



Leveraging Lexical Semantic Information for Learning Concept-Based Multiple Embedding Representations for Knowledge Graph Completion

Yashen Wang¹(✉), Yifeng Liu¹, Huanhuan Zhang¹, and Haiyong Xie^{1,2}

¹ China Academy of Electronics and Information Technology, Beijing, China
yashen_wang@126.com, yliu@csdslab.net, huanhuanz_bit@139.com,
haiyong.xie@ieee.org

² University of Science and Technology of China, Hefei, Anhui, China

Abstract. Knowledge graphs (KGs) are important resources for a variety of natural language processing tasks but suffer from incompleteness. To address this challenge, a number of knowledge graph completion (KGC) methods have been developed using low-dimensional graph embeddings. Most existing methods focus on the structured information of triples in encyclopaedia KG and maximize the likelihood of them. However, they neglect semantic information contained in lexical KG. To overcome this drawback, we propose a novel KGC method (named as TransC), that integrates the structured information in encyclopaedia KG and the entity concepts in lexical KG, which describe the categories of entities. Since all entities appearing in the head (or tail) position with the same relation have some common concepts, we introduce a novel semantic similarity to measure the distinction of entity semantics with the concept information. And then TransC utilizes concept-based semantic similarity of the related entities and relations to capture prior distributions of entities and relations. With the concept-based prior distributions, TransC generates multiple embedding representations of each entity in different contexts and estimates the posterior probability of entity and relation prediction. Experimental results demonstrate the efficiency of the proposed method on two benchmark datasets.

Keywords: Knowledge graph completion · Concept information · Representation learning

1 Introduction

Knowledge Graphs (KGs) are graph-structured knowledge bases, where factual knowledge is represented in the form of relationships between entities. Knowledge Graphs have become a crucial resource for many tasks in machine learning, data mining, and artificial intelligence applications including question answering [34],

entity linking/disambiguation [7], fact checking [29], and link prediction [44]. In our view, KGs are an example of a heterogeneous information network containing entity-nodes and relationship-edges corresponding to RDF-style triples (h, r, t) where h represents a head entity, and r is a relationship that connects h to a tail entity t .

KGs are widely used for many practical tasks, however, their completeness are not guaranteed. Nonetheless, KGs are far from completion. For instance consider Freebase, a core element in the Google Knowledge Vault project: 71% of the persons described in Freebase have no known place of birth, 75% of them have no known nationality, while the coverage for less frequent predicates can be even lower. Therefore, it is necessary to develop Knowledge Graph Completion (KGC) methods to find missing or errant relationships with the goal of improving the general quality of KGs, which, in turn, can be used to improve or create interesting downstream applications. Motivated by the linear translation phenomenon observed in well trained word embeddings [18], Many Representation Learning (RL) based algorithms [3, 15, 37, 41], have been proposed, aiming at embedding entities and relations into a vector space and predicting the missing element of triples. These models represents the head entity h , the relation r and the tail entity t with vectors \mathbf{h} , \mathbf{r} and \mathbf{t} respectively, which were trained so that $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$.

The objects most KGC models handle are encyclopedic KG (e.g., Freebase). Although these models have significantly improved the embedding representations and increased the prediction accuracy, there is still room for improvement by exploiting semantic information in the representation of entities. Generally speaking, semantic information includes concepts, descriptions, lexical categories and other textual information. As discussed in [12, 31, 35, 38], it is essential to utilize lexical KGs to help the machine to understand the world facts and the semantics. That is, the knowledge of the language should be used. Encyclopedic KGs contain facts such as Barack Obama's birthday and birthplace, while lexical KGs could definitely indicate that birthplace and birthday are properties of a person.

Generally, each entity or relation may have different semantics in different triples. For example, in the triple (David_Beckham, place_of_birth, London), David Beckham is a person, while in (David_Beckham, player_of, Manchester_United), David Beckham is a player or athlete. Unfortunately, most recent works represent each entity as a single vector which cannot capture the uncertain semantics of entities. To address the above-mentioned issue, we propose a concept-based multiple embedding model (TransC). TransC fully utilizes the entity *concept* information which represents the domains or categories of entities in lexical KG Probase. Probase is widely used in research about short-text understanding [31, 32, 39] and text representation [12, 36]. Probase uses an automatic and iterative procedure to extract concept knowledge from 1.68 billion Web pages. It contains 2.36 millions of open domain terms. Each term is a concept, an instance, or both. Meanwhile, it provides around 14 millions relationships with two kinds of important knowledge related to concepts: concept-attribute

co-occurrence (isAttributeOf) and concept-instance co-occurrence (isA). Moreover, Probase provides huge number of high-quality and robust concepts without builds. Therefore, we model each entity as multiple semantic vectors with concept information, and construct multiple concepts of relations from common concepts of related entities.

We utilize the concept-based semantic similarity to incorporate prior probability in the optimization objective. This is because all entities appearing in the head (or tail) with the same relation have some common concepts. Therefore, the prior distribution of the missing element could be derived from the semantic similarity between the missing element and the others. In the “David Beckham” example mentioned above, if the head of (David_Beckham, player_of, Manchester_United) is missing, we can predict the head is an entity with “player” or “athlete” since we know the relation is “player_of” and the tail is “Manchester_United”, a football club.

In summary, the contributions of this work are: (i) proposing a novel knowledge base completion model that combines structured information in encyclopedic KG and concept information in lexical KG. To the best of our knowledge, this is the first study aiming at combing the encyclopedic KG and the lexical KG for knowledge graph completion (KGC) task. (ii) showing the effectiveness of our model by outperforming baselines on two benchmark datasets for knowledge base completion task.

2 Related Work

Many knowledge graphs have recently arisen, pushed by the W3C recommendation to use the resource description framework (RDF) for data representation. Examples of such knowledge graphs include DBpedia [1], Freebase [2] and the Google Knowledge Vault [8]. Motivating applications of knowledge graph completion include question answering [5] and more generally probabilistic querying of knowledge bases [11, 22]. First approaches to relational learning relied upon probabilistic graphical models [10], such as bayesian networks [28] and markov logic networks [25, 26]. Then, asymmetry of relations was quickly seen as a problem and asymmetric extensions of tensors were studied, mostly by either considering independent embeddings [9] or considering relations as matrices instead of vectors in the RESCAL model [23]. Pairwise interaction models were also considered to improve prediction performances. For example, the Universal Schema approach [27] factorizes a 2D unfolding of the tensor (a matrix of entity pairs vs. relations).

Nowadays, a variety of low-dimensional representation-based methods have been developed to work on the KGC task. These methods usually learn continuous, low-dimensional vector representations (i.e., embeddings) for entities and relationships by minimizing a margin-based pairwise ranking loss [14]. The most widely used embedding model in this category is TransE [3], which views relationships as translations from a head entity to a tail entity on the same low-dimensional plane.

Based on the initial idea of treating two entities as a translation of one another (via their relationship) in the same embedding plane, several models have been introduced to improve the initial TransE model. The newest contributions in this line of work focus primarily on the changes in how the embedding planes are computed and/or how the embeddings are combined. For example, the entity translations in TransH [37] are computed on a hyperplane that is perpendicular to the relationship embedding. In TransR [15] the entities and relationships are embedded on separate planes and then the entity-vectors are translated to the relationships plane. Structured Embedding (SE) [4] creates two translation matrices for each relationship and applies them to head and tail entities separately. Knowledge Vault [8] and HolE [21], on the other hand, focus on learning a new combination operator instead of simply adding two entity embeddings element-wise. Take HolE as example, the circular correlation is used for combining entity embeddings, measuring the covariance between embeddings at different dimension shifts.

Semantic information, such as types, descriptions, lexical categories and other textual information, is an important supplement to structured information in KGs. DKRL [42] represents entity descriptions as vectors for tuning the entity and relation vectors. SSP [40] modifies TransH by using the topic distribution of entity descriptions to construct semantic hyperplanes. Entity descriptions are also used to derive a better initialization for training models [16]. With type information, type-constraint model [13] selects negative samples according to entity and relation types. In a similar way, TransT [17] leveraged the type information for the representation of entity. However, TransT have to construct or extend entity concepts from other semantic resources (e.g., WordNet), if there is no explicit concept information in a KG. TKRL [43] encodes type information into multiple representations in KGs with the help of hierarchical structures. It is a variant of TransR with semantic information and it is the first model introducing concept information.

3 Methodology

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 . We model each entity as multiple semantic vectors with concept information to represent entities more accurately. Different from using semantic-based linear transformations to separate the mixed representation, the proposed TransC models the multiple semantics separately and utilizes the semantic similarity to distinguish entity semantics. Moreover, we measure the semantic similarity of entities and relations based on entity concepts and relation concepts.

3.1 Semantic Similarity Based on Concept

As discussed in [17], all of the entities located in the head position (or tail position) with the same relation may have some common entity types or concepts, as

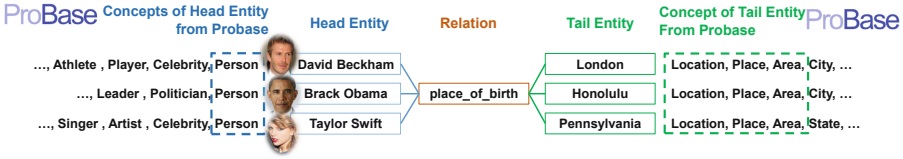


Fig. 1. example showing that the entities in the head or tail of a relation have some common concepts from the lexical KG Probase

shown in Fig. 1. In this example, all the head entities have “Person” concept and all the tail entities have concepts of “Location”, “Place” and “Area”. Therefore, we could see that, “Person” is the head concept of relation “place_of_birth”, and “Location”, “Place” and “Area” are the tail concepts of this relation. Based on aforementioned correlation, this paper introduces a concept-based semantic similarity, which utilizes *entity concepts* to construct *relation concepts*. Apparently, each relation (such as “place_of_birth” in Fig. 1) relates two components, and thus each relation r has two concept sets: (i) head concept set C_r^{head} , consisting of concepts of the entities occurring in the head position; and (ii) tail concept set C_r^{tail} , consisting of concepts of the entities occurring in the tail position. From the lexical KG Probase, we could distill entities appearing in the head position of relation r to form the head entity set, denoted as E_r^{head} . Similarly, the tail entity set, denoted as E_r^{tail} , could be constructed in the same way. Moreover, given entity e , we denote its concept set as C_e , consisting the corresponding concepts deriving from Probase by leveraging single instance conceptualization algorithm [12, 36, 39]. With efforts above, given r relation, the corresponding C_r^{head} and C_r^{tail} could be defined as follows:

$$C_r^{head} = \bigcap_{e \in E_r^{head}} C_e \tag{1}$$

$$C_r^{tail} = \bigcap_{e \in E_r^{tail}} C_e \tag{2}$$

Therefore, the semantic similarity between the relation and the head entity, to measure the distinction of entity semantics with the concept information, is defined as:

$$\text{sim}(r_{head}, h) = \frac{|C_r^{head} \cup C_h|}{|C_r^{head}|} \tag{3}$$

Similarly, the semantic similarity between the relation and the tail entity is:

$$\text{sim}(r_{tail}, t) = \frac{|C_r^{tail} \cup C_t|}{|C_r^{tail}|} \tag{4}$$

And the semantic similarity between the head entity h and tail entity t is

$$\text{sim}(h, t) = \frac{|C_h \cup C_t|}{|C_h|} \tag{5}$$

3.2 Methodology

In our perspective, the prediction probability is a *conditional probability* because except the missing element, rest of the two elements in a triple are known. E.g., when predicting the tail entity for a triple $(h, r, ?)$, we expect to maximize the probability of t under the condition that the given triple satisfies the principle $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ and the head entity and relation are h and r . Wherein, \mathbf{h} , \mathbf{r} , and \mathbf{t} denote the embedding representation of h , r , and t respectively. Intuitively, we could denote this conditional probability as $\mathcal{P}(t|h, r, fact)$, meaning that triple $(h, r, ?)$ is a fact, which means that the triple satisfies $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$ principle. According to Bayes theorem [6], $\mathcal{P}(t|h, r, fact)$ could be reformed as follows:

$$\mathcal{P}(t|h, r, fact) = \frac{\mathcal{P}(fact|h, r, t)\mathcal{P}(t|h, r)}{\mathcal{P}(fact|h, r)} \propto \mathcal{P}(fact|h, r, t)\mathcal{P}(t|h, r) \quad (6)$$

The above-mentioned Eq. 6 consists of two components: (i) $\mathcal{P}(fact|h, r, t)$ is the likelihood that (h, r, t) is a fact, which is estimated by the multiple embedding representations; (ii) $\mathcal{P}(t|h, r)$ is the prior probability of the tail entity t , estimated by the semantic similarity.

Then, we describe how to estimate the prior probabilities Eq. (6). We assume that, the prior distribution of the missing element could be derived from the semantic similarity between the missing element and the others. For example, when we predict t in the triple (h, r, t) , the entities with more common or similar concepts belonging to r and h , have higher probability. Hence, the semantic similarity between t and its context $(?, h, r)$ could be utilized to estimate t 's prior probability:

$$\mathcal{P}(t|h, r) \propto \text{sim}(r_{tail}, t)^{\alpha_{tail}} \text{sim}(h, t)^{\alpha_{relation}} \quad (7)$$

wherein, $\text{sim}(r_{tail}, t)$ is the semantic similarity between the relation r and the tail entity t , and $\text{sim}(h, t)$ is the semantic similarity between the head entity h and tail entity t . Furthermore, $\alpha_{tail} \in \{0, 1\}$ and $\alpha_{relation} \in \{0, 1\}$ are the concept similarity weights, because h and r have different impacts on the prior probability of t .

Similarly, the objective of the head entity prediction is

$$\mathcal{P}(h|r, t, fact) \propto \mathcal{P}(fact|h, r, t)\mathcal{P}(h|r, t) \quad (8)$$

This paper also estimates prior probability Eq. (8) by the concept-based semantic similarity. Similarly, the prior estimation of head entity h is defined as follows:

$$\mathcal{P}(h|r, t) \propto \text{sim}(r_{head}, h)^{\alpha_{head}} \text{sim}(t, h)^{\alpha_{relation}} \quad (9)$$

Wherein, $\text{sim}(r_{head}, h)$ is the semantic similarity between the relation r and the head entity h , and $\alpha_{head} \in \{0, 1\}$ is also the concept similarity weight, because t and r have different impacts on the prior probability of h . And the objective of the relation prediction is

$$\mathcal{P}(r|h, t, fact) \propto \mathcal{P}(fact|h, r, t)\mathcal{P}(r|h, t) \quad (10)$$

By the similar derivation, the prior estimation of relation r , i.e., the prior probability Eq. (10) could be estimated by leveraging the concept-based semantic similarity.

$$\mathcal{P}(r|h, t) \propto \text{sim}(r_{head}, h)^{\alpha_{head}} \text{sim}(r_{tail}, t)^{\alpha_{tail}} \quad (11)$$

3.3 Multiple Semantic Vector Representations

We adopt the similar assumption as discussed in [17] to generate multiple semantic vector representations for each entity, to accurately model the ubiquitous rich semantics, while each relation is still represented as a single vector.

Taking the previous TransE as an example. TransE represents each entity as a single vector, trying to describe (or compact) all semantics of the given entity. There is only one vector representation for an entity in TransE. Thus the vector representation is not accurate for any entity semantics, and discards the rich semantic representations of the given entity. To overcome this drawback, the proposed TransC represents each entity concept as a concept vector, and represents each entity as a set of concept vectors, following the assumption that relations have single semantic and entities have multiple semantics [17]. Hence separate representations of entity semantics describe the relationship among a triple more accurately. The likelihood of the vector representations for the triple $\mathcal{P}(fact, h, r, t)$ (in Eq. 6) could be defined as below:

$$\mathcal{P}(fact, h, r, t) = \sum_{i=1}^{|C_h|} \sum_{j=1}^{|C_t|} w_{h,i} w_{t,j} f_r(h_i, t_j) \quad (12)$$

where $|C_h|$ and $|C_t|$ are the number of concepts of head entity h and tail entity t , by leveraging single instance conceptualization algorithm based on Probase [24, 35, 39]; $\{w_{h,1}, \dots, w_{h,|C_h|}\}$ and $\{w_{t,1}, \dots, w_{t,|C_t|}\}$ are the distributions of random variables of h and t ; $f_r(h_i, t_j)$ is the likelihood of the component with i -th concept vector \mathbf{h}_i of the head entity h and j -th concept vector \mathbf{t}_j of the tail entity t . The previous models represents the head entity h , the relation r and the tail entity t with vectors \mathbf{h} , \mathbf{r} and \mathbf{t} respectively, which were trained so that $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$, which have been viewed as a principle. This paper also following this principle. Thus, Motivated by the linear translation phenomenon observed in well trained word embeddings [3, 18], this paper defines $f_r(h_i, t_j)$ in the form of the widely-used energy function, as follows:

$$f_r(h_i, t_j) = \|\mathbf{h}_i + \mathbf{r} - \mathbf{t}_j\|_l \quad (13)$$

Wherein, \mathbf{h}_i , \mathbf{r} and \mathbf{t}_j are the vectors of h , r and t . $l = 1$ or $l = 2$, which means either the l_1 or the l_2 norm of the vector $\mathbf{h}_i + \mathbf{r} - \mathbf{t}_j$ will be used depending on the performance on the validation set.

We model the generating process of semantic vectors as a Dirichlet process [33] like TransG [41]. In training process, the probability that the head entity

(or the tail entity) in each triple generates a new (denoted as the superscript *) concept vector, could be computed as follows:

$$\mathcal{P}_{head}^*(h, r, t) = (1 - \text{sim}(r_{head}, h)) \frac{\beta e^{-\|r_{head}\|_1}}{\beta e^{-\|r_{head}\|_1} + \mathcal{P}(fact|h, r, t)} \quad (14)$$

This formula means that, if the current set of concepts could accurately represent the head entity h , the new concept semantics may be generated. Wherein, β is the scaling parameter controlling the generation probability [17]. Similarly, the generation probability of new concept vector of the tail entity t could be defined as follows:

$$\mathcal{P}_{tail}^*(h, r, t) = (1 - \text{sim}(r_{tail}, t)) \frac{\beta e^{-\|r_{tail}\|_1}}{\beta e^{-\|r_{tail}\|_1} + \mathcal{P}(fact|h, r, t)} \quad (15)$$

3.4 Optimization with Concept Domain Sampling

Recall that we need to sample a negative triple (h', r, t') to compute hinge loss, given a positive triple $(h, r, t) \in \Delta$. The distribution of negative triple is denoted by Δ' . Previous work [3, 15, 20] generally constructs a set of corrupted triples by replacing the head entity or tail entity with a random entity uniformly sampled from the KG. However, uniformly sampling corrupted entities may not be optimal. Often, the head and tail entities associated a relation can only belong to a specific concept domain or category. E.g., in Fig. 1, the prime candidate domain for head entities is “Person”. When the corrupted entity comes from other concept domains, it is very easy for the model to induce a large energy gap between true triple and corrupted one. As the energy gap exceeds some threshold value, there will be no training signal from this corrupted triple. In comparison, if the corrupted entity comes from the same concept domain, the task becomes harder for the model, leading to more consistent training signal.

Motivated by this observation, we propose to sample corrupted head or tail from entities in the same concept domain with a probability \mathcal{P}_r and from the whole entity set with probability $1 - \mathcal{P}_r$. In the rest of the paper, we refer to the new proposed sampling method as *concept domain sampling*.

With efforts above, we define the For a triple (h, r, t) in the training set Δ , we sample its negative triple $(h', r', t') \notin \Delta$ by replacing one element with another entity or relation. When predicting different elements of a triple, we replace the corresponding elements to obtain the negative triples, wherein the negative triple set could be denoted as $\Delta'_{(h,r,t)}$. With efforts above, we denote the prediction error as $l(h, r, t, h', r', t')$. Therefore, the optimization function could be viewed as the sum of prediction errors with the above-mentioned concept domain sampling, as follows:

$$\sum_{(h,r,t) \in \Delta} \sum_{(h',r',t') \in \Delta'} \max \{0, \gamma + l(h, r, t, h', r', t')\} \quad (16)$$

The stochastic gradient descent (SGD) strategy [19] is applied to optimize the optimization function in the proposed algorithm. To optimize the parameters in the formulae Eq. (16), we defined the prediction error as follows:

$$l(h, r, t, h', r', t') = \begin{cases} -\ln \mathcal{P}(h|r, t, fact) + \ln \mathcal{P}(h'|r, t, fact) & h' \neq h \\ -\ln \mathcal{P}(t|h, r, fact) + \ln \mathcal{P}(t'|h, r, fact) & t' \neq t \\ -\ln \mathcal{P}(r|h, t, fact) + \ln \mathcal{P}(r'|h, t, fact) & r' \neq r \end{cases} \quad (17)$$

4 Experiments

We evaluate our proposed TransC on several experiments. Generally, The Knowledge Graph Completion (KGC) task could be divided into two non-mutually exclusive sub-tasks: (i) Entity Prediction task, and (ii) Relationship Prediction task. We evaluate our model on both tasks with benchmark static datasets. Moreover, Triple Classification task is also introduced for our comparative analysis.

4.1 Datasets and Baselines

To evaluate entity prediction, link prediction and triple classification, we conduct experiments on the **WN18** (WordNet) and **FB15k** (Freebase) introduced by [3] and use the same training/validation/test split as in [3]. The information of the two datasets is given in Table 1. Wherein, $\#E$ and $\#R$ denote the number of entities and relation types respectively. $\#Train$, $\#Valid$ and $\#Test$ are the numbers of triple in the training, validation and test sets respectively. Concept information of entities in FB15K and WN18 is generated by instance conceptualization algorithm based on Probase [24, 39].

The baselines include **TransE** [3], **TransH** [37], and **TransR** [15], which didn't utilize semantics information. Moreover, four semantic-based models are also included: (i) **TKRL** [43] and **TransT** [17] utilize entity types; (ii) **DKRL** [42] and **SSP** [40] take advantage of entity descriptions. Two widely-used measures are considered as evaluation metrics in our experiments: (i) Mean Rank, indicating the mean rank of original triples in the corresponding probability ranks; $HITS@N$, indicating the proportion of original triples whose rank is not larger than N . Lower mean rank or higher Hits@10 mean better performance. What's more, we follow [3] to report the filter results, i.e., removing all other correct candidates h in ranking, which is called the "Filter" setting. In contrast to this stands the "Raw" setting.

4.2 Entity Prediction

The Entity Prediction task takes a partial triple $(h, r, ?)$ as input and produces a ranked list of candidate entities as output. Our Entity Prediction task utilizes FB15K dataset and WN18 dataset as benchmark dataset, and utilizes Mean Rank and $HITS@10$ as evaluation metric.

Table 1. Statistics of FB15k and WN18 used in experiments.

Dataset	#E	#R	#Train	#Valid	#Test
WN18	40,943	18	141,442	5,000	5,000
FB15k	14,951	1,345	483,142	50,000	59,071

Table 2. Evaluation results of entity prediction on FB15k.

	Mean rank		HITS10 (%)	
	Raw	Filter	Raw	Filter
TransE	238	143	46.4	62.1
TransH	212	87	45.7	64.4
TransR	199	77	47.2	67.2
DKRL	181	91	49.6	67.4
TKRL	202	87	50.3	73.4
SSP	163	82	57.2	79.0
TransT	199	46	53.3	85.4
TransC(Concept.)	180	64	57.1	84.2
TransC(Multiple.)	207	55	52.7	85.9
TransC	175	44	58.4	86.7

Following [17], the same protocol used in previous studies is utilized here. For each triple (h, r, t) , we replace the tail t (or the head h) with the concept domain sampling strategy discussed in Sect. 3.4. We calculate the probabilities of all replacement triples and rank these probabilities in descending order.

As the datasets are the same, we directly reuse the best results of several baselines from the literature [15, 17, 37, 43]. In both “Raw” setting and “Filter” setting, a higher HITS@10 and a lower Mean Rank mean better performance. The optimal-parameter configurations are described as follows: (i) For dataset WN18, the learning rate is 0.001, the vector dimension is 100, the margin is 3, the scaling parameter β is 0.0001, concept similarity weights $\alpha_{head} = \alpha_{tail} = 1$, and $\alpha_{relation} = 0$; (ii) For dataset FB15K, the learning rate is 0.0005, the vector dimension is 150, the margin is 3, the scaling parameter β is 0.0001, concept similarity weights $\alpha_{head} = \alpha_{tail} = 1$, and $\alpha_{relation} = 0$.

The overall entity prediction results on FB15K and WN18 are reported in Tables 2 and 3, respectively. It is worth mentioning that, both the proposed **TransC** and previous [17] utilize multiple semantic vectors for representing the entity. In Table 2, For **TransC**, “Concept.” means concept information is used, and “Multiple.” means entities are represented as multiple vectors. At the end of the row, “**TransC**” means the complete model combing “Concept.” with “Multiple.”. From the result, we observe that: **TransC** outperforms all baselines on FB15k with “Filter” setting. E.g., compared with **TransT** and **SSP**, **TransC**

Table 3. Evaluation results of entity prediction on WN18.

	Mean rank		HITS10 (%)	
	Raw	Filter	Raw	Filter
TransE	263	251	75.4	89.2
TransH	401	338	73.0	82.3
TransR	238	225	79.8	92.0
DKRL	202	198	77.4	92.4
TKRL	180	167	80.6	94.1
SSP	168	156	81.2	93.2
TransT	137	130	92.7	97.4
TransC	136	125	94.5	96.9

improves the Mean Rank by 4.35% and 46.34%, and improves the HIT@10 by 1.52% and 9.75%. On WN18, **TransT** achieve the best results on metric HIT@10 for “Filter” setting. Beyond that, **TransC** performs the best. This shows that our method successfully utilizes conceptual information, and that multiple conceptual vectors could capture the different semantics of each entity more accurately than the linear transformation of a single entity vector. While **SSP** achieves the best performance for “Raw” setting on the metric mean rank.

Similar to **TransT**, the proposed **TransC** has the largest difference between the results of “Raw” and “Filter” settings on FB15K. Different from using semantic-based linear transformations to separate the mixed representation, the proposed **TransC** models the multiple semantics separately and utilizes the semantic similarity to distinguish entity semantics. In order to capture entity semantics accurately, we could dynamically generate new semantic vectors for different contexts. This indicates the importance of the prior probability ($\mathcal{P}(t|h, r)$), which significantly improves the entity prediction performance.

Because the prior distribution of the missing element could be derived from the semantic similarity between the missing element and the others. For example, when we predict the head entity h in the triple (h, r, t) , the entities with more common or similar concepts belonging to the relation r and the tail entity t , have higher probability. Therefore, **TransC** utilizes these similarities to estimate the prior probability resulting in ranking similar entities higher.

Comparing the two approaches, multiple-vector representation (denoted as **TransC(Multiple.)**) and concept information (denoted as **TransC(Concept.)**), the later one is more instrumental for performance. While, **TransC** is better than both of them, demonstrating the necessity of the combination of multiple-vector representation and concept information.

4.3 Relation Prediction

We evaluated **TransC**’s performance on relationship prediction task using the FB15K dataset, following the experiment settings in [3].

Table 4. Evaluation results of relation prediction on FB15K.

	Mean rank		HITS10 (%)	
	Raw	Filter	Raw	Filter
TransE	2.91	2.53	69.5	90.2
TransH	8.25	7.91	60.3	72.5
TransR	2.49	2.09	70.2	91.6
DKRL	2.41	2.03	69.8	90.8
TKRL	2.47	2.07	68.3	90.6
SSP	1.87	1.47	70.9	90.9
TransT	1.59	1.19	72.0	94.1
TransC	1.37	1.38	73.6	95.2

We adopt the same protocol used in entity prediction. For relationship prediction, we replaced the relationship of each test triple with all relationships in the KG, with the concept domain sampling strategy, and rank these replacement relationships in descending order. This section utilizes Mean Rank and HITS@10 as evaluation metric.

The optimal-parameter configurations are described as follows: For dataset FB15K, the learning rate is 0.0005, the vector dimension is 150, the margin is 3, the scaling parameter β is 0.0001, concept similarity weights $\alpha_{head} = \alpha_{tail} = 1$, and $\alpha_{relation} = 0$. We train the model until convergence.

The overall entity prediction results on FB15K are reported in Table 4. The experimental results demonstrate that, the proposed **TransC** significantly outperforms all baselines in the most cases: (i) **TransC** achieves the best results on the “Raw” setting; (ii) compared with **TransT** and **SSP**, **TransC** improves the HIT@10 by 1.52% and 9.75%, on “Filter” setting. Compared with **TransT**, which utilized type information, **TransC** improves HITS@1 by 1.17% and Mean Rank by 4.73% in “Raw” setting. **TransT** performs better in relation prediction task than in entity prediction task. This is because, we use the concept-based semantics of both head entity and tail entity to generate relation’s semantic.

4.4 Triple Classification

Generally, the triple classification task could be reviewed as a binary classification task, which discriminate whether the given triple is correct or not. We utilize FB15K as the benchmark dataset, and utilize the binary classification accuracy as the evaluation metric. Moreover, We adopt the same strategy for negative samples generating used in [30].

The optimal-parameter configurations are described as follows: For dataset FB15K, the learning rate is 0.0005, the vector dimension is 100, the margin is 3, the scaling parameter β is 0.0001, concept similarity weights $\alpha_{head} = \alpha_{tail} = 1$, and $\alpha_{relation} = 0$. We train the model until convergence.

Table 5. Evaluation results of triple classification on FB15K.

	Accuracy (%)
TransE	85.7
TransH	87.7
TransR	86.4
DKRL	87.1
TKRL	88.5
SSP	90.1
TransT	91.0
TransC	93.4

Evaluation results on FB15K are shown in Table 5. **TransC** outperforms all baselines significantly. Compared with **TransT** and **SSP**, **TransC** improves the accuracy by 2.64% and 3.67%. We argue that, this phenomenon is rooted in the methodology that, the proposed **TransC** represents each entity as a set of concept vectors instead of a single vector, which adapting the rich entity semantics significantly and representing entities more accurately.

5 Conclusions

The paper studies aiming at combing the encyclopedic KG and the lexical KG for knowledge graph completion. The paper constructs multiple concepts of relations from entity concepts and designs the concept-based semantic similarity for multiple embedding representations and prior knowledge discovering. In summary, we leverage lexical knowledge base for knowledge graph completion task and propose TransC, a novel algorithm for KGC, which combines structured information in encyclopaedia KG and concept information in lexical KG (e.g., Probase). Different from using semantic-based linear transformations to separate the mixed representation, such as previous TransE, the proposed TransC models the multiple semantics separately and utilizes the semantic similarity to distinguish entity semantics. Empirically, we show the proposed algorithm could make full use of concept information and capture the rich semantic features of entities, and therefore improves the performance on two benchmark datasets over previous models of the same kind. Reasoning with temporal information in knowledge bases has a long history and has resulted in numerous temporal logics. Investigating the work on incorporating temporal information in knowledge graph completion methods, may become the future direction.

Acknowledgement. This work is funded by China Postdoctoral Science Foundation (No. 2018M641436), the Joint Advanced Research Foundation of China Electronics Technology Group Corporation (CETC) (No. 6141B08010102) and Joint Advanced Research Foundation of China Electronics Technology Group Corporation (CETC) (No. 6141B0801010a).

References

1. Bizer, C., et al.: DBpedia - a crystallization point for the Web of Data. *Web Semant. Sci. Serv. Agents World Wide Web* **7**(3), 154–165 (2009)
2. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: *SIGMOD Conference*, pp. 1247–1250 (2008)
3. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: *Advances in Neural Information Processing Systems*, pp. 2787–2795 (2013)
4. Bordes, A., Weston, J., Collobert, R., Bengio, Y.: Learning structured embeddings of knowledge bases. In: *AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, USA, August 2011* (2011)
5. Bordes, A., Weston, J., Usunier, N.: Open question answering with weakly supervised embedding models. In: *Calders, T., Esposito, F., Hüllermeier, E., Meo, R. (eds.) ECML PKDD 2014. LNCS (LNAI), vol. 8724*, pp. 165–180. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44848-9_11
6. Cornfield, J.: Bayes theorem. *Rev. Linstitut Int. Stat.* **35**(1), 34–49 (1967)
7. Cucerzan, S.: Large-scale named entity disambiguation based on Wikipedia data. In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2007, Prague, Czech Republic, 28–30 June 2007*, pp. 708–716 (2007)
8. Dong, X., et al.: Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In: *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 601–610 (2014)
9. Franz, T., Schultz, A., Sizov, S., Staab, S.: TripleRank: ranking semantic web data by tensor decomposition. In: *Bernstein, A., et al. (eds.) ISWC 2009. LNCS, vol. 5823*, pp. 213–228. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04930-9_14
10. Getoor, L., Taskar, B.: *Introduction to Statistical Relational Learning*. MIT Press, Cambridge (2007)
11. Huang, H., Liu, C.: Query evaluation on probabilistic RDF databases. In: *Vossen, G., Long, D.D.E., Yu, J.X. (eds.) WISE 2009. LNCS, vol. 5802*, pp. 307–320. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04409-0_32
12. Huang, H., Wang, Y., Feng, C., Liu, Z., Zhou, Q.: Leveraging conceptualization for short-text embedding. *IEEE Trans. Knowl. Data Eng.* **30**(7), 1282–1295 (2018)
13. Krompaß, D., Baier, S., Tresp, V.: Type-constrained representation learning in knowledge graphs. In: *Arenas, M., et al. (eds.) ISWC 2015. LNCS, vol. 9366*, pp. 640–655. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-25007-6_37
14. Lin, Y., Liu, Z., Luan, H.B., Sun, M., Rao, S., Liu, S.: Modeling relation paths for representation learning of knowledge bases. In: *EMNLP* (2015)
15. Lin, Y., Liu, Z., Zhu, X., Zhu, X., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: *Twenty-Ninth AAAI Conference on Artificial Intelligence*, pp. 2181–2187 (2015)
16. Long, T., Lowe, R., Cheung, J.C.K., Precup, D.: Leveraging lexical resources for learning entity embeddings in multi-relational data. *CoRR abs/1605.05416* (2016)
17. Ma, S., Ding, J., Jia, W., Wang, K., Guo, M.: TRANS: type-based multiple embedding representations for knowledge graph completion. In: *Ceci, M., Hollmén, J., Todorovski, L., Vens, C., Džeroski, S. (eds.) ECML PKDD 2017. LNCS (LNAI), vol. 10534*, pp. 717–733. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-71249-9_43

18. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: *Advances in Neural Information Processing Systems*, vol. 26, pp. 3111–3119 (2013)
19. Needell, D., Srebro, N., Ward, R.: Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm. *Math. Program.* **155**(1–2), 549–573 (2016)
20. Nguyen, D.Q., Sirts, K., Qu, L., Johnson, M.: STransE: a novel embedding model of entities and relationships in knowledge bases. In: *HLT-NAACL* (2016)
21. Nickel, M., Rosasco, L., Poggio, T.: Holographic embeddings of knowledge graphs. In: *Thirtieth AAAI Conference on Artificial Intelligence*, pp. 1955–1961 (2016)
22. Krompaß, D., Nickel, M., Tresp, V.: Querying factorized probabilistic triple databases. In: Mika, P., et al. (eds.) *ISWC 2014. LNCS*, vol. 8797, pp. 114–129. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11915-1_8
23. Nickel, M., Tresp, V., Kriegel, H.P.: A three-way model for collective learning on multi-relational data. In: *International Conference on International Conference on Machine Learning*, pp. 809–816 (2011)
24. Park, J.W., Hwang, S.W., Wang, H.: Fine-grained semantic conceptualization of FrameNet. In: *AAAI*, pp. 2638–2644 (2016)
25. Raedt, L.D., Kersting, K., Natarajan, S., Poole, D.: *Statistical relational artificial intelligence: logic, probability, and computation*, vol. 10, no. 2, pp. 1–189 (2016)
26. Richardson, M., Domingos, P.: Markov logic networks. *Mach. Learn.* **62**(1–2), 107–136 (2006)
27. Riedel, S., Yao, L., McCallum, A., Marlin, B.M.: Relation extraction with matrix factorization and universal schemas. In: *HLT-NAACL* (2013)
28. Schmidt, D.C.: *Learning probabilistic relational models* (2000)
29. Shi, B., Weninger, T.: Fact checking in heterogeneous information networks. In: *International Conference Companion on World Wide Web*, pp. 101–102 (2016)
30. Socher, R., Chen, D., Manning, C.D., Ng, A.Y.: Reasoning with neural tensor networks for knowledge base completion. In: *International Conference on Neural Information Processing Systems*, pp. 926–934 (2013)
31. Song, Y., Wang, H., Wang, Z., Li, H., Chen, W.: Short text conceptualization using a probabilistic knowledgebase. In: *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, vol. 3, pp. 2330–2336 (2011)
32. Song, Y., Wang, S., Wang, H.: Open domain short text conceptualization: a generative + descriptive modeling approach. In: *Proceedings of the 24th International Conference on Artificial Intelligence* (2015)
33. Teh, Y.W., Jordan, M.I., Beal, M.J., Blei, D.M.: Hierarchical dirichlet processes. *Am. Stat. Assoc.* **101**(476), 1566–1581 (2006)
34. Unger, C., Lehmann, J., Ngomo, A.C.N., Gerber, D., Cimiano, P.: Template-based question answering over RDF data. In: *International Conference on World Wide Web*, pp. 639–648 (2012)
35. Wang, Y., Huang, H., Feng, C.: Query expansion based on a feedback concept model for microblog retrieval. In: *International Conference on World Wide Web*, pp. 559–568 (2017)
36. Wang, Y., Huang, H., Feng, C., Zhou, Q., Gu, J., Gao, X.: CSE: conceptual sentence embeddings based on attention model. In: *54th Annual Meeting of the Association for Computational Linguistics*, pp. 505–515 (2016)
37. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: *Twenty-Eighth AAAI Conference on Artificial Intelligence*, pp. 1112–1119 (2014)

38. Wang, Z., Zhao, K., Wang, H., Meng, X., Wen, J.R.: Query understanding through knowledge-based conceptualization. In: International Conference on Artificial Intelligence, pp. 3264–3270 (2015)
39. Wu, W., Li, H., Wang, H., Zhu, K.Q.: Probase: a probabilistic taxonomy for text understanding. In: SIGMOD Conference (2012)
40. Xiao, H., Huang, M., Meng, L., Zhu, X.: SSP: semantic space projection for knowledge graph embedding with text descriptions. In: AAAI (2017)
41. Xiao, H., Huang, M., Zhu, X.: TransG: a generative model for knowledge graph embedding. In: Meeting of the Association for Computational Linguistics, pp. 2316–2325 (2016)
42. Xie, R., Liu, Z., Jia, J.J., Luan, H., Sun, M.: Representation learning of knowledge graphs with entity descriptions. In: AAAI (2016)
43. Xie, R., Liu, Z., Sun, M.: Representation learning of knowledge graphs with hierarchical types. In: International Joint Conference on Artificial Intelligence, pp. 2965–2971 (2016)
44. Yi, T., Luu, A.T., Hui, S.C.: Non-parametric estimation of multiple embeddings for link prediction on dynamic knowledge graphs. In: Thirty First Conference on Artificial Intelligence (2017)