Neurocomputing 410 (2020) 419–431

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

CTEA: Context and Topic Enhanced Entity Alignment for knowledge graphs

Zhihuan Yan, Rong Peng *, Yaqian Wang, Weidong Li

School of Computer Science, Wuhan University, Wuhan, China

ARTICLE INFO

Article history:
Received 12 December 2019
Revised 9 March 2020
Accepted 14 June 2020
Available online 19 June 2020
Communicated by Steven Hoi

Keywords:
Knowledge graph embedding
Entity alignment
Entity contexts
Topic model

ABSTRACT

We study the problem of finding entities referring to the same real world object in multilingual knowledge graphs(KGs), i.e., entity alignment for multilingual KGs. Recently, embedding-based entity alignment methods get extended attention in this area. Most of them firstly embed the entities in low dimensional vectors space via relation structure of entities, and then align entities via these learned embeddings combined with some entity similarity function. Even achieved promising performances, these methods are defective in utilizing entity contexts and entity topic information. In this paper, we propose a novel entity alignment framework CTEA (**C**ontext and **T**opic Enhanced **E**ntity **A**lignment), which integrates entity context information and entity topic information to help alignment. This framework learns entity topic distributions from their attributes with a specially designed topic model BTM4EA, and the learned entity topic distributions are used to filter some weakly correlated entities for each entity to be aligned. Meanwhile, we embed KGs to low dimensional vectors space via translation-based KG embedding model and mine context information from these vectors with an attention attached Convolutional Neural Network(CNN). The entity embeddings, entity contexts and entity topics are combined to get the final alignment results. Extended experiments reveal that our method achieves promising performances in most cases.

1. Introduction

KGs are directed graphs which organize abundant human knowledge as structural data. They often serve intelligent systems such as search, recommendation and question answer systems as knowledge base. Typically a KG encodes structured information of millions of entities and billions of facts, and these facts are represented as two types of triples: relation triples and attribute triples. Relation triples such as (*AlanTuring*, *Education*, *PrincetonUniversity*) indicates that head entity *AlanTuring* and tail entity *PrincetonUniversity* has relation *Education*. Attribute triples hold the attributes of one entity, e.g., (*AlanTuring*, *birthdate*, 23/6/1912).

In recent years, the internet has enabled the creation of a growing number of large scale multilingual KGs such as DBpedia [1], YAGO [2] and BabelNet [3]. While single language knowledge in these multilingual KGs may be created separately from different data sources by various stakeholders or for different intentions, which lead to entities referring to the same real world object may appear in different languages in various forms. For example, entity *http://zh.dbpedia.org /resource/林青霞* and entity *http://dbpedia.org/resource/Brigitte_Lin* in DBpedia both refer to an actress in China, they contain complementary contents about this actress but appear in different sub-KGs If they are indicated as the same entity, we can obtain more comprehensive knowledge about this actress. Unfortunately, not all the equivalences have been exploited in cross-lingual KGs. Therefore, an increasing number of researchers throw themselves to working on the problem of cross-lingual entity alignment, aiming to match entities in different languages automatically. Traditional cross-lingual entity alignment approaches rely on machine translation technique or artificially designed features to discover cross-lingual equivalent. While machine translation itself is a sophisticated problem to be studied, and artificially designed features not only time-consuming and painstaking but also subjective. Recently, along with the evolvement of knowledge graph embedding models, a number of embedding-based entity alignment methods (for example, MTransE [4], JAPE [5], BootEA [6]) have been proposed for this task. Given two KGs and a set of seed entity pairs, these methods firstly project entities into low-dimensional vector space by encoding on monolingual knowledge, and then with the help of these

* Corresponding author.
E-mail addresses: zhihuanyan@whu.edu.cn (Z. Yan), rongpeng@whu.edu.cn (R. Peng), yaqian@whu.edu.cn (Y. Wang), weidonghappy@whu.edu.cn (W. Li).

https://doi.org/10.1016/j.neucom.2020.06.054
0925-2312/

entity vectors a similarity score between entities is computed via a learnt similarity score function. Even achieved promising performance, these methods are defective in the following two limitations:

Firstly, current methods make no discrimination among all target entities when finding true target entity for an source entity. We deem that filtering the weakly related entities via some entity high-level semantic before precisely aligning entities would help to improve alignment performance, especially for long-tail entities and entities which are far from the pre-aligned entities. For example, as Fig. 1 shows, the source 厦门市 and the target entity *Xiamen* have no aligned neighbors. In this case, current embedding-based entity alignment methods, which only take advantage of entity embeddings when aligning entities, have little evidence supporting equivalence between these two entities. As the aligning information provided by the pre-aligned entities will loss a lot when it propagate to the two entities. Thus the embeddings for these two entities will be fairly dissimilar, and *Xiamen* will be ranked behind when seeking target entity for 厦门市 via entity similarity. While the source entity and the target entity may have some similar high-level features (such as topic information). If we filter some poorly related entities using these high-level features, the rank of the target entity will rank ahead of many other entities.

Secondly, the current embedding-based entity alignment methods do not explicitly model the contexts of entities. Current methods like MTransE [4], IEAKE [7], and BootEA [6] embed the structure of entities via transE, which models each triplet separately. For an entity, even the information of its neighbors can be transmitted to the center entity in iteration process, they can only be implicitly captured. GCN-based entity alignment models such as GCN-Align [8] and MuGCN [9] attempt to integrate entity neighbor information to entity embedding, but they fail to consider relation semantics, which actually play an important role in entity alignment. The embeddings of entities current method learnt can only reflect the semantic of entities themselves. While similar entities always have similar context, entity context should be explicitly considered when aligning entities.

Motivated by the aforementioned observations, we propose a three-steps entity alignment framework. The first step is generating a candidate set for each source entity with the help of some high-level features of entities, the second step is generating entity embedding and entity context representations using entity structural features, and the last step is obtaining alignments based on the above results.

To filter weakly related entities, we propose to extract high-level feature from entity attributes. As all attribute names of an entity are unordered and topic models are a cluster of bag-of-word models which have good performance in text clustering. Therefore, we employ topic model to learn entity high-level semantic. The number of attributes of an entity usually not very large (16 attributes per entity on average in one of the datasets using in this paper), so we extend BTM [10], a friendly topic model to short text, to learn the topic distribution for all entities. The Jensen-Shannon(JS) divergence is used to evaluate the distance between topic distribution of two entities, and the entities ranking ahead according to this distance are clustered as the candidates.

To generate entity embeddings, we have diverse choices. Translation-based model like TransE [11], TransR [12], TransD [13] learn entity embeddings using simple operations and limited parameters,translational models are faster, require fewer parameters and are relatively easier to train; CNN-based models like ConvE [14] and ConvKB [15] learn more expressive embeddings due to their parameter efficiency and consideration of complex relations. Here, we use TransE as the embedding model as its simplest and embedding model is not the focus of this paper.

To obtain entity context representations, we aggregate entity neighbor information with attention mechanism. Graph neural networks based models such as R-GCN [16] and attention-based model [17] may be available choices. R-GCN [16] is an extension of applying graph convolutional networks GCN [18] to relational data. It gathers neighbor information of each entity and assigns them equal weights. Its aggregation mode is $h_i^{l+1} = \sigma\left(\sum_{r\in R}\sum_{j\in N_i^r}\frac{1}{c_{i,r}}W_r^{(l)}h_j^{(l)} + W_0^{(l)}h_i^{(l)}\right)$. Attention-based model [17] assigns different weight for entity neighborhood. Its aggregation mode is $h_i^{l+1} = \sigma\left(\sum_{j\in N_i}\sum_{k\in R_{ij}}\alpha_{ijk}W[h_i\|h_j\|g_k]\right)$. We could see that information transformation is performed by matrix multiplication in both R-GCN [16] and attention-based model [17]. As we all know, the expression ability of CNN is stronger than linear model, and convolution operator is parameter efficient. Therefore, we apply convolution operator to neighbors of each entity and aggregate the convolution results with attention mechanism to obtain entity context representation.

Once obtaining candidate entity set, embedding, context representation for each entity, the alignment results are gained then. Entity embeddings are used to generate entity embedding similarity matrix, each element in which represents similarity between two entity embeddings. Entity context representations are used to generate entity context similarity matrix, each element in which represents similarity between two entity context representations. The two similarity matrixes are combined with some weight to obtain the final similarity matrix. We rank the elements in each row in descending order, and put the entities in candidate be front of others. And the rank of true target entity for each entity will be obtained. The main contribution of this paper are summarized as follows:

- We propose a novel entity alignment framework CTEA which integrates entity embeddings, entity contexts and entity topics to align entities (Section 4).
- For modelling the topic information of entities, we propose BTM4EA, an entity alignment oriented topic model to learn topic from the attributes of entities (Section 4.3).

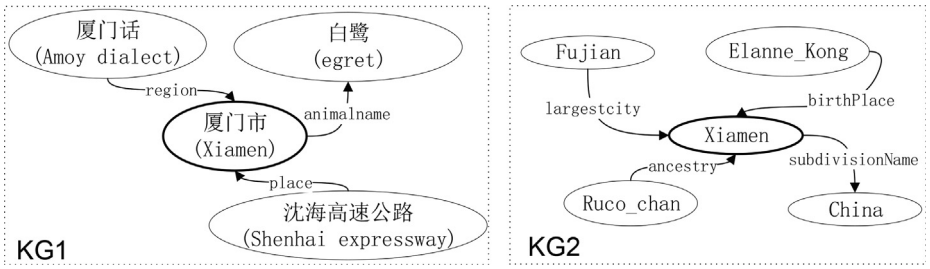

Fig. 1. Illustration of the structural heterogeneity of entities.

- We propose to utilize the context representations of entities to help alignment and design a CNN model to learn context representations from entity contexts (Section 4.4).
- The experiments on three real-world datasets show that CTEA outperforms or on par with existing embedding-based entity alignment methods. Each module of CTEA contributes to the improvements (Section 5).

The rest of this paper is organised as follows. In Section 2, some related works about KG embedding models and embedding-based entity alignment methods are introduced. Section 3 expounds on the formalizations and the problem description. Section 4 elaborates on CTEA. We evaluate our method on three benchmark datasets in Section 5. And the concluding remarks and future works are provided in Section 6.

2. Related works

2.1. KG embedding

Embedding KG into low dimensional space has drawn much attention in recent years. Recently emerged KG embedding models can be classified as two kinds: (1) translation-based, (2) neural network-based. The most famous and sample ones are a cluster of translation-based models, and TransE [11] is the initial one. For a triple (h, r, t) which means head entity h and tail entity t has relation r, TransE expects $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$, where $\mathbf{r}$, $\mathbf{h}$ and $\mathbf{t}$ denote the embedding of h, r and t, respectively.

TransE is sample and has good performance in linking predication task for 1-to-1 relations. But when it comes to 1-to-n, n-to-1, and n-to-n relations, it will make a mistake as it learning single representation for each item(i.e.,entity and relation), which will lead to identical representation for different entities. To solve this problem, TransH [19] introduces relation-related projection vectors and projects entities to relation-related hyperplanes and then executes translation on each particular hyperplanes. TransR [12] assigns a transfer matrix for each relation r to map entity vector to different relation spaces and performs the translation in the relation space. TransD[13] extends TransR by assigning two vectors for each item in a triple, one for embedding and one for projection. TranSparse [20] replaces transfer matrices in TransR with adaptive spare matrices considering different relations link different numbers and different types of entity pairs (heterogeneity), and the number of head entities and tail entities in a specific relation could be also different (imbalance). On account of the entities has various role when present to different triples, TransAt [21] pays attention to different dimension of entity vectors to adapt different relations to ensure the transformations only focus on particular dimensions. In addition, DistMult [22] uses weighted element-wise dot products to model entity relations and ComplEx [23] generalizes DisMult by leveraging complex embeddings and Hermitian dot products instead. This type of models require fewer parameters and relatively easier to train, thereby, they are favored by many KG-based applications.

Considering the less expressive of translation-based models, several neural network models which exploit deep learning techniques for KG embedding are proposed. ProjE [24] uses multilayer perceptron and ConvE [14] uses 2-D convolution over embeddings for linking oredication. And there are two graph neural network based models: R-GCN [16] and attention-based model [17], which gather neighbor information of an entity to enhance entity representation.

Though complex models have relatively better performance, here we leverage translation model as our embedding model because its conciseness.

2.2. Embedding-based entity alignment

Most existing KG embedding models focus on modeling a single KG. However, as applications that use KGs present themselves with more diversity, a single KG can hardly provide all the knowledge needed. An effective way to solve this problem is by integrating heterogeneous knowledge among different KGs via entity alignment. Current embedding-based entity alignment models can be classified into two kinds: (1) translation-based entity alignment models, (2) graph neural network(GNN)-based entity alignment models.

JE [25] leverages TransE to embed different KGs into a unified space with the goal that entities in each pre-aligned pair have similar embeddings. MTransE [4] incorporates TransE to encode KG structures into language-specific vector spaces and designs three different techniques to score the alignments. Most of existing approaches are challenged by the lack of enough prior alignment, to solve this problem, IEAKE [7] and BootEA [6] are two self-learning models which embed two KGs in a unified low dimension vectors space via parameter sharing and parameter swapping firstly, and iteratively label new entity pairs as supervision data. These models only depend on relation triples in KGs but ignore other literal features. JAPE [5,26] employ entity attributes when learn entity embeddings. To utilize the literal descriptions of entities, KDCoE [27] co-trains a multilingual KG embedding model and a multilingual literal description embedding model to take advantage of literal descriptions of entities. AttrE [26] conducts character-level literal embeddings for entity attributes. MultiKE [28,29] integrates multi views of entities to learn embeddings for entity alignment. These models exploit supplementary resource to enhance entity representations, they model high-level semantic of entities in a sense but ignore entity structural contexts.

GCN-Align [8] uses GCNs to embed entities from different languages into a unified vector space. It combines the structure embedding and attribute embedding to get accurate alignments. MuGNN [9] and KEGC [30] jointly complete KG and align entities via cross-KG attention. However, they did not explicitly model relation semantic. VR-GCN [31] integrates the strength of GCN and the translational property in knowledge graphs in order to model relation information.

To represent entities with their contextual information, GraphMatching [32] introduces topic entity graph and formulated entity alignment task as graph matching problem. It leverage not only the representations of entities and entity contexts but global information to find alignments. But it ignored the relation labels in relation triples when model entity contexts.

Instead of making a difference in alignment model, CEAFF [33] proposes a collective embedding-based EA framework with adaptive feature fusion mechanism. It generates structural similarity, semantic similarity and string similarity matrices firstly, then these similarity matrices are combined with an off-line fusion strategy. The combination mode of entity embeddings and entity context representations is also off-line in our method as it will be stated later.

In consideration of expensiveness for accessing pre-aligned entity pairs, work [34] aligns entities under adversarial learning frameworks in fully unsupervised or weakly-supervised fashion.

As a departure from prior works, we divide entity alignment as three steps process: candidates generation, entity representations learning (include entity embeddings and entity context representations), and entity alignment. As we will show later in this paper, doing so exclude many entities which are not globally similar with an source entity when searching its true target entity.

3. Formalization and problem statement

A knowledge graph can be formalized as $KG = \{E, R, A, RT, AT\}$, where E is the entity set, R is the relation set, A is the attribute set, $RT = \{(h, r, t) \in H \times R \times T\}$ is the relation triple set with $H \subseteq$ E is the head entity set and $T \subseteq E$ is the tail entity set, $AT = \{(e, a, v) \in E \times A \times V\}$ is the attribute triple set with v is the value of attribute a. In triple (h, r, t), the embeddings of head entity h, relation r and tail entity t are represented by boldface **h, r, t** respectively; score function of triple (h, r, t) is represented by $f_r(h, t)$. Without loss of generality, we only research on the case of aligning entities between two KGs and formalize the entity alignment scenario in this paper as follows: Let $KG_1 = \{E_1, R_1, A_1, RT_1, AT_1\}$, $KG_2 = \{E_2, R_2, A_2, RT_2, AT_2\}$ denote the source KG and the target KG to be aligned. $Seeds = Sour_{train} \times Tar_{train}$ denotes the pre-aligned entity pairs set and meanwhile part of the training set, where $Sour_{train} \subseteq E_1$ is called **training source entities** and $Tar_{train} \subseteq E_2$ is called **training target entities**. An element $e_s \in Sour_{train}$ is called an **training source entity**, and its counterpart $e_t \in Tar_{train}$ is called its **true target entity**. Entity alignment is to find a map $m : Sour_{train} \rightarrow Tar_{train}$ and align more entities based on this map. The map m is either injection, surjection or bijection in different kinds of entity alignment tasks and we focus on the bijection case in this paper, that is to say, we assume there are one-to-one alignments between **testing source entities** $Sour_{test}$ and **testing target entities** Tar_{test}.

4. Context and Topic Enhanced Entity Alignment

In this section, we first introduce the framework of CTEA, then we elaborate on the technical details of the method and discuss several design issues.

4.1. Overview

The framework of CTEA is depicted in Fig. 2, it contains three modules: Entity Topic Learning (TL) module, Structure Embedding (SE) module, and Context Modelling (CM) module. In the training phase, given two KGs, named by KG_1 and KG_2 in different languages, the two sets of attribute triples AT_1 and AT_2 are firstly used to construct virtual documents (attribute sets in fact), and these virtual documents will be transported to the TL module to train the topic model, and the topic model will output the topic distribution of each entity. On the other hand, the two sets of relation triples RT_1 and RT_2 are input to the SE Module, where the two entities in each entity seed are initialized as the same embedding. The entity representations and relation representations produced by the SE Module, superadd the pre-aligned entities pairs, are input to the CM module and used to optimize the parameters in CM.

For a testing source entity e_s, to obtain its true target entity, the topic distribution distances between it and all target entities are calculated firstly, and then candidate entities for it are generated according to these distances. On the other hand, the embedding for it is used to calculate embedding similarity between it and all target entities' embeddings; the learned context representation for it is used to calculate context similarity between it and all target entities' context. The two similarities will be combined to find its true target entity with the help of the generated candidates.

4.2. Topic learning for entities from attributes

Current embedding-based entity alignment models make no different among all target entities when finding the true target entity for a source entity. However, when a source entity has few neighbors or is far from pre-aligned entities, entity embeddings are not sufficient to accurately reflect what it is. Thus incorrect target entities will be recommended for it via calculating similarities between its embedding and other entities' embeddings. We hope to employ some high-level features to filter weakly related entities in order to increase the possibility the true target entity be recommended.

Entity attributes are inherent features which are very important for aligning entities, we employ entity attributes to model entity high-level semantic. As all attribute names of an entity can be regarded as a bag-of-word and topic models are a cluster of bag-of-word models which have good performance in text clustering. Therefore, we employ topic model to learn entity high-level semantic. As the number of attributes for a single entity is not very

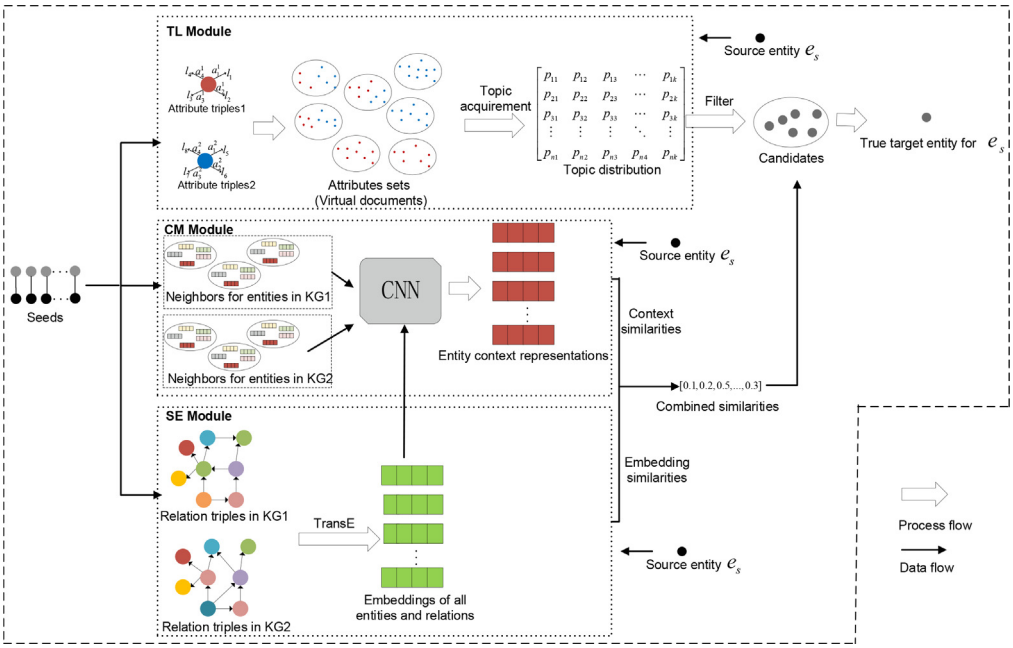

Fig. 2. The framework of CTEA.

large, we improve BTM [10], a friendly topic model for short text which does not emphasize the order of words, to model the topics of entities, which is called BTM4EA.

One of the necessary problem when utilizing topic model is how to define a document. Here we apply all attributes of two entities in an pre-aligned entity pair to generate a virtual document. Technically, we only take advantage of attribute names and their range types but overlook the specific attribute values on account of their complexity. We define attribute set of entity e as $A_e = \{a|(e,\ a,\ v) \in AT\}$. In order for the two entities in a pre-aligned entity pair to have similar topic distributions, the attributes of these two entities are collected into the same virtual document. That is to say, the whole corpus for training topic model is constructed as follows:

$$Corpus = \{d = (A_{e_1}, A_{e_2})|(e_1, e_2) \in Seeds\},$$

where A_{e_1} and A_{e_2} are the attributes set of entity e_1 and entity e_2, respectively. Note that one can let $e_1 = e_2$ when infer topic information for an newly arrived entity. An attribute biterm denotes an unordered attribute-pair correlated with each other. We call two attributes are correlated if they are commonly used together to describe two aligned entities. All attribute biterms B_{attr} are obtained in the following way:

$$B_{attr} = \bigcup_{(A_{e_1}, A_{e_2}) \in Corpus} \{(a_i, a_j)|a_i \in A_{e_1}, a_j \in A_{e_2}\}.$$

Suppose α and β are the Dirichlet priors for topic distribution of the whole corpus and attribute distribution of each topic. The specific generative process for B_{attr} in BTM4EA can be described as follows:

- Set a topic number
- Draw topic distribution $\theta \sim Dir(\alpha)$ for the whole attributes biterms set B_{attr}
- Draw a topic-specific attribute distribution $\phi_z \sim Dir(\beta)$ for each topic
- For each attribute biterm b, assign a topic $z \sim Multi(\theta)$ and select two attribute $a_i, a_j \sim Multi(\phi_z)$

where $Multi(\cdot)$ means a multinomial distribution. Following the above process, the generative probability of the whole corpus can be written as:

$$\begin{aligned} p(B) &= \prod_{(i,j)} p(b = (a_i, a_j)) \\ &= \prod_{(i,j)} \sum_z p(z)p(a_i|z)p(a_j|z) \\ &= \prod_{(i,j)} \sum_z \theta_z \phi_{i|z} \phi_{j|z}. \end{aligned}$$

Once obtaining the topic distribution θ of the whole corpus and the attribute distribution of each topic ϕ_z, the topic of each attribute biterm and further the topic distribution of one entity will be obtained (will be described below). Here we use Gibbs sampling to infer θ and ϕ_z just like BTM does. Gibbs sampling is a simple and widely used Markov chain Monte Carlo algorithm. It is more accurate and more memory-efficient than other inference methods like variational inference and maximum posterior estimation.

The process of Gibbs sampling can guarantee the attributes which usually co-occur tending to share the same topics. Otherwise, we hope attributes with the same type could be more possibly share the same topic. Here attribute type denotes the range type of its value. We distinguish four kinds of attribute types in this paper: *Integer, Double, Datetime*, and *String*. So when sample topic for an specific attribute biterm $b = (a_i, a_j)$, we increase the weight of the topics assigned to the attributes which have the same type with it. The conditional distribution is calculated as follows:

$$P(z|\mathbf{z}_{-b}) \propto p(z_{a_i a_j})(n_z + \alpha)\frac{(n_{a_i|z} + \beta)(n_{a_j|z} + \beta)}{\left(\sum_a n_{a|z} + M\beta\right)^2},$$

where

$$p(z_{a_i a_j}) = \frac{2\ *\ n^+_{z_{a_i a_j}} + n^-_{z_{a_i a_j}}}{\sum_z 2\ *\ n^+_{z_{a_i a_j}} + n^-_{z_{a_i a_j}}}$$

is the weight attached to z according to the types of a_i and a_j. $n^+_{z_{a_i a_j}}$ is defined as:

$$n^+_{z_{a_i a_j}} = card(S_z \cap (Attr_{a_i} \cup Attr_{a_j})),$$

where S_z is the set of attributes assigned as topic z, $Attr_{a_i}$ is the set of attributes having the same type with a_i, $card(\#)$ is the cardinal number of a set, and $Attr_{a_j}$ is the set of attributes having the same type with a_j. $n^-_{z_{a_i a_j}}$ is defined as:

$$n^-_{z_{a_i a_j}} = card(S_z \cap \neq g(Attr_{a_i} \cup Attr_{a_j})),$$

where S_z, $Attr_{a_i}$ and $Attr_{a_j}$ are the same as described above, and $\neq g(\cdot)$ is the complementary of some set. The Gibbs sampling process is shown in Algorithm 1.

Algorithm 1. Gibbs sampling algorithm for BTM4Attr

Input:
hyper-parameter α, β;
the number of topics Z;
biterm set B_{attr};
Output:
topic distribution θ;
attribute distribution of each topic ϕ_z;
1: Initialize topic assignment randomly for each biterm
2: **for** $n_{iter} = 1$ to N_{iter} **do**
3: **for** each $b = (a_i, a_j)$ in B_{attr} **do**
4: obtain $n^+_{z_{a_i a_j}}$ and $n^-_{z_{a_i a_j}}$;
5: calculate $p(z_{a_i a_j})$;
6: draw topic z_b from $p(z|z_{-b}, B_{attr}, \alpha, \beta)$;
7: update $n_{z_b}, n_{a_i|z_b}, n_{a_j|z_b}$;
8: **end for**
9: **end for**
10: compute ϕ_z and θ

With Gibbs sampling process, we can get the attribute-topic distribution ϕ and topic distribution θ of the whole corpus:

$$\phi_{a_i|z} = \frac{n_{a_i|z} + \beta}{\sum_{a_k} n_{a_k|z} + |A|\beta}$$
$$\theta_z = \frac{n_z + \alpha}{B_{attr} + Z\alpha},$$

where A is the set of all attributes. As we model the generation process of the whole corpus and do not model the generation process of a single document, thus we can not directly obtain the topic proportions of an attribute set of an entity and thereby can not derive the topic distribution of an entity directly. To infer the topics of an entity, we assume that the topic proportions of an attribute set equals to the expectation of the topic proportions of biterms contained in the attribute set, which is calculated as following:

$$P(z|d) = \sum_b P(z|b)P(b|d),$$

where

$$P(z|b) = \frac{p(z)p(b|z)}{p(m)} = \frac{p(z)p(a_i)p(a_j)}{\sum_z p(z)p(a_i)p(a_j)}.$$

In BTM, it is assumed that topics of each word in a document have equal importance, but we believe that the more important a biterm is in the document, the more important its topic is in inferring the topic distribution of the document. So different from the way of BTM in calculating the topic distribution of a document which totally calculate a uniform distribution on all biterms, we take the importance of biterms in a document into account. We use the tf_idf of one biterm to measure the importance of the biterm:

$$P(b|d) = \frac{tf\_idf_b^d}{\sum_b tf\_idf_b^d},$$

where tf_idf_b is the tf_idf value of biterm b in d, which represents the importance of b in d. And it is calculated as follows:

$$tf\_idf_b^d = \frac{n_b^d}{|B_d|} * log\frac{|D|}{|D_b|},$$

where $|B_d|$ is the number of biterms in d, $|D|$ is the number of all virtual documents, and $|D_B|$ is the number of virtual documents containing b. After the convergence of Gibbs sampling, we apply Jensen-Shannon divergence to calculate the distance between the distributions of two entities. Assuming topic distribution of source entity e_s is $\vec{p}_{e_s}$, topic distribution of target entity e_t is $\vec{p}_{e_t}$, the topic distance between e_s and e_t is calculated as follows:

$$d(e_s, e_t) = JS(\vec{p}_{e_s}||\vec{p}_{e_t}) = \frac{1}{2}KL\left(\vec{p}_{e_s}||\frac{\vec{p}_{e_s}+\vec{p}_{e_t}}{2}\right) + \frac{1}{2}KL\left(\vec{p}_{e_t}||\frac{\vec{p}_{e_s}+\vec{p}_{e_t}}{2}\right),$$

where KL is the Kullback–Leibler divergence between two distributions, which is defined as:

$$KL(\vec{p}_{e_s}, \vec{p}_{e_t}) = \sum_{i=1}^{N}\vec{p}_{e_s}(i)\left(log\frac{\vec{p}_{e_s}(i)}{\vec{p}_{e_t}(i)}\right),$$

where N is the number of topics.

4.3. Structure embedding

Given a knowledge graph KG, the aim of structure embedding is to model its geometric structures and learn representations for each entity and each relation in it. To simplify the embedding process, we use translation-based embedding model. We model both entities and relations as vectors and consider the relation as the transformation from the head entity to the tail entity. Given a relation triple $tr = (h,\ r,\ t)$, we expect $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$. We use the following score function to measure the plausibility of triple $tr = (h,\ r,\ t)$:

$$f_r(\mathbf{h}, \mathbf{t}) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|,$$

where $\|\cdot\|$ denotes L_2 norm. We prefer low scores for triples in KG, and high scores for negative triples sampled by replacing the head entities or tail entities of positive triples with random entities. As BootEA [6] states, for entity alignment task, absolutely low scores for positive triples help to reduce the drift of embeddings in the unified space and to better capture the common semantics of different KGs, we also apply this strategy here. As we deem that the semantics of entities with abundant neighbors can be more accurately captured, we add fake triples for zero incoming or zero outgoing entities as [35] does. At last the loss function of structure embedding is:

$$L_{SE} = \sum_{(h,r,t)\in Tr^+\cup Tr^+_{fake}}[f_r(\mathbf{h},\mathbf{t}) - \gamma_1]_+ + \sum_{(h,r,t)\in Tr^-\cup Tr^-_{fake}}[\gamma_2 - f_r(\mathbf{h},\mathbf{t})]_+,$$

where Tr^+ is the set of all positive triples in KG, Tr^+_{fake} are the fake triples we construct, and Tr^- and Tr^-_{fake} are the sets of all negative triples sampled by replacing head or tail entities of triples in Tr_+ and Tr^+_{fake}, respectively. $[\cdot]_+ = max(0,\cdot)$, γ_1, γ_2 are two hyper-parameters to limit the scores of positive triples and negative triples. We let two entities in a pre-aligned entity pairs share the same embedding to bridge the gap between two KGs.

4.4. Context modelling

As is described above, entity embedding obtained in the Structure Embedding module only reflects the semantic of one entity itself. Though information of the neighbors can be transformed to the center entity during training process, entity embedding can not explicitly contain the semantics of its neighbors. As we know, aligned entities tend to have similar contexts. So we recur to the contexts of entities to help aligning. Entity context of an entity is defined as follows:

Context of an entity: For relation triples $RT = \{(h,r,t) \in H \times R \times T\}$ and entity e, set $Con_e = \{(h,r)|(h,r,t) \in G \wedge t == e\}$ is called the context of entity e, and each element in Con_e is called neighbor of e, which is denoted as nei_e.

We propose a multi-channel CNN employing attention mechanism to model the context of an entity. Fig. 3 illustrates the architecture of the network.

This network contains four layers: an input layer, a convolution layer, an attention layer and an output layer.

Input Layer Given a source entity $e_s \in Sour_{train}$, the input layer collects all its neighbors into a context set and output the representations of its neighbors $\{\mathbf{nei}^i_{e_s}|i = 1,2,\ldots,card(Con_{e_s})\}$, where $card(\cdot)$ is the cardinal number of a set, $\mathbf{nei}^i_{e_s}$ is the raw representation of $nei^i_{e_s}$, which is defined as follows:

$$\mathbf{nei}^i_{e_s} = \mathbf{h}_i + \mathbf{r}_i.$$

It is worthy to note that the neighbors of an entity are unordered, so the i above only shows the number of neighbors but does not contain any order information.

Convolution Layer The convolution layer is used to extract features from the neighbor representations. To extract features from different aspects, we use K convolution kernels with different size, and each convolution kernel has Ω output channels. For neighbor $nei^i_{e_s}$ of source entity e_s, the output of channel ω of convolution kernel k is

$$\boldsymbol{v}_{e_s,k,\omega,i} = ReLU(\mathbf{W}^\omega_k * \mathbf{nei}^i_{e_s} + \boldsymbol{b}^\omega_k),$$

where $*$ is a convolution operator, $\boldsymbol{b}^\omega_k$ is a bias for $\boldsymbol{W}^\omega_k$ and $ReLU$ is an activation function.The outputs of kernel k in Convolution Layer o^o_k is the aggregation of the outputs for all neighbors from all channels in this kernel:

$$o^o_k = \{\boldsymbol{v}_{e_s,k,\omega,i}|\omega = 1,2,3,\ldots,\ \Omega, i = 1,2,3,\ldots,card(Con_{e_s})\}.$$

As we do not distinguish which neighbor the elements in o^o_k come from, we rewrite o^o_k as follows for convenience:

$$o^o_k = \{\boldsymbol{v}_{e_s,k,n}|n = \Omega \times card(Con_{e_s})\}.$$

Note that these outputs of each kernel are unordered as the same reason described above.

Attention Layer The attention layer is used to filter important information from features extracted by the convolution layer. In

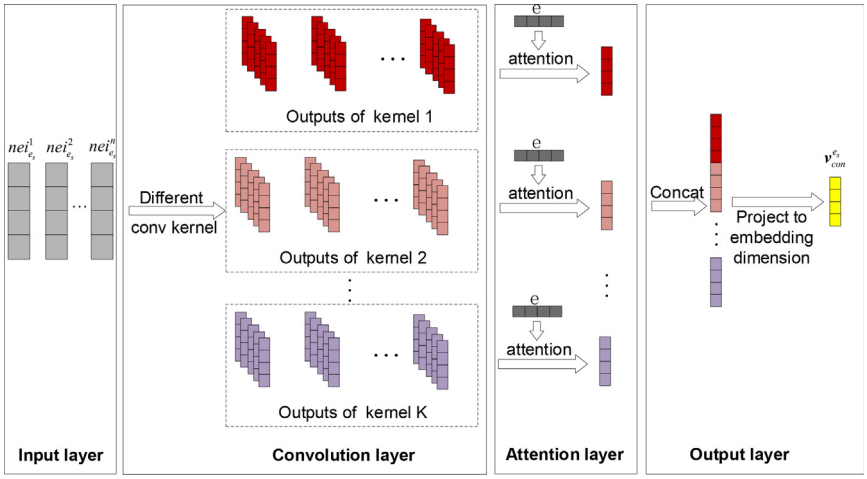

Fig. 3. The process of generating context representation for source entity e_s. $\mathbf{nei}^1_{e_s}, \mathbf{nei}^2_{e_s}, \ldots, \mathbf{nei}^i_{e_s}$ are neighbor representations of e_s, and the final output $\mathbf{v}_{e_s}$ is the context representation for e_s.

the attention layer, the feature vectors generated from each convolution kernel are transformed to a vector with the same dimension as entity embedding. The attention mechanism is employed to each kernel. The feature vectors that are more similar to the source entity embedding are more important for the source entity, and the attention mechanism can enhance their importance. The weight of feature vector $\boldsymbol{v}_{e_s,k,n}$ is computed by:

$$a_n = \frac{exp(-\|\boldsymbol{v}_{e_s,k,n} - \mathbf{e}_s\|)}{\sum_{i=1}^{\Omega \times card(Con_{e_s})} exp(-\|\boldsymbol{v}_{e_s,k,n} - \mathbf{e}_s\|)}.$$

Therefore, the final output of the attention layer O^A is generated by:

$$O^A = Concat(o^A_1, o^A_2, \ldots, o^A_K),$$

where o^A_k is the weighted sum of the feature vectors in o^k_o:

$$o^A_k = \sum_{n=1}^{\Omega \times card(Con_{e_s})} a_n \boldsymbol{v}_{e_s,k,n}.$$

Output Layer The output layer uses the output of the attention layer O^A as the input and outputs a context representation $\mathbf{v}^{e_s}_{con}$ with the same dimension as $\mathbf{e}_s$. We do not use any activation and dropout on this layer. The output of this layer is

$$\mathbf{v}^{e_s}_{con} = \mathbf{M}_o O_A + \mathbf{b}_o,$$

where $\mathbf{M}_o \in \mathbb{R}^{dim \times (dim * K)}$ is the weight matrix of the output layer, $\mathbf{b}_o$ is the bias for $\mathbf{M}_o$ This vector represents the holistic semantics of the context of entity e_s. We hope the context representations of similar entities to be similar. To this end, we minimize the distance between the context representations for the source entity and target entity in each entity pairs in *Seeds*. The loss function of CM module is:

$$Loss_{CM} = \sum_{(e_s,e_t)\in Seeds} \|\mathbf{v}^{e_s}_{con} - \mathbf{v}^{e_t}_{con}|.$$

We train SE and CM jointly with the whole loss function:

$$Loss = Loss_{SE} + Loss_{CM}.$$

4.5. Entity alignment based on the three modules

.

Algorithm 2. Entity alignment based on the three modules

Input:
Matric $topicDis_{Sour_{test}}$; // topic distribution of all test source entities
Matric $topicDis_{Tar_{test}}$; // topic distribution of all test target entities
Matric $embed_{Sour_{test}}$; // embeddings of all test source entities
Matric $embed_{Tar_{test}}$; // embeddings of all test target entities
Matric $con_{Sour_{test}}$; // context representations of all test source entities
Matric $con_{Tar_{test}}$; // context representations of all test target entities
Output: entity alignment results
1: Calculate entity embedding similarities using $embed_{Sour_{test}}$ and $embed_{Tar_{test}}$ and obtain embedding similarity matrix Sim_{embed}
2: Calculate entity context representation similarities using $con_{Sour_{test}}$ and $con_{Tar_{test}}$ and obtain context similarity matrix Sim_{con}
3: Calculate the weighted sum of matrix Sim_{embed} and matrix Sim_{con} as Sim;
4: **for** $i = 1$ to $Card(Sour_{test})$ **do**
5: Calculate JS divergences between e_{s_i} and all entity in Tar_{test} and save them in $d_{topic}(e_{s_i})$;
6: Gather target entities whose JS divergences rank in top τ as candidates $can_{e_{s_i}}$;
7: Sort all target entities in descending order according to the ith row of Sim and let the entities in $can_{e_{s_i}}$ be front of others;
8: Obtain the rank of true target for the ith source entity.
9: **end for**

We use TL module to choose a candidate set for each test source entity and use this candidate set to help find its true target. To be specific, given an test source entity e_s, we calculate the JS divergence between it and each entity in the test target set and sort

these JS divergences in ascending order, then we choose the top τ (τ is an empirical parameter) entities as the candidates, which help the combination of SE and CM to look for its true target entity.

Entity embeddings obtained from SE are used to generate entity embedding similarity matrix Sim_{embed}, each element in which represents similarity between two entity embeddings. Entity context representations obtained from CM are used to generate entity context similarity matrix Sim_{con}, each element in which represents similarity between two entity context representations. The two similarity matrices are combined with some weight to obtain the final similarity matrix Sim:

$$Sim = \mu Sim_{embed} + (1-\mu)Sim_{con}.$$

For each testing source entity, we rank the elements in each row in descending order, and let the entities in candidate be front of others. And the rank of true target entity for each entity will be obtained.

As an embedding-based entity alignment method, CTEA does not extremely emphasize on the precision of correct alignment but pursue lower ranks for all test source entities. We take into account of both the combined representations and topic distribution for source entities when ranking target entities. The detail process is described in Algorithm 2.

5. Experiment

CTEA has been verified on three open datasets. We use Tensorflow to develop our approch. The experiments are conducted on a personal workstation with an Intel Core i7-8700 K 3.70 GHz CPU, a NVIDIA Geforce GTX 1080 Ti GPU and 64 GB memory. Our source code, datasets and experimental results will be available online.

5.1. Datasets and baseline

We use DBP15K as the dataset in our experiments, which are constructed by Ref. [5]. DBP15k contains three cross-lingual datasets built from the multilingual versions of DBpedia: DBP_{zh-en}(Chinese to English), DBP_{ja-en}(Japanese to English), DBP_{fr-en} (French to English). Each dataset contains 15 thousand aligned entity pairs. Like most works using these data sets, we use 30% of these pairs as seeds and left the remaining as testing data. Table 1 lists the statistics of the three datasets.

We compare CTEA with seven state-of-the-art embedding-based entity alignment methods, including four translation-based models IEAKE [7], MtransE [4], JAPE [5], and BootEA [6] and three GCN-based models GCN-Align [8], MuGCN [9] and CEAFF w/o C, which is variant of Ref. [33] without collective alignment strategy.

5.2. Training details

In SE module, each of the KGs learns its embeddings based on TransE firstly to capture its structural information. Hence each KG has a unique embedding space. However, we calculate the similarity between entities' embeddings in SE module and between entities' context representations in CM module in a unified space. We have to embed two KGs in a unified vector space. There are usually three ways to do this: (1) embed KGs in separate spaces and transfer the embeddings of one KG into the embeddings space of another KG; (2) let the two entities in a seed pair share the same embedding; (3) use pre-aligned entity pairs to generate new cross-KG triples to build a bridge between different KGs at the level of triple. And we employ the second way in this paper.

As knowledge graphs only contain positive triples, negative triples need to be constructed in advance to satisfy the demand of the margin-based objective functions in SE module. A widely-used sampling method to generate negative triples is uniform negative sampling, where a negative triple is obtained by randomly replacing the head entity or tail entity of a positive triple. But just as BootEA stated, the sampled replacer may be totally orthogonal with the original entity, so it will contribute little to embedding learning. In this paper, we apply neighbors sampling. Given entity e to be replaced, we choose its top s nearest neighbors from the embedding as candidates and randomly sampling an entity from the candidates. We initialize the embeddings of all entities and relations based on the normal distribution, and use the gradient descent optimization algorithm Ada-Grad[36] to optimize our models. The length of all embeddings is restricted to 1 to avoid trivially optimizing the objective by increasing the norm of embeddings. The following hyper-parameters are used in the experiments. For TL module, the number of topics Z = 30, the hyper-parameter for dirichlet distribution $\alpha = 0.1,\ \beta = 1$. For SE module, the embedding size for entities and relations $dim = 75;\ s = 10$ for negative sampling, i.e., 10 triples are sampled for each positive triple; the maximum of positive triples $\gamma_1 = 0.1$, the minimum of negative triples $\gamma_2 = 2$. For CM module, the number of kernels K = 3 and the convolution kernel size are 2×1, 3×1 and 4×1 respectively, the number of channels of each kernel $\Omega = 8$. μ for balancing entity embedding and entity context is set to 0.7.

We train SE 500 epoches and followed by 4000 epoches CM training to learning the parameter in CM. It is noted that we do not update parameters in SE module when train CM and do not update parameters in CM module when train SE. TL model is trained before SE and CM. And the results of TL are taken as off-line data when test CTEA.

We use hit@10, Mean Rank(MR) and Mean Reciprocal(MRR) as evaluation metrics to assess the performance of CTEA, which are defined as follows:

- **MR** Mean Rank of correct alignment,which is derived as follows:

$$MR = \frac{\sum_{e_s \in Sour_{test}} rank_{e_s}}{card(Sour_{test})},$$

  where $rank_{e_s}$ is the rank of the true target of entity e_s.
- **MRR** Mean Reciprocal of correct alignment,which is derived as follows:

$$MRR = \frac{\sum_{e_s \in Sour_{test}} \frac{1}{rank_{e_s}}}{card(Sour_{test})}.$$

Table 1
Datasets used in entity alignment.

Datasets		#Rels	#Ents	#Triples	#Seeds	#Test
$DBP15K_{zh-en}$	Chinese	2803	66469	153929	4500	10500
	English	2317	98125	237674		
$DBP15K_{ja-en}$	Japanese	2043	65744	164373	4500	10500
	English	2096	95680	233319		
DBP15Kfr-en	French	1379	66858	19219	4500	10500
	English	2209	105889	278590		

- **hit@10** Proportion of correct alignments ranked in top-10, which is calculated by:

$$hit@10 = \frac{card(\{e_s \in Sour_{test} | rank_{e_s} \leqslant 10\})}{card(Sour_{test})}.$$

Higher hit@10, MRR and lower MR indicate better performance. In test phase, for each entity in test source entity set, we compute the distance between it and all entities in test target entity set, and then record the rank of the true target. As we obtain a candidate set for each test source entity with the help of TL, we will let candidate entities to rank ahead of others to increase their probabilities be selected as the true target entity.

5.3. Overall performance

Table 2 shows the comparison results of CTEA and other embedding-based entity alignment methods. We found CTEA outperforms other methods except for CEAFF w/o C [33], especially on MR. The mainly reason is that CTEA filter some weakly related entities via TL module before aligning and explicitly take advantage of entity context information.

More specifically, compared with other trans-based models, CTEA outperforms all baselines on all data sets regarding hit@10 and MR. Comparing with the second-best method except for CEAFF, the average gains of CTEA regarding hit@10 is 5.6%, the average reductions of MR is 21. This due that CTEA explores the contexts and topic features of entities when finding their true targets, candidates generated by TL can filter many outlying target entities for source entities and context representation can while other trans-based models like JE [25], MTransE [4] and BootEA [6] only depend on KG structure and ignore other valuable features like attributes. JAPE [5] takes advantage of entity attributes but it models attributes co-occurrence using word2vec which may not appropriate to short text and thus does not appropriate to model entity attributes. Regarding MRR, CTEA outperforms other methods on DBP_{zh-en} and DBP_{ja-en}, but inferiors to BootEA [6] on DBP_{fr-en}, it may because that BootEA [6] is an iterative method where newly aligned entity pairs will be added as to seeds. When compared with GCN-based models, CTEA also achieves superior results. Even GCN-based methods GCN-Align [8] and MUGCN [9] attempt to aggregate entity neighbors to entity embedding, they do not explicitly model entity context, nor filter weakly related entities for source entities.

As for CEAFF w/o C [33], its performance is far better than CTEA and other baselines according to most of the metrics. For example, in DBP_{fr-en}, it reached 0.979 and 0.947 according to hit@10 and MRR, which is 5% and 21% higher than the second best ones, respectively. We attribute this results to its utilization of pre-trained entity name embeddings, a type of very discriminatory information, which largely contribute to its good performance. For example, entity name 'China' in English and entity name '中华人民共和国' in Chinese have very similar embedding in word embedding space already, which made the similarity between the two entities possessing these two names vary large without any training process. It would be reasonable to believe that integrating entity name embedding to CTEA or other models will also bring better results.

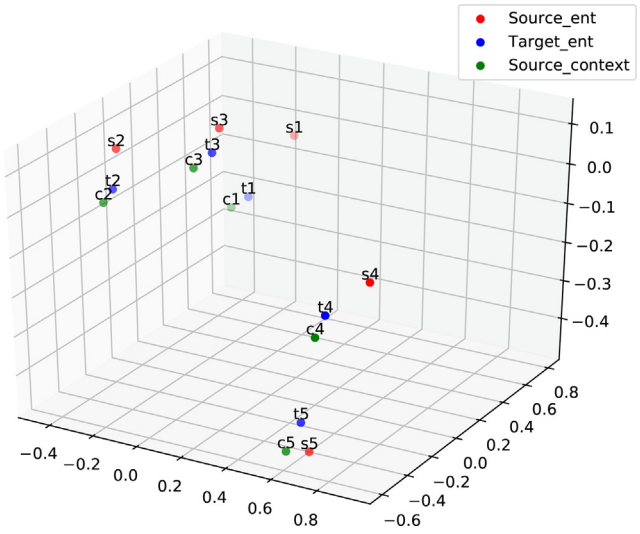

Fig. 4. Illustration of the influence of entity context. (s1:来尔沃尔足球俱乐部, s2:乌尔奇人, s3:Ipad_(第四代), s4:伯明翰, s5:金珍洙, t1:Millwall_F.C., t2:Ulch_people, t3:IPad_(4th_-generation), t4:Birmingham, t5:Kim_Jin_su, ci (i = 1,2,3,4,5):context representation of si (i = 1,2,3,4,5), respectively).

5.4. Effectiveness of modules

To verify the effectiveness of CM module and TL module, we construct two sub-frameworks by getting rid of CM and TL, respectively, and report the results in Table 2. The two modules all contribute to entity alignment.

Effectiveness of CM When getting rid of CM, the performance is a little decline, about 1% reduction on hit@10 for example. CM supplements SE when two entities to be aligned have some aligned neighbors. We select five entities from $Sour_{test}$ of DBP_{zh_en} and project their embeddings, their context representations and their true targets' embeddings to three-dimension space using PCA, as shown in Fig. 4. We find that for some test source entities, their embeddings are far from their targets' embeddings, while their context

Table 2
Entity alignment results comparing with other methods.

Methods		DBP_{zh-en}			DBP_{ja-en}			DBP_{fr-en}		
		hit@10	MRR	MR	hit@10	MRR	MR	hit@10	MRR	MR
Trans-based	JE	0.428	-	766	0.400	–	835	0.388	–	832
	MTansE	0.614	0.364	74	0.575	0.349	73	0.556	0.335	42
	IEAKE	0.735	0.516	–	0.693	0.474	–	0.685	0.451	109
	JAPE	0.745	0.490	64	0.685	0.476	99	0.667	0.430	92
	BootEA	0.848	0.703	55	0.854	0.701	57	0.874	0.731	31
GCN-based	GCN-Align	0.744	0.549	267	0.764	0.542	212	0.776	0.536	221
	MuGCN	0.844	0.611	–	0.857	0.621	–	0.870	0.621	–
	CEAFF w/o C	0.874	**0.774**	–	0.907	**0.827**	–	**0.979**	**0.947**	–
Ours	**CTEA-TL**	0.882	0.689	54	0.890	0.680	56	0.911	0.693	42
	CTEA-CM	0.893	0.697	32	0.902	0.698	30	0.917	0.700	29
	CTEA	**0.905**	0.707	**26**	**0.914**	0.704	**23**	0.923	0.716	**21**

The best score is in bold.

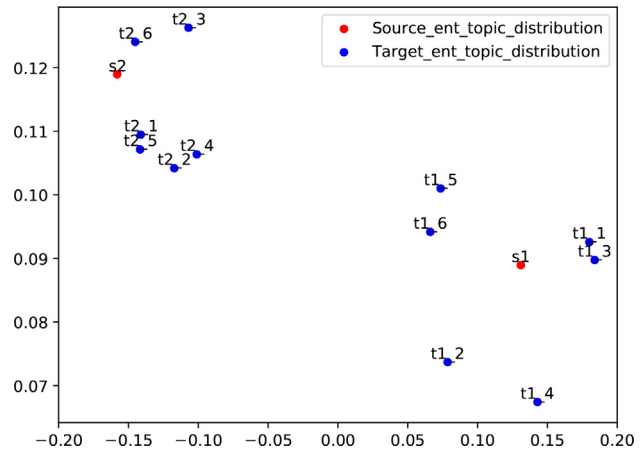

Fig. 5. The candidates obtained by entity topic. (s1:艾哈迈德.马田-达夫塔里, t1_1: Lindsey_Graham, t1_2:Ahmad_Matin-Daftari, t1_3:Samuel_Osgood, t1_4:Christine_Lagarde, t1_5:President_of_Tunisia, t1_6:Edward,_the_Black_Prince s2:卡斯柏.古斯克, t2_1:Takumi_Minamino, t2_2:Marc_Overmars, t2_3:Vagner_Love, t2_4:Kasper_Kusk, t2_5:Paul_Pogba, t2_6: Fabio_Quagliarella).

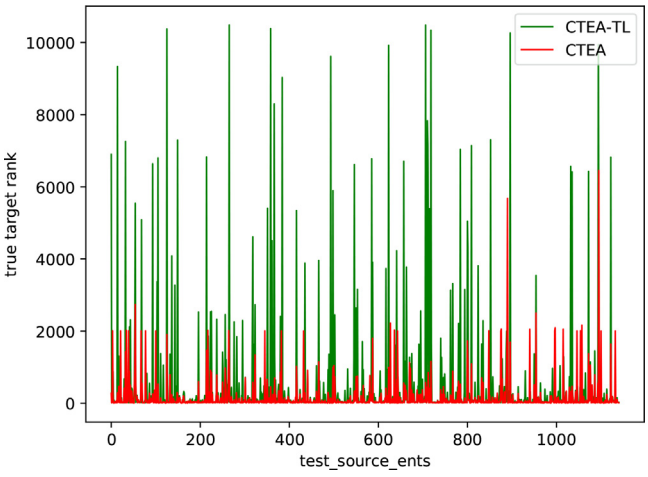

Fig. 6. True target ranks comparison between CTEA and CTEA-TL for source entities in DBP_{zh-en}.

representations are closer to their targets'embeddings, which explains the effectiveness of entity context representations.

Effectiveness of TL When getting rid of TL, the performance is obviously decline, especially for MR. It shows that TL module indeed filters the entities which have different global semantics when finding the true target of a specific source entity. As shown in Fig. 5, TL helps source entity find candidate entities having similar topic distributions with it from the target entity set. For example, for source entity *s1: 艾哈迈德.马田-达夫塔里(Ahmad_Matin-Daftari in English)* in DBPzh_en, TL recommends some entities as the candidates, such as *t1_1, t1_2, t1_3, t1_4, t1_5,t1_6* which are all politicians as these entities have attributes which are usually co-occurrent in attributes of s1. From these candidates, the following modules can find the true one more easily. To further explain that TL exactly filter a number of outlying entities, we collect the source entities in $Sour_{test}$ of DBP_{zh_en} whose true targets sort greater than 10 got from CTEA-TL, and show the ranks from both CTEA-TL and CTEA in Fig. 6. We can see the red line is lower than the green one overall, there is fewer entity with large rank when adding TL, this in line with our intention to designing TL module.

5.5. Impact of candidate number

The responsibility of TL in our framework is to filter some entities and generate a candidate set for each source entity. As TL roughly generates a candidate set, how to decide the number of candidates is a troublesome problem. If the number of candidates is too small, the true targets of some test source entities may be excluded; if the number is too large, the candidates may not work as what we want. We examine the relation between candidate number and the whole performance of CETA.

Except for topic number Z = 30 with which we report the whole results, we also select Z = 10 and Z = 50 to test how the whole performance of CTEA influenced by candidate number. We select candidate numbers τ among{1000,2000,3000,4000,5000,6000} and report the performance of CTEA under these candidate numbers in Figs. 7 and 8. As shown in Figs. 7 and 8, under Z = 30, when $\tau \leq 4000$, the whole performance increases with τ increases, because the probability of containing the true target in the candidate set is larger as τ increases. Note that the performance of CTEA is inferior to CTEA-TL when $\tau \leq 2000$, as the candidate numbers are too small in these cases where TL excludes some true targets from the candidate set. When $\tau > 4000$, the whole performance is declined gradually, as the candidate set is too large to filter the entities having little semantic similarity with the source entity, which lead to that CTEA degenerates into CTEA-TL finally. Note that we can get almost equal results when select Z = 30 and Z = 50. When Z = 10, the whole performance has the similar tendency but the results are inferior to Z = 30 and Z = 50, as the topic number is too small for TL to learn respective high-level semantics for entities and thus it can not generate proper candidates.

6. Conclusion

In this paper, we propose a novel entity alignment framework CTEA, which merge entity context and entity topic information to help aligning entities. We first learn entity topic via an improved topic model BTM4EA from entity attributes. Then JS divergence between entities are used to generate the candidates for each source entity. To take advantage of entity context to help align entity, we embed KG in low dimensional space and mine entity context from the neighbors of the center entities using multi-channel CNN. Entity embeddings and entity contexts are combined

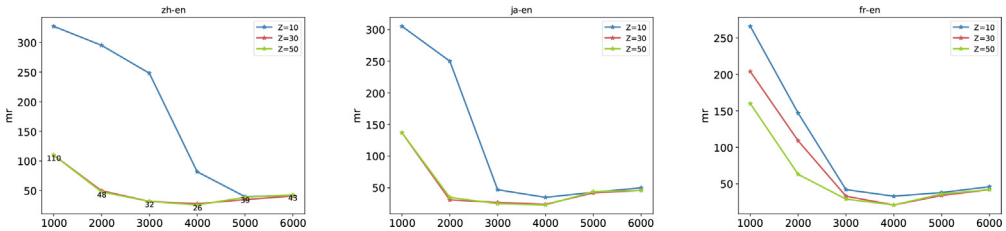

Fig. 7. Impact of candidate number on MR under different topic number.

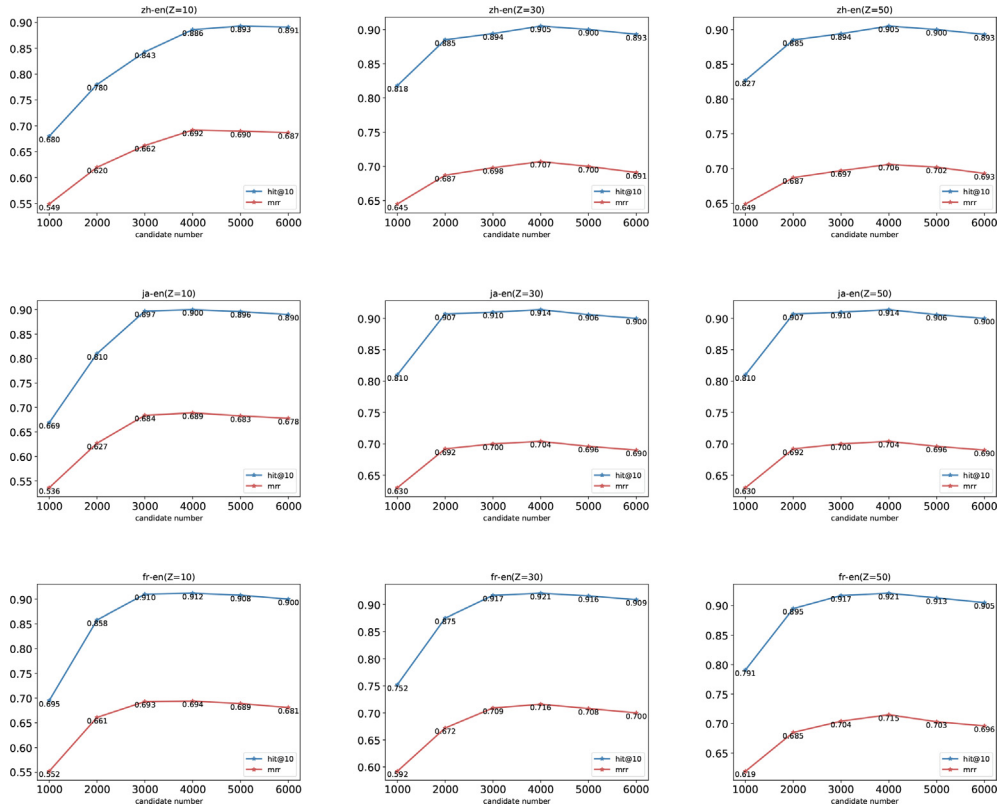

Fig. 8. Impact of candidate number on Hit@10 and MRR under different topic number.

to find the true target entities with the help of candidates. The experiments on three real-world data sets and further analysis demonstrate the effectiveness of our method. We train TL and the other two modules separately in this paper, however, the topic assignment for entities although influenced by the representations of entities. In the future, we will optimize the topic model online and con-train TL with SE and CM. And we will seek for other models, such as GAT, to model the context for entities. We use entity structural information to generate entity embeddings in this paper. While there are other features, such as entity name utilized by Refs. [33,28], entity description utilized by Ref. [29], which can be used to generate more representative entity embeddings and further more representative entity context representations. We will consider to apply these features to refine our three-steps entity alignment framework in the future.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Zhihuan Yan: Conceptualization, Methodology, Writing - original draft. **Rong Peng:** Supervision, Funding acquisition, Writing - review & editing. **Yaqian Wang:** Data curation. **Weidong Li:** Investigation, Visualization.

Acknowledgments

This work is supported by the National Key Research and Development Plan of China under Grant Nos. 2017YFB0503702, 2016YFB0501801, and National Natural Science Foundation of China under Grant No. 61170026.

References

[1] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, S. Hellmann, Dbpedia – a crystallization point for the web of data, J. Web Semant. 7 (3) (2009) 154–165, https://doi.org/10.1016/j.websem.2009.07.002.

[2] T. Rebele, F.M. Suchanek, J. Hoffart, J. Biega, E. Kuzey, G. Weikum, YAGO: A multilingual knowledge base from wikipedia, wordnet, and geonames, in: The Semantic Web - ISWC 2016 - 15th International Semantic Web Conference, Kobe, Japan, October 17-21, 2016, Proceedings, Part II, 2016, pp. 177–185, doi: 10.1007/978-3-319-46547-0_19..

[3] R. Navigli, S.P. Ponzetto, Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network, Artif. Intell. 193 (2012) 217–250, https://doi.org/10.1016/j.artint.2012.07.001.

[4] M. Chen, Y. Tian, M. Yang, C. Zaniolo, Multilingual knowledge graph embeddings for cross-lingual knowledge alignment, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19–25, 2017, 2017, pp. 1511–1517, doi: 10.24963/ijcai.2017/209..

[5] Z. Sun, W. Hu, C. Li, Cross-lingual entity alignment via joint attribute-preserving embedding, in: The Semantic Web - ISWC 2017–16th International Semantic Web Conference, Vienna, Austria, October 21–25, 2017, Proceedings, Part I, 2017, pp. 628–644, doi: 10.1007/978-3-319-68288-4_37..

[6] Z. Sun, W. Hu, Q. Zhang, Y. Qu, Bootstrapping entity alignment with knowledge graph embedding, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13–19, 2018, Stockholm, Sweden., 2018, pp. 4396–4402. doi::10.24963/ijcai.2018/611..

[7] H. Zhu, R. Xie, Z. Liu, M. Sun, Iterative entity alignment via joint knowledge embeddings, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19–25, 2017, 2017, pp. 4258–4264, doi: 10.24963/ijcai.2017/595..

[8] Z. Wang, Q. Lv, X. Lan, Y. Zhang, Cross-lingual knowledge graph alignment via graph convolutional networks, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018, 2018, pp. 349–357..

[9] Y. Cao, Z. Liu, C. Li, Z. Liu, J. Li, T. Chua, Multi-channel graph neural network for entity alignment, in: Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28–August 2, 2019, Volume 1: Long Papers, 2019, pp. 1452–1461..

[10] X. Yan, J. Guo, Y. Lan, X. Cheng, A biterm topic model for short texts, in: 22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13–17, 2013, 2013, pp. 1445–1456, doi: 10.1145/2488388.2488514..
[11] A. Bordes, N. Usunier, A. García-Durán, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, in: Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5–8, 2013, Lake Tahoe, Nevada, United States., 2013, pp. 2787–2795..
[12] Y. Lin, Z. Liu, M. Sun, Y. Liu, X. Zhu, Learning entity and relation embeddings for knowledge graph completion, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25–30, 2015, Austin, Texas, USA., 2015, pp. 2181–2187..
[13] 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 of the Asian Federation of Natural Language Processing, ACL 2015, July 26–31, 2015, Beijing, China, Volume 1: Long Papers, 2015, pp. 687–696..
[14] T. Dettmers, P. Minervini, P. Stenetorp, S. Riedel, Convolutional 2d knowledge graph embeddings, in: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, 2018, pp. 1811–1818..
[15] D.Q. Nguyen, T.D. Nguyen, D.Q. Nguyen, D.Q. Phung, A novel embedding model for knowledge base completion based on convolutional neural network, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1–6, 2018, Volume 2 (Short Papers), 2018, pp. 327–333..
[16] M.S. Schlichtkrull, T.N. Kipf, P. Bloem, R. van den Berg, I. Titov, M. Welling, Modeling relational data with graph convolutional networks, in: ESWC, 2018, pp. 593–607, doi: 10.1007/978-3-319-93417-4_38..
[17] D. Nathani, J. Chauhan, C. Sharma, M. Kaul, Learning attention-based embeddings for relation prediction in knowledge graphs, in: Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28–August 2, 2019, Volume 1: Long Papers, 2019, pp. 4710–4723.
[18] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, 2017, URL:https://openreview.net/forum?id=SJU4ayYgl..
[19] Z. Wang, J. Zhang, J. Feng, Z. Chen, Knowledge graph embedding by translating on hyperplanes, in: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27–31, 2014, Québec City, Québec, Canada., 2014, pp. 1112–1119..
[20] G. Ji, K. Liu, S. He, J. Zhao, Knowledge graph completion with adaptive sparse transfer matrix, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12–17, 2016, Phoenix, Arizona, USA, 2016, pp. 985–991..
[21] W. Qian, C. Fu, Y. Zhu, D. Cai, X. He, Translating embeddings for knowledge graph completion with relation attention mechanism, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13–19, 2018, Stockholm, Sweden., 2018, pp. 4286–4292, doi: 10.24963/ijcai.2018/596..
[22] B. Yang, W. Yih, X. He, J. Gao, L. Deng, Embedding entities and relations for learning and inference in knowledge bases, in: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings, 2015..
[23] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, G. Bouchard, Complex embeddings for simple link prediction, in: Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19–24, 2016, 2016, pp. 2071–2080, URL:http://proceedings.mlr.press/v48/trouillon16.html..
[24] B. Shi, T. Weninger, Proje: Embedding projection for knowledge graph completion, in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4–9, 2017, San Francisco, California, USA, 2017, pp. 1236–1242..
[25] Y. Hao, Y. Zhang, S. He, K. Liu, J. Zhao, A joint embedding method for entity alignment of knowledge bases, in: Knowledge Graph and Semantic Computing: Semantic, Knowledge, and Linked Big Data - First China Conference, CCKS 2016, Beijing, China, September 19–22, 2016, Revised Selected Papers, 2016, pp. 3–14, doi: 10.1007/978-981-10-3168-7_1..
[26] B.D. Trisedya, J. Qi, R. Zhang, Entity alignment between knowledge graphs using attribute embeddings, in: The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019, 2019, pp. 297–304..
[27] M. Chen, Y. Tian, K. Chang, S. Skiena, C. Zaniolo, Co-training embeddings of knowledge graphs and entity descriptions for cross-lingual entity alignment, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13–19, 2018, Stockholm, Sweden, 2018, pp. 3998–4004, doi: 10.24963/ijcai.2018/556..
[28] Q. Zhang, Z. Sun, W. Hu, M. Chen, L. Guo, Y. Qu, Multi-view knowledge graph embedding for entity alignment, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10–16, 2019, 2019, pp. 5429–5435, doi: 10.24963/ijcai.2019/754..
[29] H. Yang, Y. Zou, P. Shi, W. Lu, J. Lin, X. Sun, Aligning cross-lingual entities with multi-aspect information, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3–7, 2019, 2019, pp. 4430–4440, doi: 10.18653/v1/D19-1451, URL:https://doi.org/10.18653/v1/D19-1451..
[30] C. Li, Y. Cao, L. Hou, J. Shi, J. Li, T. Chua, Semi-supervised entity alignment via joint knowledge embedding model and cross-graph model, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3–7, 2019, 2019, pp. 2723–2732, doi: 10.18653/v1/D19-1274, URL:https://doi.org/10.18653/v1/D19-1274..
[31] R. Ye, X. Li, Y. Fang, H. Zang, M. Wang, A vectorized relational graph convolutional network for multi-relational network alignment, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10–16, 2019, 2019, pp. 4135–4141, doi: 10.24963/ijcai.2019/574, URL:https://doi.org/10.24963/ijcai.2019/574..
[32] K. Xu, L. Wang, M. Yu, Y. Feng, Y. Song, Z. Wang, D. Yu, Cross-lingual knowledge graph alignment via graph matching neural network, in: Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers, 2019, pp. 3156–3161..
[33] W. Zeng, X. Zhao, J. Tang, X. Lin, Collective embedding-based entity alignment via adaptive features, CoRR abs/1912.08404, arXiv:1912.08404, URL:http://arxiv.org/abs/1912.08404..
[34] M. Qu, J. Tang, Y. Bengio, Weakly-supervised knowledge graph alignment with adversarial learning, CoRR abs/1907.03179, arXiv:1907.03179, URL:http://arxiv.org/abs/1907.03179..
[35] Z. Yan, R. Peng, Y. Wang, W. Li, Enhance knowledge graph embedding via fake triples, in: International Joint Conference on Neural Networks, IJCNN 2019 Budapest, Hungary, July 14–19, 2019, 2019, pp. 1–7, doi: 10.1109/IJCNN.2019.8852374..
[36] J.C. Duchi, E. Hazan, Y. Singer, Adaptive subgradient methods for online learning and stochastic optimization, J. Mach. Learn. Res. 12 (2011) 2121–2159.

Zhihuan Yan is a Ph.D. candidate in the School of Computer Science at Wuhan University. She received the Master's degree in Mathematics and Applied Mathematics from University of Electronic Science and Technology of China (UESTC). Her research interests include entity alignment, knowledge graph embedding, Graph Convolutional Network and hyperbolic deep learning.

Rong Peng is a professor in the School of Computer Science at Wuhan University. Her main research interests include requirements engineering, knowledge engineering, and service computing.

Yaqian Wang is a Ph.D. candidate in the School of Computer Science at Wuhan University. She received the B.S. degree in Software Engineering from Henan University. Her research interests include neural relation extraction, entity classification and named entity recognition.

Weidong Li is a Ph.D. candidate in the School of Computer Science at Wuhan University. He received the B.S. degree in Information and Computing Science from Henan University of Technology. His main research interests include knowledge graph completion, natural language generation and machine learning.