, J. Lehmann, QaldGen: Towards Microbenchmarking of Question Answering Systems over Knowledge Graphs, in: Proc. of 18th International Semantic Web Conference (ISWC 2019), 2019, pp. 277–292.

[23] B. Fawei, J. Z. Pan, M. J. Kollingbaum, A. Z. Wyner, A Semi-automated Ontology Construction for Legal Question Answering, New Generation Computing (2020) 453–478.

[24] H. He, A. Balakrishnan, M. Eric, P. Liang, Learning Symmetric Collaborative Dialogue Agents with Dynamic Knowledge Graph Embeddings, in: Proc. of ACL2017, 2017.

[25] X. Wang, X. He, Y. Cao, M. Liu, T.-S. Chua, KGAT: Knowledge Graph Attention Network for Recommendation, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2019), 2019, pp. 950–958.

[26] H. Wang, F. Zhang, M. Zhao, W. Li, X. Xie, M. Guo, Multi-Task Feature Learning for Knowledge Graph Enhanced Recommendation, in: Proc. of 28th the Web Conference, 2019.

[27] H. Wu, M. Wang, Q. Zeng, W. Chen, T. Nind, E. R. Jefferson, M. Bennie, C. Black, J. Z. Pan, C. Sudlow, D. Robertson, Knowledge Driven Phenotyping, in: Proc. of Medical Informatics Europe (MIE 2020), 2020, pp. 1327–1328.

[28] I. J. Tripodi, T. J. Callahan, J. T. Westfall, N. S. Meitzer, R. D. Dowell, L. E. Hunter, Applying knowledge-driven mechanistic inference to toxicogenomics, Toxicology in Vitro (2020).

[29] S. Deng, N. Zhang, W. Zhang, J. Chen, J. Z. Pan, H. Chen, Knowledge-Driven Stock Trend Prediction and Explanation via Temporal Convolutional Network, in: Proc. of the World Wide Web Conference (WWW 2019), 2019, pp. 678–685.

[30] H. Xu, F. Giunchiglia, Sko types: an entity-based scientific knowledge objects metadata schema, Journal of Knowledge Management 19 (2015) 60–70.

[31] S. Auer, V. Kovtun, M. Prinz, A. Kasprzik, M. Stocker, M.-E. Vidal, Towards a Knowledge Graph for Science, in: Proc. of the 8th International Conference on Web Intelligence, Mining and Semantics (WIMS 2018), 2018, pp. 1327–1328.

[32] P. Banksy, E. Edelstein, J. Z. Pan, A. Wyner, A Dynamic and Informative Intelligent Survey System, in: Proc. of the Joint International Semantic Technology Conference (JIST2019), 2019, pp. 226–241.

[33] E. Edelstein, J. Z. Pan, R. Soares, A. Wyner, Knowledge-driven intelligent survey systems towards open science, New Generation Computing (2020) 397–421.

[34] J. Z. Pan, S. Pavlova, C. Li, N. Li, Y. Li, J. Liu, Content based Fake News Detection Using Knowledge Graphs, in: Proc. of the International Semantic Web Conference (ISWC2018), 2018, pp. 669–683.

[35] J. Z. Pan, S. Staab, U. Aßmann, J. Ebert, Y. Zhao (Eds.), Ontology-Driven Software Development, Springer, 2013.

[36] J. Z. Pan, Y. Zhao (Eds.), Semantic Web Enabled Software Engineering, IOS Press, 2014.

[37] K. Siegemund, E. Thomas, Y. Zhao, J. Z. Pan, U. Assmann, Towards Ontology-driven Requirements Engineering, in: Proc. of the International Workshop on Semantic Web Enabled Software Engineering (SWESE2011), 2011.

[38] S. Taylor, N. Jekjantuk, C. Mellish, J. Z. Pan, Reasoning Driven Configuration of Linked Data Content Management Systems, in: Proc. of the 3rd Joint International Conference on Semantic Technologies (JIST 2013), 2013.

[39] J. Chen, H. Chen, J. Z. Pan, M. Wu, N. Zhang, G. Zheng,

When Big Data Meets Big Smog: a Big Spatio-temporal Data Framework for China Severe Smog Analysis, in: Proc. of the 2nd ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data (BigSpatial@SIGSPATIAL 2013), 2013.
[40] S. R. Bader, I. Grangel-González, P. Nanjappa, M.-E. Vidal, M. Maleshkova, A Knowledge Graph for Industry 4.0, in: Proceedings of the 17th Extended Semantic Web Conference (ESWC 2020), 2020, pp. 465–480.
[41] J. Chen, F. Lecue, J. Z. Pan, I. Horrocks, H. Chen, Knowledge-based Transfer Learning Explanation, in: Proc. of the International Conference on Principles of Knowledge Representation and Reasoning (KR2018), 2018, pp. 349–358.
[42] F. Lecue, J. Chen, J. Z. Pan, H. Chen, Augmenting Transfer Learning with Semantic Reasoning, in: Proc. of 28th International Joint Conference on Artificial Intelligence (IJCAI 2019), 2019, pp. 1779–1785.
[43] J. Chen, F. Lecue, Y. Geng, J. Z. Pan, H. Chen, Ontology-guided Semantic Composition for Zero-shot Learning, in: Proc. of 17th International Conference on Knowledge Representation and Reasoning (KR 2020), 2020, pp. 850–854.
[44] Y. Geng, J. Chen, Z. Chen, J. Z. Pan, Z. Ye, Z. Yuan, Y. Jia, H. Chen, OntoZSL: Ontology-enhanced Zero-shot Learning, in: Proc. of 30th The Web Conference (WWW 2021), 2021.
[45] B. Shi, T. Weninger, Open-world knowledge graph completion, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.
[46] D. K. ande Volker Tresp, Ensemble solutions for link-prediction in knowledge graphs, in: Proc. of 2nd Workshop on Linked Data for Knowledge Discovery, 2015.
[47] K. Wiharja, J. Z. Pan, M. J. Kollingbaum, Y. Deng, Schema Aware Iterative Knowledge Graph Completion, Journal of Web Semantics (2020).
[48] G.-D. A. Bordes A, Usunier N, Translating embeddings for modeling multi-relational data (2013) 2787–2795.
[49] F. J. Wang Z, Zhang J, Knowledge graph embedding by translating on hyperplanes (2014) 1112–1119.
[50] Z. X. Lin Y, Liu Z, Learning entity and relation embeddings for knowledge graph completion (2015) 2181–2187.
[51] G. Ji, S. He, L. Xu, K. Liu, J. Zhao, Knowledge graph embedding via dynamic mapping matrix, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), volume 1, 2015, pp. 687–696.
[52] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, L. D. Jackel, Handwritten digit recognition with a back-propagation network, in: D. S. Touretzky (Ed.), Advances in Neural Information Processing Systems 2, Morgan-Kaufmann, 1990, pp. 396–404.
[53] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al., Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (1998) 2278–2324.
[54] Y. Kim, Convolutional neural networks for sentence classification, arXiv preprint arXiv:1408.5882 (2014).
[55] T. Mikolov, A. Joulin, S. Chopra, M. Mathieu, M. Ranzato, Learning longer memory in recurrent neural networks, arXiv preprint arXiv:1412.7753 (2014).
[56] I. Sutskever, J. Martens, G. E. Hinton, Generating text with recurrent neural networks, in: Proceedings of the 28th International Conference on Machine Learning (ICML-11), 2011, pp. 1017–1024.
[57] S. Rajaram, P. Gupta, B. Andrassy, T. Runkler, Neural architectures for relation extraction within and across sentence boundaries in natural language text (2018). doi:10.13140/RG.2.2.36649.70245.
[58] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, E. Hovy, Hierarchical attention networks for document classification, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, San Diego, California, 2016, pp. 1480–1489. URL: https://www.aclweb.org/anthology/N16-1174. doi:10.18653/v1/N16-1174.
[59] K. Popat, S. Mukherjee, A. Yates, G. Weikum, Declare: Debunking fake news and false claims using evidence-aware deep learning, arXiv preprint arXiv:1809.06416 (2018).
[60] A. Joulin, E. Grave, P. Bojanowski, T. Mikolov, Bag of tricks for efficient text classification, arXiv preprint arXiv:1607.01759 (2016).
[61] S. Lai, L. Xu, K. Liu, J. Zhao, Recurrent convolutional neural networks for text classification., in: AAAI, volume 333, 2015, pp. 2267–2273.
[62] K. Annervaz, S. B. R. Chowdhury, A. Dukkipati, Learning beyond datasets: Knowledge graph augmented neural networks for natural language processing, arXiv preprint arXiv:1802.05930 (2018).
[63] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in neural information processing systems, 2017, pp. 5998–6008.
[64] Z. Huang, W. Xu, K. Yu, Bidirectional lstm-crf models for sequence tagging, arXiv preprint arXiv:1508.01991 (2015).
[65] E. B. Duc-Thuan Vo, Open information extraction (2016) 1–8.
[66] J. R. Finkel, T. Grenager, C. Manning, Incorporating non-local information into information extraction systems by gibbs sampling, in: Proceedings of the 43rd annual meeting on association for computational linguistics, Association for Computational Linguistics, 2005, pp. 363–370.
[67] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (1997) 1735–1780.
[68] C. Cortes, V. Vapnik, Support-vector networks, Machine learning 20 (1995) 273–297.
[69] A. McCallum, K. Nigam, et al., A comparison of event models for naive bayes text classification, in: AAAI-98 workshop on learning for text categorization, volume 752, Citeseer, 1998, pp. 41–48.
[70] L. E. Peterson, K-nearest neighbor, Scholarpedia 4 (2009) 1883.
[71] C. Castillo, M. Mendoza, B. Poblete, Information credibility on twitter, in: Proceedings of the 20th international conference on World wide web, 2011, pp. 675–684.
[72] J. Ma, W. Gao, Z. Wei, Y. Lu, K.-F. Wong, Detect rumors using time series of social context information on microblogging websites, in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, 2015, pp. 1751–1754.
[73] Z. Sun, Z.-H. Deng, J.-Y. Nie, J. Tang, Rotate: Knowledge graph embedding by relational rotation in complex space., in: Proceedings of ICLR2019, 2019.