Joint Representations of Knowledge Graphs and Textual Information via Reference Sentences

Zizheng JI(a), Member, Zhengchao LEI(b), Tingting SHEN†(c), and Jing ZHANG††(d), Nonmembers

SUMMARY The joint representations of knowledge graph have become an important approach to improve the quality of knowledge graph, which is beneficial to machine learning, data mining, and artificial intelligence applications. However, the previous work suffers severely from the noise in text when modeling the text information. To overcome this problem, this paper mines the high-quality reference sentences of the entities in the knowledge graph, to enhance the representation ability of the entities. A novel framework for joint representation learning of knowledge graphs and text information based on reference sentence noise-reduction is proposed, which embeds the entity, the relations, and the words into a unified vector space. The proposed framework consists of knowledge graph representation learning module, textual relation representation learning module, and textual entity representation learning module. Experiments on entity prediction, relation prediction, and triple classification tasks are conducted, results show that the proposed framework can significantly improve the performance of mining and fusing the text information. Especially, compared with the state-of-the-art method [19], the proposed framework improves the metric of H@10 by 5.08% and 3.93% in entity prediction task and relation prediction task, respectively, and improves the metric of accuracy by 5.08% in triple classification task.

key words: joint representation learning, knowledge graph, reference sentence, attention mechanism

1. Introduction

Knowledge Graphs (KGs) are graph-structured knowledge bases, where factual knowledge is represented in the form of relations between entities. KGs have become a crucial resource for many tasks in machine learning, data mining, and artificial intelligence applications including question and answer systems [3], Web search [4] and many other natural language processing tasks. The application of knowledge graph often suffers severely from data sparsity and low computational efficiency with the growth of knowledge scale. Recent years have witnessed the great advance of representation learning (RL) models for knowledge graph, which represents entities and relations as elements of a continuous vector space, and enhances the knowledge learning ability in terms of knowledge reasoning, knowledge fusion, and knowledge completion [5]–[10]. Motivated by the linear translation phenomenon observed in well trained word embedding, the translation-based model [10]–[12] is the current baseline algorithm for knowledge graph representation learning. The algorithm views the relation as a translation phenomenon, and it has played an important auxiliary role in optimizing knowledge graph representation learning based on translation-based model [13]–[15].

The combination of knowledge graph and textual information has become a hot topic of current research. Based on the representation of entities and relations in the knowledge graph as vectors in low-dimensional semantic space, the task mines entities and relations from text and maps them to the same semantic space. Wang [16] and Toutanova [17] proposed a joint representation learning model that maps entities in the knowledge graph and words into the text to the same vector space through the alignment mechanism, and Han [15] modeled the context information to some extent. The representation learning models have achieved better efficiency by leveraging textual information. However, the noise in the text greatly affects the knowledge representation ability.

To solve this problem, we introduce the “reference sentence” to model the entities in knowledge graph. The “referential sentence” refers to a sentence in a high-quality corpus or knowledge base that contains the entity to be modeled and can give an accurate definition or connotation interpretation of the entity at the semantic level as shown in Fig. 1. Moreover, similar to human behaviors, “reference sentence” often employs semantic interpretations to learn the meaning of new words [18]. In the process of entity modeling, using

![Fig. 1](image-url) An example of the reference sentence of Wikipedia about Cristiano Ronaldo.
the “reference sentences” to analyze the meaning of the inferred entity can effectively alleviate the influence of textual noise and improve the semantic expression ability.

This paper proposes a novel framework for joint representation learning of knowledge graphs and textual information via reference sentences. The framework includes knowledge graph representation learning module, textual relation representation learning module, and textual entity representation learning module. The knowledge graph representation learning module is used to model the knowledge graph. Two other models fully exploit the rich semantic information contained in the text to improve knowledge modeling. Specifically, in the textual entity representation learning module, the attention mechanism is used to automatically select the most appropriate reference sentences to alleviate the interference of textual noise. The three modules mentioned above enhance the ability to represent the potential interaction between the knowledge graph representation learning process and the textual information representation learning process by constructing a “tightly coupled” loss function.

Extensive experiments on the tasks of entity prediction, relation prediction, and triple classification are conducted. The results show that the proposed model significantly outperforms the baseline algorithm.

2. Related Work

2.1 Knowledge Graph Representation Learning

Motivated by the linear translation phenomenon observed in well-trained word embeddings, many Translating-based models have been proposed, aiming at embedding entities and relations into a vector space and predicting the missing element of triples. Inspired by the study of word embedding [19], TransE [10] considers the relation \( r \) in each triple \( (h, r, t) \) as the “translation” of the head entity \( h \) to the tail entity \( t \) in low-dimensional space. That is, \( h + r = t \), where \( h, t, \) and \( r \) represent vectors of the head entity, the tail entity, and the relation, respectively. TransE can achieve baseline effects on large-scale, sparse knowledge graphs. However, it cannot effectively model more complex relations such as 1-to-N, N-to-1, N-to-N, and cannot be used for long-distance modeling in knowledge graph. To solve the above problems, TransH [11] transfers the translation between entities to the hyperplane of the particular relation. TransR [12] models the entities and relations in different semantic spaces and then learns the relations between entities while maps the entities from physical space to relational space. To improve the performance of the translation based methods, Guu [19] utilizes the representation learning of multi-hop relation paths.

2.2 Joint Representation Learning of Knowledge Graph and Textual Information

Textual information played a complementary role in knowledge graph representation learning since it’s ability to model the knowledge from a different [15], [20]–[22]. Wang [16] encoded entities and words into a low-dimensional vector space simultaneously by aligning the entity references in the associated text with the Wikipedia anchor text. Zhong [23] introduced entity descriptions for entities and words. Toutanova [17] used dependency syntax analysis to enhance the representation of textual relations. Han [15] proposed a model that mapped words, entities and relations to the same semantic vector space, while focused on mining textual relations. Xie [6] utilized convolutional neural networks (CNN) to construct entity representations from entity descriptions. Yamada [24] and Cao [25] proposed the multivariate entity reference embedding model and modeled the various connotations for each entity by jointly using knowledge graph and text to solve the entity disambiguation.

3. Framework of Joint Representation Learning

For each triplet \((h, r, t) \in T\), it contains two entities \( h, t \in \mathbb{E} \) and a relation \( r \in \mathbb{R} \), in which \( \mathbb{E} \) represents the entity set, and \( \mathbb{R} \) represents the relation set. \( T \) is the triple set, and \( \mathbb{G} \) is knowledge graph. In this paper, \( \mathbb{D} \) denotes to the corpus, while \( \mathbb{V} \) denotes to the dictionary. For convenience of expression, notation “words” in this paper refers to “words” and “phrases”.

The dimensions of the entity vector and relation vector are both \( k \). There are two kinds of vector representations for each entity. For \( h \) and \( t \), the entity vector representations obtained from the knowledge graph are called “Structure Representation”, represented as \( h_\mathbb{G} \) and \( t_\mathbb{G} \) respectively. On the other hand, the representations learned from plain texts are called “Text Representation”, represented as \( h_\mathbb{D} \) and \( t_\mathbb{D} \) respectively. The textual representation of the words in dictionary \( \mathbb{V} \) is constructed only from corpus \( \mathbb{D} \). Representations of the relation learned from knowledge graph are donated as \( r_\mathbb{G} \), and the relation between two entities extracted from texts is donated as \( r_\mathbb{D} \).

3.1 Overall Framework

The joint representation of entities, relations, and words is defined as parameters \( \theta = \{ \theta_\mathbb{G}, \theta_\mathbb{D}, \theta_\mathbb{V} \} \), wherein: (i) \( \theta_\mathbb{G} \) denotes the set of parameters for entity representation learning from knowledge graph (i.e., the structural representation of entity); (ii) \( \theta_\mathbb{D} \) denotes the set of parameters for representation learning from plain text (i.e., the textual representation of entity); (iii) \( \theta_\mathbb{V} \) denotes the set of parameters for relation representation learning from knowledge graph (i.e., the structural representation of relation); (iv) \( \theta_v \) denotes the set of parameter for relation representation learning from plain text (i.e., the textual representation of relation); (v) \( \theta_v \) denotes the set of parameter of word representation in \( \mathbb{V} \).

The objective function to be minimized

\[
\hat{\Theta} = \arg \min_{\Theta} L_{\Theta}(\mathbb{G}, \mathbb{D}),
\]

in which \( L_{\Theta}(\mathbb{G}, \mathbb{D}) \) is the loss defined in \( \mathbb{G} \) and \( \mathbb{D} \).
In Eq. (2), $\alpha$, $\beta$, and $\lambda$ represent the Harmonic Factors. $\alpha$ and $\beta$ are used to balance the ratio between the knowledge graph and plain text. $||\theta||_2$ is the L2 norm of $\theta$. $\mathcal{L}_{\Theta_{h_0} \Theta_{t_0}}(G, D)$ is to utilized learn the structural representations of entities and vector representations of the relations from $G$. $\mathcal{L}_{\Theta_{h_0} \Theta_{t_0}}(G, D)$ is to utilized learn the vector representation of the relation from $D$. $\mathcal{L}_{\Theta_{h_0} \Theta_{t_0}}(G, D)$ is to learn the textual representation from $D$. The loss function defined in Eq. (2) is capable of tightly coupling the representation learning process of knowledge graph with the representation learning process of textual information. Word vectors are learned with Skip-Gram model based on negative sampling [28]. Since entities and relations are not explicitly labeled in plain text, alignment is needed to be identified from plain texts to support the learning of representations for entities and relations, which is fulfilled through “entity-text” and “relation-text” alignment.

1. “Entity-Text” Alignment

There are various entities exits in plain text, such as reference sentences. Since the entities mentioned may have complex cases for ambiguity. For example, the entity “Independence Day” may refer to either a movie or a US holiday. Therefore, it is necessary to construct the entity-text alignment. The alignment can be built by using Entity Linking technology or Anchor Text information of Wikipedia [16]. In this paper, if an entity from $D$ mentioned in $V$ aligns with an entity from $G$, the vector representation of this entity is set as the entity vector learned from $G$.

2. “relation-text” Alignment

There exists many studies investigating to extract textual relations from texts [7], [29]. For the relation $r \in \mathbb{R}$ defined in $G$, all the entity pairs $\Delta_r = \{(h, t)\}$ corresponding to relation $r$ will be gathered from $G$. Subsequently, for each entity pair $(h, t) \in \Delta_r$, all the sentences containing both head entity $h$ and tail entity $t$ are extracted from $D$ and used as the positive samples of $r$. These sentences are encoded into relation representations with deep neural network.

As shown in Fig. 2, we propose the joint representation learning framework of knowledge graph and textual information reference sentences, which consists of three modules:

1. Representation learning module of knowledge graph (parts with the yellow background in Fig. 2). TransE is used to learn the vector representation $h_G$ and $t_G$ of entities, and vector representation of relation $r_G$ from $G$. Note that, arbitrary representation learning models could be adopted here, because of the generality of proposed framework.

2. Representation learning module of textual relations (parts with green background in Fig. 2)) is used for learning the textual relation representation $r_D$. Sentences with $h$ and $t$ are modeled with convolutional neural network with Distant Supervision strategy [30].

3. Representation learning module of textual entities (parts with pink background in Fig. 2) is used for learning textual representation $h_D$ and $t_D$ of entities. Long and short-term memory (LSTM) network is used to encode generate the vector representations of reference sentences for an entity. Then the attention model is introduced to choose the top-$m$ most informative sentences to generate the textual representation of the given entity.

The whole network will be optimized by stochastic gradient descent (SGD).

3.2 Representation Learning of Knowledge Graph

Translation-based model [10] is used to learn the vector representations of entities and relations from knowledge graph. In the following part, we will show the modeling process of TransE. In experiments, we compare the results using...
different approaches based on translation models (TransE, TransR, and TransH).

For each entity pair \((h, t)\) in \(\mathbb{G}\), we set \(r_{ht}\) is the “translation” from \(h\) to \(t\).

\[
r_{ht} = t - h
\]

(3)

As each triple \((h, r, t)\) ∈ \(\mathcal{T}\) indicates that there is an explicit relation vector \(r_0\) between \(h\) and \(t\), thus the scoring function for the triple is defined as

\[
\varphi_r(h, t) = \|r_{ht} - r_0\|_2 = \|(t - h) - r_0\|_2
\]

(4)

Equation (4) shows that, for each triple \((h, r, t)\) ∈ \(\mathcal{T}\), we expect

\[
t \approx h + r.
\]

Therefore, the loss function for all the triples in \(\mathcal{T}\) is defined as

\[
L_{\Theta_{\theta_{\psi}}}(\mathcal{G}) = \sum_{(h, r, t) \in \mathcal{T}} \sum_{(h', r', t') \in \mathcal{T}} \left[ \mu + \varphi_r(h, t) - \varphi_r(h', t') \right]_+.
\]

(5)

in which \(\mathcal{T}\) is the triples in a knowledge base, while \(\mathcal{T}'\) is the negative triples. \(\mathcal{T}'\) is constructed as follows

\[
\mathcal{T}' = \{(h', r, t)\} \cup \{(h, r', t)\} \cup \{(h, r, t')\}.
\]

(6)

\(h' \in \mathbb{E}\) is the head entity obtained by random sampling, \(r' \in \mathbb{R}\) represents the relation obtained by random sampling, and \(t' \in \mathbb{E}\) represents the tail entity obtained by random sampling. \(\mu\) is the spacing parameter greater than 0. \([x]_+\) stands for a positive function defined as

\[
[x]_+ = \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases}
\]

(7)

Vector representations of entities and relations in \(\mathbb{G}\) will be obtained after the training completed.

3.3 Representation Learning of Textual Relations

Given a sentence containing two entities, the words in the sentence (especially the combination of consecutive words) tend to reveal the implicit features of the textual relation between two entities. Xu [20] and Sorokin [31] show that textual relations can be learned with deep neural network and encoded in the low-dimensional semantic space. Compared with traditional algorithm [30], the deep learning-based algorithm can accurately capture the semantic relations between entities from texts without using explicit syntax information [29], [32]. In our work, CNN is used to extract the textual relations from plain texts. For sentence \(s = (x_1, x_2, \ldots, x_n)\) containing entity pair \((h, t)\), if relation \(r\) defined in \(\mathbb{G}\) exists in the entity pair, the word vector \(x_1, x_2, \ldots, x_n\) in sentence \(s\) acts as input, while \(r_D\) is the output obtained by CNN. The minimum distance between \(r_G\) and \(r_D\) is defined as the loss function for CNN.

\[
\psi_r(s) = \|r_D - r_G\|_2
\]

(8)

Therefore, the loss function defined for all sentences of \(\mathcal{D}\) is

\[
L_{\Theta_{\theta_{\psi}}}(\mathbb{G}, \mathcal{D}) = \sum_{s \in \mathcal{D}} \sum_{r \in r} [\gamma + \psi_r(s) + \psi_r(s)]_+.
\]

(9)

While \(\gamma\) is a separation parameter greater than 0.

3.4 Representation Learning of Textual Entities

Representation learning model of textual entities aims to reason and model the semantic connotation of the entity from its reference sentence to reduce the impact of noise on entity representation learning. The scoring function is defined for each triple

\[
\sigma_r(h, t) = \|(t_D - h_D - r_D)\|_2
\]

(10)

Wherein the head entity vector \(h_D\) and tail entity vector \(t_D\) are all textual representations learned from reference sentences. The loss function of all triples in \(\mathcal{T}\) is defined as

\[
L(\mathcal{D}) = \sum_{(h, r, t) \in \mathcal{T}} \sum_{(h', r', t') \in \mathcal{T}} [\eta + \sigma_r(h, t) - \sigma_r(h', t')]_+.
\]

(11)

Wherein \(\eta\) is the interval parameter greater than 0. \(\mathcal{T}\) is set of all triples of a knowledge base. \(\mathcal{T}'\) is the negative samples constructed by

\[
\mathcal{T}' = \{(h', r, t)\} \cup \{(h, r, t')\}
\]

(12)

3.4.1 Word Representation

Word representation generally consists of word content feature and word position feature [33]. In this paper, Skip-Gram based negative sampling is used to learn the word embedding, which can be directly used as word content feature. Words order help us understand the sentence better [35], thus it is necessary to take the position of entity in reference sentence into account. Given a sentence, the position feature of the entity is set to 0, and the other words are labeled as the relative distance from this entity. The position feature of the word on the left side is a negative value, while on the right side is positive. At the same time, the upper of the relative distance is set to \(d\). If the relative distance is greater than \(d\), the value will be \(-d\) or \(d\).

3.4.2 Reference Sentence Encoder

There are already many algorithms take the word order into account to represent semantic information, such as Recurrent Neural Network (RNN) based methods. Such approaches have been widely used in many Natural Language Processing tasks. This paper uses LSTM to encode the meanings of entities from reference sentences.
A reference sentence is fed in RNN in the representation learning of textual entity proposed in this paper. In conventional RNN, RNN maintains a hidden state \( h \) over time. At each time step \( i \), the hidden state \( h_i \) is updated with the formula

\[
h_i = \tanh(Wx_i + Uh_{i-1} + b).
\] (13)

Each word representation of the input sentence is fed into RNN one by one. The hidden state in RNN is dynamically adjusted according to Eq. (13). While finished, \( h_n \) will be output, and \( n \) indicates the number of words in the sentence. The final hidden state vector \( h_n \) is considered as sentence-level representation. As \( h_n \) is representation of reference sentence \( s_e \) related with entity \( e \), \( h_n \) is mark as \( s_e \) in this paper.

Although RNN has been widely used in many tasks, it is still affected by vanishing gradient. Meanwhile, the final hidden state of traditional RNN is difficult to capture the earlier local information in the case of long sentences. To solve the above problems, we introduce LSTM [36], [37] in earlier local information in the case of long sentences. To translation [26], sentiment classification [38], and automatic formative from many candidates and has been widely used automatically assigns higher values to instances with more information and has been widely used in many natural language processing tasks, such as machine translation [26], sentiment classification [38], and automatic question and answering [27].

There exists a structural representation \( e_D \) for each entity \( e \). To a sentence-level representation \( s_e \) belonging to entity \( e \), the attention factor between \( s_e \) and \( e_D \) is defined as

\[
att(s_e, e_D) = \frac{s_e \cdot e_D}{||s_e|| \cdot ||e_D||}
\] (14)

Reference sentences with higher \( att(s_e, e_D) \) are considered to express \( e \) better. Finally, top m reference sentences are collected. The textual representation \( e_D \) of \( e \) is obtained as shown in Eq. (15).

\[
e_D = \sum_{i=1}^{m} \frac{att(s_{e,i}, e_D) \cdot s_{e,i}}{\sum_{i=1}^{m} att(s_{e,i}, e_D)}
\] (15)

3.4.3 Attention Mechanism of Reference Sentences

After encoding the sentence-level representation from reference sentence using the above method, we will construct the text representation by integrating these sentence-level representations. In this paper, we propose a multi-instance learning algorithm based on attention mechanism to select the top-m most informative ones from many reference sentences to explain the entity explicitly. The attention mechanism automatically assigns higher values to instances with more informative from many candidates and has been widely used in many natural language processing tasks, such as machine translation [26], sentiment classification [38], and automatic question and answering [27].

There exists a structural representation \( e_D \) for each entity \( e \). To a sentence-level representation \( s_e \) belonging to entity \( e \), the attention factor between \( s_e \) and \( e_D \) is defined as

\[
att(s_e, e_D) = \frac{s_e \cdot e_D}{||s_e|| \cdot ||e_D||}
\] (14)

Reference sentences with higher \( att(s_e, e_D) \) are considered to express \( e \) better. Finally, top m reference sentences are collected. The textual representation \( e_D \) of \( e \) is obtained as shown in Eq. (15).

\[
e_D = \sum_{i=1}^{m} \frac{att(s_{e,i}, e_D) \cdot s_{e,i}}{\sum_{i=1}^{m} att(s_{e,i}, e_D)}
\] (15)

3.5 Initialization and Optimization

In this paper, the entities and words that can be aligned will be initialized with the words embedding of \( G \) learned by Skip-Gram [34]. The relations and other entities are randomly initialized. For \( L_{\Theta_h}(G) \), \( L_{\Theta_o}(G, D) \) and \( L_{\Theta_e}(G, D) \), standard back propagation and mini-batch SGD are used in the optimization.

4. Experiments

In this section, we will conduct experiments on several tasks, including entity prediction, relation prediction, and triple classification, to verify the proposed Joint Representations framework for knowledge graphs and textual information via Reference Sentences method (denoted as JRRS).

4.1 Datasets

The widely-used dataset Freebase-FB15K [2] is used to learn the joint representation in our experiments. FB15K consists of 14,951 entities and 1,345 relations. The training set includes 483,142 triples, while validation and test set includes 50,000 and 59,071 triples respectively.

For each entity in FB15K, Wikipedia is used as a corpus to extract reference sentences. Meanwhile, NewYorkTimes corpus aligned with FB15K is chosen as plain text for joint training. The sentences that linked to anchor texts of entities in FB15K are considered to enhance alignment. A total of 876,227 sentences containing both head and tail entity are extracted and are marked with relations in corresponding triples.

4.2 Evaluation Tasks

We compare and analyze the joint representation learning method proposed in this paper with other comparative algorithms on the widely used related to knowledge representation and knowledge reasoning tasks [7], [10], [16], [20], as follows (shown in Fig. 3):

1. entity prediction task
   This task is to predict the vector of the missing entity based on the vectors of an entity and it’s relation in the triples.

2. relation prediction task
   This task is to predict the vector of the missing relation based on the head entity vector and tail entity vector in the triples.

3. triple classification task
   This task is to determine whether the given triple is correct or not.

![Fig. 3](image-url) The overview of the detail of the evaluation task.
4.3 Parameter Settings

In this work, the learning rates of knowledge graph $\epsilon_0 \in \{0.1, 0.01, 0.001\}$. The text-related learning rates $\epsilon_D \in \{0.01, 0.025, 0.05\}$. $\mu$, $\gamma$ and $\eta$ are set to 1. $\lambda$ is 1. Both of $\alpha$ and $\beta$ are set as $\{0.01, 0.001, 0.0001\}$. The dimensions $k$ of all vectors are in $\{50, 100, 150, 200\}$. Multiple comparison experiments are conducted by combining different parameters. The optimal parameter in our experiments is $\epsilon_0 = 0.001, \epsilon_D = 0.025, \alpha = 0.001, \beta = 0.0001$ and $k = 150$.

4.4 Baselines

We use two kinds of other methods in our experiments to verify the proposed method: (i) the translation based methods, including TransE [10], TransH [11], and TransR [12]; (ii) the joint representation learning method of knowledge graph and textual information proposed recently, including Wang [16] and Han [15]. Wherein, Wang [16] associated the knowledge graph and textual information with alignment. However, it uses the local information (i.e., entities) in plain texts merely. Han [15] projected words, entities, and relations into the same semantic space focusing on mining textual relations.

4.5 Results and Analysis

4.5.1 Entity Prediction

For any missing entity in this task, all the comparative methods need to sort the candidate entities without providing the best results. All entities in FB15K are sorted in descending order with score function $||t - h - r||_2$. The head and tail entities of triples in the test set are replaced with these entities. The proportion of correct entities among the top 10 candidate entities (Hits@10) is used as evaluation metrics [10].

Table 1 shows the results. Wherein, † and ‡ indicate the improvements ($p < 0.05$) for TransR and Han’s method, respectively. Generally, relations in knowledge graph can be divided into four categories, one-to-one (1-to-1), one-to-many (1-to-N), many-to-one (N-to-1), and many-to-many (N-to-N). Table 1 shows the average Hits@10 for different kinds of relations, and the overall average Hits@10 for all triples. In addition, “U” and “B” in Table 1 represent two negative sampling strategies, respectively [12].

As shown in Table 1, our method (i.e., JRRS) achieves the best performance on all kinds of relations. The overall Hits@10 is 75.6% higher than TransE, demonstrating the importance of information contained in the text on improving knowledge representation. Meanwhile, it is necessary to jointly model the representation learning of knowledge graph with texts to represent knowledge. Compared to the methods of Wang and Han in overall performance, the proposed JRRS method is 6.7% higher than Wang and 5.1% higher than Han. This is due to the close coupling of learning knowledge graph representation and textual information. It is also proved the necessity of refining entity vector representation through reference sentences. In addition, Wang simply utilizes the words separated from texts while ignoring word order and context. Our method is improved significantly by introducing sentence encoder to model the context. Han also introduced context in [15]. However, [15] is easily affected by the noise of texts. While in our work, reference sentences are introduced to enhance the modeling of textual representation and reduce noise effectively. Detailed comparative analysis of Han and JRRS is presented in Fig. 4. Wherein, “PH” and “PT” represent predicting head entity task and predicting tail entity task, respectively.

4.5.2 Relation Prediction

For any missing relation, all the comparative methods should generate a unique relation as the best one based on $||t - h - r||_2$. Since lacking enough relations, we choose the
4.5.3 Triple Classification

A triple \((h, r, t)\) in the test set is considered as correct if its scoring is blow than the threshold \(\xi\), otherwise wrong. Therefore, this task is generally regarded as two-classification problems. \(\xi\) is the maximum classification accuracy on the validation set. To verify the contribution of attention mechanism to the proposed framework, a variant of the proposed framework, denoted as JRRS-NoATT is proposed by removing attention mechanism. The classification accuracy is used as the evaluation metric in experiments. \(\dagger\) and \(\ddagger\) indicate the improvements \((p < 0.05)\) for TransR and Han, respectively.

As shown in Table 3, our framework (i.e., JRRS and even JRRS-NoATT) performances better than other algorithms, although Wang and Han also introduce textual information into knowledge representation. In addition, the differences between Ours and Ours-NoATT fully demonstrate the importance of attention mechanism on improving the triple classification. In our work, the attention mechanism dynamically selects more effective reference sentences from the candidates to represent entities. Meanwhile, the attention mechanism alleviates the noise caused by low-quality reference sentences significantly.

### Table 2

| Methods    | 1-to-1 | 1-to-N | N-to-1 | N-to-N | Overall |
|------------|--------|--------|--------|--------|---------|
| TransE     | 0.325  | 0.659  | 0.352  | 0.553  | 0.536   |
| TransH(U)  | 0.330  | 0.820  | 0.585  | 0.673  | 0.652   |
| TransH(B)  | 0.330  | 0.879  | 0.556  | 0.756  | 0.732   |
| TransR(U)  | 0.380  | 0.782  | 0.737  | 0.784  | 0.759   |
| TransR(B)  | 0.390  | 0.895  | 0.660  | 0.811  | 0.785   |
| Wang       | 0.403  | 0.881  | 0.858  | 0.932  | 0.902   |
| Han        | 0.409† | 0.894† | 0.871† | 0.946† | 0.916†  |
| JRRS (Ours)| 0.414‡ | 0.921‡ | 0.907‡ | 0.967‡ | 0.952‡  |

### Table 3

| Methods      | Accuracy |
|--------------|----------|
| TransE       | 0.771    |
| TransH(U)    | 0.821    |
| TransH(B)    | 0.827    |
| TransR(U)    | 0.831    |
| TransR(B)    | 0.839    |
| Wang         | 0.842    |
| Han          | 0.886†   |
| JRRS (Ours)  | 0.931‡   |
| JRRS-NoATT (Ours) | 0.898‡   |

### Table 4

| JRRS (Ours) | Han | Wang |
|-------------|-----|------|
| (Colin_Firth, Nationality, ?) | Britain | Britain |
| (Ireland, islands_in_group, ?) | British_Isles | x_Hawaiian_Islands |
| (Jon_Favreau, profession, ?) | Film_director | Film_director |
| (Feroz_Khan, languages, ?) | Hindi | x_English |
| (Nashua, people_born_here, ?) | Mandy_Moore | Mandy_Moore |
| (Scotland, x) | x_English | x_Nepali |

Knowledge Graphs (KGs) are graph-structured knowledge bases, where factual knowledge is represented in the form of relations between entities. KGs have become a crucial resource for many tasks in machine learning, data mining, and artificial intelligence applications including question and answer systems, Web search and many other natural language processing (NLP) and artificial intelligence (AI) tasks. For example, in question and answer systems, knowledge graphs could be used to directly answer complex question. Unfortunately, the application of knowledge graph often suffers severely from data sparsity and low computational efficiency with the growth of knowledge scale. The proposed framework contributes directly to overcoming this crucial problem and completing the current KGs. To provide a direct and precise description for the benefits of our work, Table 4 sketches some examples about the proposed JRRS and comparative baselines (e.g., Han and Wang) in entity prediction task.

From the case study claimed by Table 4, we could conclude that, our JRRS could predict the missing tail entity for some tough input, (Ireland, islands_in_group, ?) and (Feroz_Khan, languages, Hindi) (the 5th line in Table 4) can be updated into the current knowledge graph with help of the proposed joint knowledge representation framework, and then we could answer the question “What language can Feroz Khan speak?”
easily for the downstreaming question and answer task, which is essential for artificial intelligence applications.

5. Conclusion

In this paper, high-quality references sentence are selected for the entities in the knowledge graph through the attention mechanism, which is used to model the representation learning of the textual entity. The knowledge graph and text information based on the denomination of the reference sentence is proposed to represent the learning framework. The framework consists of a knowledge graph representation learning module, a textual relation representation learning module, a textual entity representation learning module, and projects words, entities, and relations to the same semantic vector space. The experimental results on the common evaluation tasks such as entity prediction tasks, relation prediction tasks, and triple classification tasks show that the proposed framework performance is significantly better than other solutions. Especially, compared with the state-of-the-art method [15], the proposed framework improves the metric of H@10 by 5.08% and 3.93% in entity prediction task and relation prediction task, respectively, and improves the metric of accuracy by 5.08% in triple classification task.

References

[1] Q. Wang, Z. Mao, B. Wang, and L. Guo, “Knowledge Graph Embedding: A Survey of Approaches and Applications,” IEEE Transactions on Knowledge and Data Engineering, vol.29, no.12, pp.2724–2743, 2017.
[2] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, “Freebase: a collaboratively created graph database for structuring human knowledge,” Proceedings of the 2008 ACM SIGMOD international conference on Management of data, pp.1247–1250, 2008.
[3] X. Wei, H. Huang, L. Nie, H. Zhang, X.-L. Mao, and T.-S. Chua, “I Know What You Want to Express: Sentence Element Inference by Incorporating External Knowledge Base,” IEEE Transactions on Knowledge and Data Engineering, vol.29, no.2, pp.344–358, 2016.
[4] F.M. Suchanek, G. Kasneci, and G. Weikum, “YAGO: A Core of Semantic Knowledge,” Proceedings of the 16th international conference on World Wide Web, pp.697–706, 2007.
[5] D.-H. Pham and A.-C. Le, “Learning multiple layers of knowledge representation for aspect based sentiment analysis,” Data & Knowledge Engineering, vol.114, pp.26–39, 2018.
[6] R.B. Xie, Z.Y. Liu, and M.S. Sun, “Representation learning of knowledge graphs with hierarchical types,” Proceedings of the 25th International Joint Conference on Artificial Intelligence, pp.2965–2971, 2016.
[7] Y.K. Lin, Z.Y. Liu, and M.S. Sun, “Knowledge representation learning with entities, attributes and relations,” Proceedings of the 25th International Joint Conference on Artificial Intelligence, pp.2866–C2872, 2016.
[8] Q. Xie, X. Ma, Z. Dai, and E. Hovy, “An Interpretable Knowledge Transfer Model for Knowledge Base Completion,” Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pp.950–962, 2017.
[9] R. Jenatton, N.L. Roux, A. Bordes, and G.R. Obozinski, “A latent factor model for highly multi-relational data,” Advances in Neural Information Processing Systems, vol.25, pp.3167–3175, 2012.
[10] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, “Translating embeddings for modeling multi-relational data,” Advances in Neural Information Processing Systems, vol.26, pp.2787–2795, 2013.
[11] Z. Wang, J.W. Zhang, J.L. Feng, and Z. Chen, “Knowledge graph embedding by translating on hyperplanes,” Proceedings of the 28th AAAI Conference on Artificial Intelligence, pp.1112–1119, 2014.
[12] Y.K. Lin, Z.Y. Liu, M.S. Sun, Y. Liu, and X. Zhu, “Learning entity and relation embeddings for knowledge graph completion,” Proceedings of the 29th AAAI Conference on Artificial Intelligence, pp.2181–2187, 2015.
[13] Z.G. Wang and J.Z. Li, “Text-enhanced representation learning for knowledge graph,” Proceedings of the 25th International Joint Conference on Artificial Intelligence, pp.1293–1299, 2016.
[14] L. Huang, J. May, X. Pan, H. Ji, X. Ren, J. Han, L. Zhao, and J.A. Hendler, “Liberal Entity Extraction: Rapid Construction of Fine-Grained Entity Typing Systems,” Big Data, vol.5, no.1, pp.19–31, 2017.
[15] X. Han, Z.Y. Liu, and M.S. Sun, “Joint representation learning of text and knowledge for knowledge graph completion,” CoRR, abs/1611.04125, 2016.
[16] Z. Wang, J.W. Zhang, D.L. Wang, Z. Chen, and M. Li, “Knowledge graph and text jointly embedding,” Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, pp.1591–1601, 2014.
[17] K. Toutanova, D. Chen, P. Pantel, H. Poon, P. Choudhury, and M. Gamon, “Representing Text for Joint Embedding of Text and Knowledge Bases,” Proceedings of The 19th SIGNLL Conference on Computational Natural Language Learning, pp.1499–1509, 2015.
[18] J.L. Shelefbine, “Student factors related to variability in learning word meanings from context,” Journal of Literacy Research, vol.22, no.1, pp.71–97, 1990.
[19] K. Guu, J. Miller, and P. Liang, “Traversing Knowledge Graphs in Vector Space,” Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp.318–327, 2015.
[20] J. Xu, K. Chen, X. Qiu, and X. Huang, “Knowledge graph representation with jointly structural and textual encoding,” Proceedings of the 26th International Joint Conference on Artificial Intelligence, pp.1318–1324, 2017.
[21] S. Guo, Q. Wang, L. Wang, B. Wang, and L. Guo, “Jointly Embedding Knowledge Graphs and Logical Rules,” Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp.192–202, 2016.
[22] B. An, B. Chen, X.P. Han, and L. Sun, “Accurate Text-Enhanced Knowledge Graph Representation Learning,” The 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp.745–755, 2018.
[23] H. Zhong, J. Zhang, Z. Wang, H. Wan, and Z. Chen, “Aligning Knowledge and Text Embeddings by Entity Descriptions,” Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp.267–272, 2015.
[24] I. Yamada, H. Shindo, H. Takeda, and Y. Takefuji, “Joint Learning of the Embedding of Words and Entities for Named Entity Disambiguation,” Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, pp.250–259, 2016.
[25] Y. Cao, L. Huang, H. Ji, X. Chen, and J.Li, “Bridge Text and Knowledge by Learning Multi-Prototype Entity Mention Embedding,” Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, pp.1623–1633, 2017.
[26] D. Bahdana, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” Proceedings of 3rd International Conference on Learning Representations, pp.1–15, 2015.
[27] L. Yang, Q. Ai, J. Guo, and W.B. Croft, “aNMM: Ranking Short Answer Texts with Attention-Based Neural Matching Model,” Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, pp.287–296, 2016.
[28] T. Mikolov, G.S. Corrado, K. Chen, and J. Dean, “Efficient estima-
Zizheng Ji received the B.S. degree in computer science and technology from Beijing Information Science and Technology University. He is pursuing his PhD degree from Beijing Institute of Technology Currently. He is interested in machine learning and knowledge graph.

Zhengchao Lei received the B.S. and PhD degrees in Computer Science from Beijing Institute of Technology in 2011 and 2019, respectively. His research interests include computer vision, saliency detection, and knowledge graph.

Tingting Shen received her Master degree from Beijing Institute of Technology. Her research mainly focuses on data mining and knowledge graph.

Jing Zhang received his B.S. degree from Capital University of Economics and Business and PhD degree from China Agricultural University. He is working in Capital University of Economics and Business. He is interested in artificial intelligence.

Y. Xu, L. Mou, G. Li, Y. Chen, H. Peng, and Z. Jin, “Classifying Relations via Long Short Term Memory Networks along Shortest Dependency Paths,” Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp.1785–1794, 2015.

M. Mintz, S. Bills, R. Snow, and D. Jurafsky, “Distant supervision for relation extraction without labeled data,” Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL, pp.1003–1011, 2009.

D. Sorokin and I. Gurevych, “Context-Aware Representations for Knowledge Base Relation Extraction,” Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp.1785–1790, 2017.

M.G. Xiao and C. Liu, “Semantic relation classification via hierarchical recurrent neural network with attention,” 26th International Conference on Computational Linguistics, pp.1254–1263, 2016.

D.J. Zeng, K. Liu, S.W. Lai, G.Y. Zhou, and J. Zhao, “Relation classification via convolutional deep neural network,” 24th International Conference on Computational Linguistics, pp.2335–2344, 2014.

Q. Le and T. Mikolov, “Distributed representations of sentences and documents,” Proceedings of the 31st International Conference on Machine Learning, pp.1188–1196, 2014.

S. Wang, J. Zhang, and C. Zong, “Learning sentence representation with guidance of human attention” Proceedings of the 26th International Joint Conference on Artificial Intelligence, pp.4137–4143, 2017.

S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Computation, vol.9, no.8, pp.1735–1780, 1997.

F. Karim, S. Majumdar, H. Darabi, and S. Chen, “LSTM Fully Convolutional Networks for Time Series Classification,” IEEE Access, vol.6, pp.1662–1669, 2018.

X. Zhou, X. Wan, and J. Xiao, “Attention-based LSTM Network for Cross-Lingual Sentiment Classification,” Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp.247–C256, 2016.