Neural Knowledge Acquisition via Mutual Attention between Knowledge Graph and Text

Xu Han[1], **Zhiyuan Liu**[1*], **Maosong Sun**[1,2]
[1]Department of Computer Science and Technology,
State Key Lab on Intelligent Technology and Systems,
National Lab for Information Science and Technology, Tsinghua University, Beijing, China
[2]Beijing Advanced Innovation Center for Imaging Technology,
Capital Normal University, Beijing, China

Abstract

We propose a general joint representation learning framework for knowledge acquisition (KA) on two tasks, knowledge graph completion (KGC) and relation extraction (RE) from text. In this framework, we learn representations of knowledge graphs (KGs) and text within a unified parameter sharing semantic space. To achieve better fusion, we propose an effective mutual attention between KGs and text. The reciprocal attention mechanism enables us to highlight important features and perform better KGC and RE. Different from conventional joint models, no complicated linguistic analysis or strict alignments between KGs and text are required to train our models. Experiments on relation extraction and entity link prediction show that models trained under our joint framework are significantly improved in comparison with other baselines. Most existing methods for KGC and RE can be easily integrated into our framework due to its flexible architectures. The source code of this paper can be obtained from `https://github.com/thunlp/JointNRE`.

Introduction

People construct various knowledge graphs (KGs, also known as Knowledge Bases) to organize world knowledge. A typical knowledge graph (KG) is usually a multiple relational directed graph, recorded as a set of relational triples (h, r, t), which indicate relation r between two entities h and t, e.g., (*Mark Twain*, `PlaceOfBirth`, *Florida*). KGs play an important role in many applications such as question answering and web search because of their rich structural information.

Nonetheless, KGs are far from completion. There are two typical approaches to extend KGs, knowledge graph completion (KGC) and relation extraction (RE). KGC aims to enrich KGs with novel facts based on the inherent structure of KGs, including graph-based models (Lao and Cohen 2010), tensor-based models (Socher et al. 2013) and translation models (Bordes et al. 2013). RE aims to extract relational facts from plain text. Many efforts are also devoted to RE, such as kernel-based models (Zelenko, Aone, and Richardella 2003), embedding-based models (Gormley, Yu, and Dredze 2015), and neural models (Socher et al. 2012).

*Corresponding author: Zhiyuan Liu (liuzy@tsinghua.edu.cn)

Mintz et al. (2009) propose a distant supervision method to align text with KGs to generate labeled instances, which is a milestone work for RE and also a pioneering attempt to combine KGs and text. Although this alignment mechanism is simple, it inspires some works to jointly consider KGs and text for KA. Weston et al. (2013) directly sum up two ranking scores of KGs and text for feature fusion. Toutanova et al. (2015) and Xie et al. (2016) use relevant text descriptions to enhance entity or relation embeddings. These models conduct one-to-one alignments to link KGs with corresponding text. However, not all entities and relations in KGs can be aligned well with text.

Methods requiring non-strict data correspondence have also been proposed. Wang et al. (2014a) align entities of KGs and entity mentions in text by sharing their embeddings. Riedel et al. (2013) and Verga and McCallum (2016) adopt probabilistic models of matrix factorization and collaborative filtering to capture correlations between knowledge relations and textual patterns via their co-occurrence entity pairs. These works perform well with soft alignments between KGs and text. However, they either consider only partial text information (just entity mentions or textual relations) or rely on complicated linguistic analysis which may bring inevitable parsing errors. Their mainly designed for surface statistical features frameworks also make themselves hard to generalize well and incorporate complex structural and semantic information.

To address these issues, we propose a general joint representation learning framework. As shown in Figure 1, the framework employs a joint learning mechanism for both KGs and text, which is based on comprehensive alignments with respect to words, entities, and relations instead of partial information. For entities mentioned in text, their embeddings are shared with their corresponding mentions to build entity-level alignments. For relations of KGs and their corresponding textual relations, a transfer mapping matrix is adopted to build relation-level alignments. Moreover, we apply neural networks instead of conventional linguistic analysis to automatically encode sentence semantics, which is a powerful way to model large-scale noisy web text.

In order to further alleviate problems caused by noise in datasets and obtain more discriminative representations, we propose a novel mutual attention mechanism. The attention mechanism allows the models of KGs and text to use their

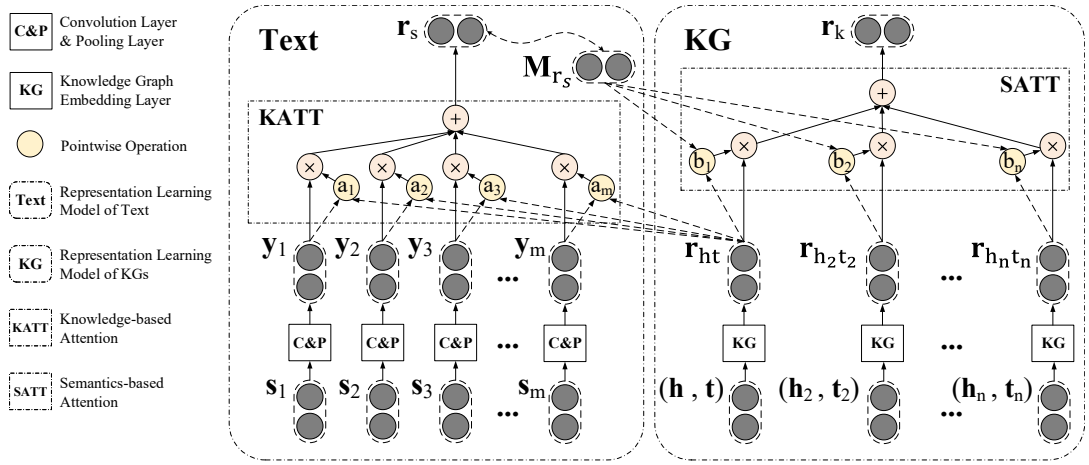

Figure 1: The framework for joint representation learning of KGs and text with the mutual attention.

special information to highlight important features for each other. Noisy data labeled by distant supervision will be distinguished under the guidance of KGs. Meanwhile, textual features also be fed back to select the most important facts for knowledge representation learning. During the training process pushing forward, models learned under mutual guidance between KGs and text are enhanced step by step.

We conduct experiments on real-world datasets whose KGs are extracted from Freebase and text is derived from New York Times (NYT) corpus. We evaluate models on both KGC and RE. Experimental results demonstrate our method effectively perform joint representation learning and obtain more informative knowledge and text representation, which significantly outperforms other baseline methods in KA from either KGs or text. Additionally, experiments also show that our loosely-coupled framework is flexible and most existing embedding-based methods for KGC and RE can be easily integrated into the framework.

Related Work

Our work relates to representation learning of KGs and textual relations, joint learning for KA, and neural networks with attention. We review related works as follows.

Representation Learning of KGs. A variety of approaches have been proposed to encode entities and relations into a continuous low-dimensional space. TransE (Bordes et al. 2013) regards the relation r in the given fact (h, r, t) as a translation from h to t within the low-dimensional space. TransE achieves good results and has many extensions, including TransR (Lin et al. 2015), TransD (Ji et al. 2015), etc. Tensor-based models, such as RESCAL (Nickel, Tresp, and Kriegel 2011), NTN (Socher et al. 2013), DISTMULT (Yang et al. 2015) and HOLE (Nickel et al. 2016), are also effective but trained slowly. In this paper, we incorporate TransE and TransD as representative in our framework to handle representation learning of KGs.

Representation Learning of Textual Relations. Many methods aim to extract relational facts from large-scale text corpora. Mintz et al. (2009) propose distant supervised model. Then Hoffmann et al. (2011) propose a multi-instance mechanism. In recent years, convolutional neural networks (CNN) (Zeng et al. 2014; 2015; 2017), recurrent neural networks (RNN) (Zhang and Wang 2015) and long short-term memory networks (LSTM) (Miwa and Bansal 2016) have been proposed to identify relations between entities in given sentences. These neural models are capable of accurately capturing textual relations without explicit linguistic analysis. In this paper, we apply CNN to embed textual relations due to its time efficiency.

Joint Learning for Knowledge Acquisition. Some works attempt to combine KGs and text for KA. Weston et al. (2013) directly sum up knowledge and text ranking scores. Xie et al. (2016) and Wang and Li (2016) use neural networks to embed text descriptions into KG embedding spaces. Toutanova et al. (2015) extract textual relations using dependency parsing to incorporate text information. These models need well-aligned datasets and cannot be well generalized to most general cases of combining KGs and text.

Wang et al. (2014a) train words and entities together to let them share parameters. Riedel et al. (2013) propose universal schema to transmit information between relations of KGs and textual patterns via their common entity pairs. Verga et al. (2016) further incorporate neural networks to relax constraints imposed by entity pairs in universal schema. These models have no need of strictly aligned datasets but only take partial information into consideration. In this paper, we build a general joint learning framework, which aligns words, entities and relations at the same time.

Neural Networks with Attention. In KA, Lin et al. (2016) and Luo et al. (2017) build a sentence-level attention over multiple instances to reduce weights of noisy instances. Verga and McCallum (2016) use neural networks with attention to merge similar semantic patterns in universal schema. We propose a mutual attention in this paper. Our attention combines models and serves as a channel for information sharing. Moreover, the attention lets models of KGs and text use additional information for mutual model improvements.

Methodology

In this section, we introduce the framework of joint representation learning and the mutual attention, starting with notations and definitions.

Notations and Definitions

We denote KGs as $G = \{E, R, T\}$, where E, R and T indicate sets of entities, relations and facts respectively. Each fact triple $(h, r, t) \in T$ indicates a relation $r \in R$ between $h \in E$ and $t \in E$.

Accompanying with G, we denote the text corpus consisting of sentences as D. The vocabulary of D is denoted as V. Each sentence in D is a sequence with n words $s = \{w_1, \ldots, w_n\}, w_i \in V$. In each sentence, there are two annotated entity mentions along with a textual relation $r_s \in R$ between them.

For each entity, relation and word $h, t \in E$, $r \in R$ and $w \in V$, we use the bold face $\mathbf{h}, \mathbf{t}, \mathbf{r}, \mathbf{w} \in \mathbb{R}^{k_w}$ to indicate their low-dimensional vectors respectively, where k_w is the embedding dimension.

Overall Framework of Joint Learning

In this framework, we aim to jointly learn representations of entities, relations and words within the same continuous space. After denoting all these representations as model parameters $\theta = \{\theta_E, \theta_R, \theta_V\}$, the framework aims to find optimal parameters

$$\hat{\theta} = \arg\max_{\theta} P(G, D|\theta), \tag{1}$$

where $\theta_E, \theta_R, \theta_V$ are parameters for entities, relations and words respectively. $P(G, D|\theta)$ is the conditional probability defined over the knowledge graph G and the text corpus D given the parameters θ. The conditional probability can be further decomposed as

$$P(G, D|\theta) = P(G|\theta_E, \theta_R)P(D|\theta_V). \tag{2}$$

$P(G|\theta_E, \theta_R)$ is used to learn representations of both entities and relations from G, whose formula is to maximize the likelihood of the facts in G. $P(D|\theta_V)$ is used to learn representations of sentence words as well as textual relations from the text corpus D, whose formula is to maximize the likelihood of the sentences and their corresponding textual relations in D.

In summary, we have

$$P(G|\theta_E, \theta_R) = \prod_{(h,r,t)\in G} P((h, r, t)|\theta_E, \theta_R), \tag{3}$$

$$P(D|\theta_V) = \prod_{s\in D} P((s, r_s)|\theta_V), \tag{4}$$

where $P((h, r, t)|\theta_E, \theta_R)$ denotes the conditional probability of relational triples (h, r, t) in the knowledge graph G and $P((s, r_s)|\theta_V)$ denotes the conditional probability of sentences and their corresponding textual relations (s, r_s) in the text corpus D.

Representation Learning of KGs To learn from relational triples of KGs, we will optimize the conditional probability $P(h|(r, t), \theta_E, \theta_R)$, $P(t|(h, r), \theta_E, \theta_R)$, and $P(r|(h, t), \theta_E, \theta_R)$ instead of $P((h, r, t)|\theta_E, \theta_R)$. This decomposition is an empirical approach for convenience of calculations, which has also been used by some previous works (Wang et al. 2014a; Lin, Liu, and Sun 2016).

For each entity pair (h, t) in G, we define its latent relation embedding $\mathbf{r}_{ht}$ as a translation from $\mathbf{h}$ to $\mathbf{t}$, which can be formalized as

$$\mathbf{r}_{ht} = \mathbf{t} - \mathbf{h}. \tag{5}$$

Meanwhile, each triple $(h, r, t) \in T$ has an explicit relation r between h and t. Hence, we can define the scoring function for each triple as follows,

$$f_r(h, t) = b - \|\mathbf{r}_{ht} - \mathbf{r}\|, \tag{6}$$

where b is a bias constant. Based on the above scoring function, the conditional probability can be formalized over all triples in T as follows,

$$P(r|(h, t), \theta_E, \theta_R) = \frac{\exp(f_r(h, t))}{\sum_{r'\in R} \exp(f_{r'}(h, t))}. \tag{7}$$

$P(h|(r, t), \theta_E, \theta_R)$ and $P(t|(h, r), \theta_E, \theta_R)$ are defined in the same way. In fact, this representation objective is consistent with TransE (Bordes et al. 2013), and thus we name this model Prob-TransE.

We also adopt TransD (Ji et al. 2015), which is an extension of TransE, to encode relational triples,

$$\begin{aligned} &\mathbf{r}_{ht} = \mathbf{t}_r - \mathbf{h}_r, \\ &\mathbf{h}_r = \mathbf{M}_{rh}\mathbf{h}, \quad \mathbf{t}_r = \mathbf{M}_{rt}\mathbf{t}, \\ &\mathbf{M}_{rh} = \mathbf{r}_p\mathbf{h}_p^\top + \mathbf{I}^{k_r\times k_w}, \\ &\mathbf{M}_{rt} = \mathbf{r}_p\mathbf{t}_p^\top + \mathbf{I}^{k_r\times k_w}. \end{aligned} \tag{8}$$

We name this model Prob-TransD. Entities and relations are in different spaces in Prob-TransD. $\mathbf{r}_p \in \mathbb{R}^{k_r}$ and $\mathbf{h}_p, \mathbf{t}_p \in \mathbb{R}^{k_w}$ are projection vectors. $\mathbf{M}_{rh}, \mathbf{M}_{rt}$ are used to map entity embeddings into relation spaces. In order to simplify expressions, k_r and k_w are the same in our framework.

Representation Learning of Textual Relations Given a sentence containing two entities, the sentence usually exposes implicit features of the textual relation between the two entities. We apply CNN for textual relation representation learning.

For each word in a given sentence s containing (h, t) with a textual relation r_s, we concatenate its word embedding $\mathbf{w}_i \in \mathbb{R}^{k_w}$ (Mikolov et al. 2013) and position embedding $\mathbf{p}_i \in \mathbb{R}^{k_p\times 2}$ (Zeng et al. 2014) to build its input embedding $\mathbf{x}_i \in \mathbb{R}^{k_i} (k_i = k_w + k_p \times 2)$,

$$\begin{aligned} \mathbf{s} &= \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \\ &= \{[\mathbf{w}_1; \mathbf{p}_1], \ldots, [\mathbf{w}_n; \mathbf{p}_n]\}, \end{aligned} \tag{9}$$

where k_w and k_p are the dimensions of word embedding and position embedding respectively.

In the convolution layer, we slide a window of size m over the input sequence $\mathbf{s}$. For each move, we can get a hidden

layer vector as

$$\hat{\mathbf{x}}_i = \left[\mathbf{x}_{i-\frac{m-1}{2}}; \ldots; \mathbf{x}_i; \ldots; \mathbf{x}_{i+\frac{m-1}{2}}\right], \quad (10)$$
$$\mathbf{h}_i = \tanh(\mathbf{W}\hat{\mathbf{x}}_i + \mathbf{b}),$$

where $\mathbf{W} \in \mathbb{R}^{k_c \times mk_i}$ is the convolution kernel, $\mathbf{b} \in \mathbb{R}^{k_c}$ is a bias vector, k_c is the dimension of hidden layer vectors.

In the pooling layer, a max-pooling operation over the hidden layer vectors $\mathbf{h}_1, \ldots, \mathbf{h}_n$ is applied to get the final output embedding $\mathbf{y}$ as follows,

$$[\mathbf{y}]_j = \max\{[\mathbf{h}_1]_j, \ldots, [\mathbf{h}_n]_j\}, \quad (11)$$

where $[\mathbf{y}]_j$ and $[\mathbf{h}_i]_j$ are the j-th value of the output embedding $\mathbf{y}$ and the hidden vector $\mathbf{h}_i$ respectively. Our method will further get the scoring function,

$$\mathbf{o} = \mathbf{M}\mathbf{y}, \quad (12)$$

where $\mathbf{M} \in \mathbb{R}^{\|R\| \times k_c}$ is the representation matrix to calculate the relation scores. Then we define the conditional probability $P((s, r_s)|\theta_V)$ as follows,

M用来判断这句话里面包含的是哪个关系

$$P((s, r_s)|\theta_V) = \frac{\exp(\mathbf{o}_{r_s})}{\sum_{r \in R} \exp(\mathbf{o}_r)}. \quad (13)$$

Mutual Attention between KGs and Text

Our mutual attention consists of two parts, the knowledge-based attention for text model guidance and the semantics-based attention for knowledge model guidance. Both parts cooperate with each other during the training.

Knowledge-based Attention For each $(h, r_s, t) \in T$, there may be several sentences $\pi_{r_s} = \{s_1, \ldots, s_m\}$ containing (h, t) and implying the relation r_s, where m is the total number of sentences containing (h, t). These sentences' output embeddings are $\{\mathbf{y}_1, \ldots, \mathbf{y}_m\}$. Sentences labeled by the distant supervision algorithm contain some vague and wrong semantic components. Hence, we argue that some sentences contribute more to the final textual relation representation.

Additional logical knowledge information can be used to enhance sentence embedding under the joint learning framework. We use the latent relation embedding $\mathbf{r}_{ht} \in \mathbb{R}^{k_w}$ as the knowledge-based attention over sentences to highlight important sentences and reduce noisy components:

$$\mathbf{e}_j = \tanh(\mathbf{W}_s\mathbf{y}_j + \mathbf{b}_s), \quad (14)$$
$$a_j = \frac{\exp(\mathbf{r}_{ht} \cdot \mathbf{e}_j)}{\sum_{k=1}^{m} \exp(\mathbf{r}_{ht} \cdot \mathbf{e}_k)},$$
$$\mathbf{r}_s = \sum_{j=1}^{m} a_j \mathbf{y}_j,$$

where $\mathbf{W}_s \in \mathbb{R}^{k_w \times k_c}$ is the weight matrix and $\mathbf{b}_s \in \mathbb{R}^{k_w}$ is the bias vector. a_j is the weight for the jth sentence output $\mathbf{y}_j$. We take a weighted sum of sentence output embeddings for the global textual relation representation $\mathbf{r}_s$. Then, we formalize $P((\pi_{r_s}, r_s)|\theta_V)$ instead of $\prod_{j=1}^{m} P(s_j, r_s|\theta_V)$ as follows,

$$\mathbf{o} = \mathbf{M}\mathbf{r}_s, \quad (15)$$
$$P((\pi_{r_s}, r_s)|\theta_V) = \frac{\exp(\mathbf{o}_{r_s})}{\sum_{r \in R} \exp(\mathbf{o}_r)}.$$

Semantics-based Attention For each relation $r \in R$, there are several entity pairs $\psi_r = \{(h_1, t_1), \ldots, (h_n, t_n)\}$ that can form fact triples in T with the relation r. The latent relation embeddings of these pairs are $\{\mathbf{r}_{h_1t_1}, \ldots, \mathbf{r}_{h_nt_n}\}$, where n is the number of entity pairs. In knowledge graph representation models, we hope that all latent relation embeddings between entity pairs are close to explicit relation embeddings.

Because of complicated related situations between entities and errors from initial construction of KGs, it is difficult to match explicit relations with all latent relations during the training process. In order to make knowledge graph representation models more effective, we attempt to use semantic information extracted from text models to help explicit relations fit most reasonable entity pairs as follows,

$$\mathbf{e}_r = \tanh(\mathbf{W}_s\mathbf{M}_r + \mathbf{b}_s), \quad (16)$$
$$b_j = \frac{\exp(\mathbf{e}_r \cdot \mathbf{r}_{h_jt_j})}{\sum_{k=1}^{n} \exp(\mathbf{e}_r \cdot \mathbf{r}_{h_kt_k})},$$
$$\mathbf{r}_k = \sum_{j=1}^{n} b_j \mathbf{r}_{h_jt_j},$$

where $\mathbf{W}_s$ and $\mathbf{b}_s$ are the same weight matrix and bias vector used in Eq. (14) to map neural vectors to the entity and relation space. $\mathbf{M}_r$ is the text representation embedding for the relation r used in Eq. (12), which contains textual relation semantics. b_j is the weight for the jth latent relation embedding $\mathbf{r}_{h_jt_j}$.

We merge these entity pairs and formalize the conditional probability $P(r|\psi_r, \theta_E, \theta_R)$ instead of the original formalization $\prod_{j=1}^{n} P(r|(h_j, t_j), \theta_E, \theta_R)$ as follows,

$$f_r(\psi_r) = b - \|\mathbf{r}_k - \mathbf{r}\|, \quad (17)$$
$$P(r|\psi_r, \theta_E, \theta_R) = \frac{\exp(f_r(\psi_r))}{\sum_{r' \in R} \exp(f_{r'}(\psi_r))}.$$

Experiments

Initialization and Implementation Details

KG-Text Alignments Since entities and relations are not explicitly labeled in text, we have to identify entities and relations in text to support joint representation learning. The process is realized by the following entity-text alignments and relation-text alignments.

Entity-Text Alignments. Many entities are mentioned in text. Due to complex polysemy of entity mentions (e.g., an entity name Washington in a sentence could indicate either a person or a location), it is non-trivial to build entity-text alignments. In this paper, we simply use anchor text annotated in articles to build alignments between entities in E and entity mentions in V.

Relation-Text Alignments. Inspired by the idea of distant supervision (Min et al. 2013), for a relation $r \in R$, we collect all entity pairs $Pair_r = \{(h, t)|(h, r, t) \in T\}$ connected by r. Afterwards, for each entity pair in $Pair_r$, we extract all sentences from D containing both entities, and regard them as the positive instances of the relation r.

Optimization Details Here we introduce the learning and optimization details for our joint models. We define the optimization function as the log-likelihood of the objective function in Eq. (2),

$$\begin{aligned}\mathcal{L}_\theta(G, D) &= \log P(G, D|\theta) + \lambda\|\theta\|_2 \\ &= \log P(G|\theta_E, \theta_R) + \log P(D|\theta_V) \\ &+ \lambda\|\theta\|_2\end{aligned} \tag{18}$$

where λ is a harmonic factor, and $\|\theta\|_2$ is the regularizer defined as L_2 distance. All models are optimized simultaneously using stochastic gradient descent (SGD). The word embeddings used for CNN are pre-trained from plain text by Skip-Gram (Mikolov et al. 2013). In practice, we will optimize knowledge and text models in parallel.

Datasets and Experiment Settings

Datasets The datasets used for experiments contain two parts, knowledge graphs and text corpus, whose details are as follows.

Knowledge Graph. We select Freebase (Bollacker et al. 2008) as the KG for joint learning. Freebase is a widely-used large-scale KG. In this paper, we adopt datasets extracted from Freebase, FB15K and FB60K in our experiments. FB15K has been used as the benchmark for KGC. FB60K is extended from the dataset developed by (Riedel, Yao, and McCallum 2010), which has been used as the benchmark for RE. We list the statistics of FB15K and FB60K in Table 1, including the number of entities, relations, and facts.

Dataset	Relation	Entity	Fact
FB15K	1,345	14,951	592,213
FB60K	1,324	69,512	335,350

Table 1: The statistics of FB15K and FB60K.

Text Corpus. We select sentences from the articles of New York Times. We extract $194,385$ sentences containing both head and tail entities in FB15K and annotate with the corresponding relations in triples. The sentences are labeled with $47,103$ FB15K triples, including 699 relations and 6053 entities. We name the corpus NYT-FB15K. The sentences for FB60K come from the dataset used in (Riedel, Yao, and McCallum 2010), containing $570,088$ sentences, $63,696$ entities, 56 relations and $293,175$ facts. We name the corpus NYT-FB60K.

Following the previous usage of these datasets, FB15K and NYT-FB15K are used as the benchmark for KGC, FB60K and NYT-FB60K are used as the benchmark for RE in our experiments.

Parameter Settings In our joint models, we select the learning rate α_k for $P(G|\theta_E, \theta_R)$ among $\{0.1, 0.01, 0.001\}$, and learning rate α_t for $P(D|\theta_V)$ among $\{0.1, 0.01, 0.001\}$. The sliding window size m is among $\{3, 5, 7\}$. For other parameters, since they have limited effect on results, we simply follow the settings used in (Zeng et al. 2014; Lin et al. 2016) so that we can fairly compare joint learning results with these baselines. To compare with previous works, the dimension k_w is 50 for RE and 100 for KGC. Table 2 show all parameters used in our experiments.

Harmonic Factor λ	0.0001
Knowledge Learning Rate α_k	0.001
Text Learning Rate α_t	0.01
Hidden Layer Dimension k_c	230
Word/Entity/Relation Dimension k_w	50
Position Dimension k_p	5
Window Size m	3
Dropout Probability p	0.5

Table 2: Parameter settings.

Relation Extraction

Most distant supervision models automatically annotate sentences in text corpora as training instances and then extract textual features to build relation classifiers. We want to investigate the effectiveness of our joint framework with respect to improving CNN models via this task.

Evaluation Results We follow Weston et al. (2013) to conduct our evaluation. The evaluation constructs candidate triples by combining entity pairs in the test set with various relations, and rank these triples according to their corresponding sentence representations. By regarding the triples in the KGs as correct and others as incorrect, we evaluate different methods with their precision-recall curves.

The evaluation results on NYT-FB60K test set are shown in Figure 2, where "JointD+KATT" and "JointE+KATT" indicate the CNN model with knowledge-based attention learned jointly with Prob-TransD and Prob-TransE respectively. "CNN+ONE" indicates the CNN model with the at-least-one mechanism (Zeng et al. 2015). "CNN+ATT" indicates the CNN model with sentence-level attention (Lin et al. 2016), which is the state-of-the-art method for RE. We also compare these neural models with feature-based methods, including Mintz (Mintz et al. 2009), MultiR (Hoffmann et al. 2011), MIML (Surdeanu et al. 2012) and Sm2r (Weston et al. 2013). The results are also shown in Figure 2. From the results, we observe that:

(1) As compared with feature-based methods in Figure 2, the joint models significantly outperform all these methods over the entire range of recall. The joint models preserve stable and competitive precision when the recall is smaller than 0.15. The joint models also increase by 10% to 20% when the recall is larger than 0.15.

(2) Besides JointD+KATT and JointE+KATT, CNN+ATT and CNN+ONE also have more than 10% increase when the recall is larger than 0.15. All these demonstrate that deep neural models which are not restricted to the feature engineering are robust and effective.

(3) Though the results of the feature-based methods drop much more faster, they still have reasonable precision among the recommendations with the highest scores. It shows that human-designed features are very limited in some fields but still effective. In the future, it is a very meaningful attempt to add these features to our joint learning framework as extra guidance.

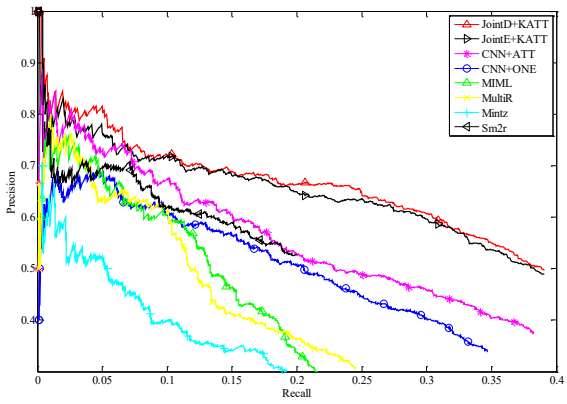

Figure 2: Aggregate precision/recall curves of different RE models.

P@N(%)	100			300		
Method	ONE	ATT	KATT	ONE	ATT	KATT
CNN+	67.3	**76.2**	-	58.1	59.8	-
JointE+	67.5	74.1	75.8	63.0	63.2	68.0
JointD+	**68.5**	74.6	**80.6**	**67.0**	**67.3**	**68.7**
P@N(%)	500			Mean		
Method	ONE	ATT	KATT	ONE	ATT	KATT
CNN+	43.7	48.5	-	56.4	61.5	-
JointE+	57.3	59.3	63.0	62.6	65.5	68.9
JointD+	**58.6**	**61.1**	**63.7**	**64.8**	**67.7**	**71.0**

Table 3: Evaluation results on P@N with different model combination (%).

Effect of Joint Learning and Attention We usually pay more attention to recommendations with the highest confidence scores in RE. To compare the results in detail with respect to independent and joint learning models, we empirically compare different models via their prediction accuracy over recommendations with the highest confidence. We select the CNN model used in Zeng et al. (2014) as our sentence encoders. These encoders are combined with different kinds of multi-instance learning methods, including the at-least-one mechanism (ONE), sentence-level attention (ATT), and our knowledge-based attention (KATT). "JointD" and "JointE" indicate the CNN models learned jointly with Prob-TransD and Prob-TransE respectively, and "CNN" indicates the CNN model learned independently. The results are shown in Table 3, including P@100, P@300, P@500 and the mean of them. From the results, we observe that:

(1) All the sentence encoders combined with different multi-instance learning methods get significant improvements after being trained under our joint learning framework. From the average results of the prediction accuracy, CNN+ONE increases by 6% and CNN+ATT increases by 5% after joint learning.

(2) As compared with the sentence encoders learned jointly with Prob-TransE, the encoders learned jointly with Prob-TransD are further enhanced. Prob-TransD is more complex than Prob-TransE, which can better extract knowledge features and comprehend relationships between entities. The results demonstrate that the joint learning framework successfully takes advantages of KGs to train the sentence encoders, and the representation ability of KG models can affect the final results.

(3) In Table 3, ATT and KATT perform much better than ONE. The training sentences constructed via distant supervision contain noise, and not all sentences contain entity pairs can exactly indicate textual relations. Hence, the attention mechanism is beneficial and effectively highlights the most meaningful sentences.

(4) The comparison between ATT and KATT further shows that the simple attention mechanism without using information in KGs is not enough. The same relation often has nuances when it is between different entity pairs, and a vague global attention cannot select important sentences accurately. Hence, information from KGs helps knowledge-based attention be more discriminative than sentence-level attention. This indicates the effectiveness of our knowledge-based attention.

Knowledge Graph Completion

Entity link prediction has been used for KGC evaluation in Bordes et al. (2013). We need to predict the tail entity when given a triple $(h, r, ?)$ or predict the head entity when given a triple $(?, r, t)$. We want to investigate the effectiveness of our joint model with respect to improving KG models via this task.

Evaluation Results For each test triple (h, r, t), we replace the head and tail entities with all entities in FB15K ranked in descending order of distance scores calculated by Eq. (6). The relational fact (h, r, t) is expected to have a better score than any other corrupted triples. We follow previous works and use the proportion of correct entities in top-10 ranked entities (Hits@10) as the evaluation metric.

The relations in KGs can be divided into four classes: 1-to-1, 1-to-N, N-to-1 and N-to-N relations (Bordes et al. 2013). We report the average Hits@10 scores when predicting missing head entities and tail entities with respect to different classes of relations. We also report the overall performance by averaging the Hits@10 scores over triples.

Since the evaluation setting is identical, we simply report the results of SE, SME, TransE, TransH, TransR/CTransR, TransD (Bordes et al. 2011; 2012; 2013; Wang et al. 2014b; Lin et al. 2015; Ji et al. 2015). The models for knowledge representation without joint learning in our framework are named "Prob-TransE" and "Prob-TransD". The models learned under our joint learning framework with our semantics-based attention are named "JointE+SATT" and "JointD+SATT". The results are shown in Table 4. From the results, we observe that:

(1) The joint models achieve improvements under four classes of relations when predicting head and tail entities. This indicates models trained under our joint framework take advantages of plain text and significantly improve

Method	Predicting Head				Predicting Tail				Overall
	1-to-1	1-to-N	N-to-1	N-to-N	1-to-1	1-to-N	N-to-1	N-to-N	Triple Avg.
SE (Bordes et al. 2011)	35.6	62.6	17.2	37.5	34.9	14.6	68.3	41.3	39.8
SME (Bordes et al. 2012)	35.1	69.6	19.9	40.3	32.7	14.9	76.0	43.3	41.3
TransE (Bordes et al. 2013)	43.7	65.7	18.2	47.2	43.7	19.7	66.7	50.0	47.1
TransH (Wang et al. 2014b)	66.8	87.6	30.2	64.5	65.5	39.8	83.3	67.2	64.4
TransR (Lin et al. 2015)	78.8	89.2	38.1	66.9	79.2	38.4	90.4	72.1	68.7
TransD (Ji et al. 2015)	81.2	94.8	47.1	79.3	81.6	53.9	93.7	82.5	78.9
Prob-TransE	66.5	88.8	39.8	79.0	66.4	51.9	85.6	81.5	76.6
JointE+SATT	82.7	**96.2**	45.0	80.7	81.7	57.7	93.6	84.0	79.3
Prob-TransD	79.1	93.0	42.2	79.2	79.2	51.6	90.9	82.7	78.2
JointD+SATT	**82.7**	95.2	**47.8**	**81.6**	**82.0**	**57.9**	**94.7**	**84.7**	**80.4**

Table 4: Evaluation results on link prediction of head and tail entities (%).

Dataset	Method	Hits@10
FB15K	DKRL(Xie et al. 2016)	67.4
	TEKE(Wang and Li 2016)	73.0
	DESP(Zhong et al. 2015)	77.3
	JointE+SATT	79.3
	JointD+SATT	**80.4**
FB15K-237	E+DISTMULT (Toutanova et al. 2015)	60.2
	E+DISTMULT(CONV) (Toutanova et al. 2015)	61.1
	JointE+SATT	69.2
	JointD+SATT	**69.9**

Table 5: Evaluation results on link prediction of different joint learning models (%).

knowledge graph representations in relation-level.

(2) The improvements on "1-to-1", "1-to-N" and "N-to-1" relations are much more significant as compared to those on "N-to-N". This indicates that our joint framework is more effective to embed textual relations for those deterministic relations.

(3) TransD is a model extended from TransE and has a complicated entity embedding mechanism. After integrated into joint learning framework, it further improves its performance. It means that other knowledge graph representation methods similar to TransE and TransD, such as TransH and TransR, can also be integrated into our framework via the same way.

Comparison with Other Joint Learning Models We also compare our models with other joint learning models for KGC. DESP (Zhong et al. 2015), TEKE (Wang and Li 2016), and DKRL (Xie et al. 2016) learn entity embeddings from KGs and text descriptions. E+DISTMULT(CONV) (Toutanova et al. 2015) extracts textual relations using dependency parsing to incorporate text into DISTMULT. Following the previous experiment settings, we compare our joint models with DESP, TEKE, and DKRL in FB15K. We also use FB15K-237 aligned with our NYT corpus to train our joint models for comparison with E+DISTMULT(CONV). The Evaluation results are shown in Table 5. From the results, we observe that our joint models which directly encode from sentences outperform methods based on dependency parsing. Though we train our joint models with non-strictly aligned text corpus, our models are still significantly more effective, even as compared with methods using strictly aligned text descriptions.

Conclusion and Future Work

In this paper, we propose a general joint framework for representation learning of KGs and text. Our framework embeds entities, relations, and words within a unified space. More specifically, the framework work well with non-strictly aligned data. We also propose the mutual attention between KGs and text, which is made up of the knowledge-based attention and the semantics-based attention. These two parts enhance joint models during the training process. On both RE and KGC, experiment results show that the joint learning framework effectively performs representation learning for both KGs and text. By incorporating different knowledge representation learning models, we also show the framework is open to existing models. In the future, we will explore to adopt RNN or LSTM for encoding textual relations in an efficient manner. To take more rich information especially some effective human-designed features as the guidance for our joint framework, such as relation paths in KGs, is also necessary.

Acknowledgments

This work is supported by the 973 Program (No. 2014CB340501), the National Natural Science Foundation of China (NSFC No. 61572273, 61532010), China Association for Science and Technology (2016QNRC001), and Tsinghua University Initiative Scientific Research Program (20151080406).

References

Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of KDD*.

Bordes, A.; Weston, J.; Collobert, R.; Bengio, Y.; et al. 2011.

Learning structured embeddings of knowledge bases. In *Proceedings of AAAI*.

Bordes, A.; Glorot, X.; Weston, J.; and Bengio, Y. 2012. Joint learning of words and meaning representations for open-text semantic parsing. In *Proceedings of AISTATS*.

Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of NIPS*.

Gormley, M. R.; Yu, M.; and Dredze, M. 2015. Improved relation extraction with feature-rich compositional embedding models. In *Proceedings of EMNLP*.

Hoffmann, R.; Zhang, C.; Ling, X.; Zettlemoyer, L.; and Weld, D. S. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of ACL-HLT*.

Ji, G.; He, S.; Xu, L.; Liu, K.; and Zhao, J. 2015. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of ACL*.

Lao, N., and Cohen, W. W. 2010. Relational retrieval using a combination of path-constrained random walks. *Proceedings of Machine learning*.

Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; and Zhu, X. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of AAAI*.

Lin, Y.; Shen, S.; Liu, Z.; Luan, H.; and Sun, M. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of ACL*.

Lin, Y.; Liu, Z.; and Sun, M. 2016. Knowledge representation learning with entities, attributes and relations. In *Proceedings of IJCAI*.

Luo, B.; Feng, Y.; Wang, Z.; Zhu, Z.; Huang, S.; Yan, R.; and Zhao, D. 2017. Learning with noise: Enhance distantly supervised relation extraction with dynamic transition matrix. In *Proceedings of ACL*.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *Proceedings of ICLR*.

Min, B.; Grishman, R.; Wan, L.; Wang, C.; and Gondek, D. 2013. Distant supervision for relation extraction with an incomplete knowledge base. In *Proceedings of HLT-NAACL*.

Mintz, M.; Bills, S.; Snow, R.; and Jurafsky, D. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of ACL-IJCNLP*.

Miwa, M., and Bansal, M. 2016. End-to-end relation extraction using lstms on sequences and tree structures. *Proceedings of ACL*.

Nickel, M.; Rosasco, L.; Poggio, T. A.; et al. 2016. Holographic embeddings of knowledge graphs. In *Proceedings of AAAI*.

Nickel, M.; Tresp, V.; and Kriegel, H.-P. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of ICML*.

Riedel, S.; Yao, L.; McCallum, A.; and Marlin, B. M. 2013. Relation extraction with matrix factorization and universal schemas. In *HLT-NAACL*.

Riedel, S.; Yao, L.; and McCallum, A. 2010. Modeling relations and their mentions without labeled text. In *Proceedings of ECMLPKDD*.

Socher, R.; Huval, B.; Manning, C. D.; and Ng, A. Y. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of EMNLP-CoNLL*.

Socher, R.; Chen, D.; Manning, C. D.; and Ng, A. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Proceedings of NIPS*.

Surdeanu, M.; Tibshirani, J.; Nallapati, R.; and Manning, C. D. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of EMNLP*.

Toutanova, K.; Chen, D.; Pantel, P.; Poon, H.; Choudhury, P.; and Gamon, M. 2015. Representing text for joint embedding of text and knowledge bases. In *Proceedings of EMNLP*.

Verga, P., and McCallum, A. 2016. Row-less universal schema. In *Proceedings of ACL*.

Verga, P.; Belanger, D.; Strubell, E.; Roth, B.; and McCallum, A. 2016. Multilingual relation extraction using compositional universal schema. In *Proceedings of NAACL*.

Wang, Z., and Li, J.-Z. 2016. Text-enhanced representation learning for knowledge graph. In *Proceedings of IJCAI*.

Wang, Z.; Zhang, J.; Feng, J.; and Chen, Z. 2014a. Knowledge graph and text jointly embedding. In *Proceedings of EMNLP*.

Wang, Z.; Zhang, J.; Feng, J.; and Chen, Z. 2014b. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of AAAI*.

Weston, J.; Bordes, A.; Yakhnenko, O.; and Usunier, N. 2013. Connecting language and knowledge bases with embedding models for relation extraction. In *Proceedings of EMNLP*.

Xie, R.; Liu, Z.; Jia, J.; Luan, H.; and Sun, M. 2016. Representation learning of knowledge graphs with entity descriptions. In *Proceedings of AAAI*.

Yang, B.; Yih, W.-t.; He, X.; Gao, J.; and Deng, L. 2015. Embedding entities and relations for learning and inference in knowledge bases. *Proceedings of ICLR*.

Zelenko, D.; Aone, C.; and Richardella, A. 2003. Kernel methods for relation extraction. *Proceedings of JMLR*.

Zeng, D.; Liu, K.; Lai, S.; Zhou, G.; Zhao, J.; et al. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING*.

Zeng, D.; Liu, K.; Chen, Y.; and Zhao, J. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of EMNLP*.

Zeng, W.; Lin, Y.; Liu, Z.; and Sun, M. 2017. Incorporating relation paths in neural relation extraction. In *Proceedings of EMNLP*.

Zhang, D., and Wang, D. 2015. Relation classification via recurrent neural network. *arXiv preprint arXiv:1508.01006*.

Zhong, H.; Zhang, J.; Wang, Z.; Wan, H.; and Chen, Z. 2015. Aligning knowledge and text embeddings by entity descriptions. In *Proceedings of EMNLP*.