Knowledge Graph Completion via Local Semantic Contexts

Xiangling Zhang[1], Cuilan Du[2], Peishan Li[1], and Yangxi Li[2](✉)

[1] School of Information, Renmin University of China, Beijing, China
{zhangxiangling,percentcent}@ruc.edu.cn
[2] National Computer Network Emergency Response Technical Team/Coordination Center of China (CNCERT/CC), Beijing, China
liyangxi@outlook.com

Abstract. Knowledge graphs are playing an increasingly important role for many search tasks such as entity search, question answering, etc. Although there are millions of entities and thousands of relations in many existing knowledge graphs such as Freebase and DBpedia, they are still far from complete. Previous approaches to complete knowledge graphs are either factor decomposition based methods or machine learning based ones. We propose a complementary approach that estimates the likelihood of a triple existing based on similarity measure of entities and some common semantic patterns of the entities. Such a way of triple estimation is very effective which exploits the semantic contexts of entities. Experimental results demonstrate that our model achieves significant improvements on knowledge graph completion compared with the state-of-art techniques.

Keywords: Knowledge graph completion · Entity semantic similarity · Knowledge graph

1 Introduction

In recent years, a number of large-scale knowledge graphs such as DBPedia [1], Freebase [2] and YAGO2 [22] have been created. Some (e.g., Googles Knowledge Graph and Microsoft Bings Satori) have been applied in search engines to support important search tasks such as entity search and question answering. Facts in those knowledge graphs are usually expressed in the form of triple $< subject, predicate, object >$ (denoted as $< s, p, o >$ in short). Although many of these open domain knowledge graphs are very huge in terms of massive entities and relations contained, they are still incomplete on both entities and their relations. For example, 75 % persons in Freebase lack nationality information [4]. This somehow affects the wide and effective applications of knowledge graphs. Therefore, the study on knowledge graph completion is important and necessary.

Knowledge graph completion can be generally described as to estimate the probability of a triple $< s, p, o >$ which do not appear in a knowledge graph,

S.B. Navathe et al. (Eds.): DASFAA 2016, Part I, LNCS 9642, pp. 432–446, 2016.
DOI: 10.1007/978-3-319-32025-0_27

given s, p, and o existing individually in the knowledge graph. Traditionally, people address this problem by building models to predict a triple using the whole facts of the knowledge graph. A number of tensor-based methods have been recently proposed [5,6,9,17,18] in which the knowledge graph is modeled as a tensor. Triple prediction is then achieved through the factorization of adjacent tensor. Existing triples in knowledge graphs are treated as positive examples, and those non-existing triples are treated as negative ones according to closed-world assumption. However, it is observed [9] that the closed-world assumption is inappropriate. Krompass et al. [9] therefore propose a local closed-world assumption which improves the prediction accuracy.

Another fold of approaches for knowledge graph completion is to convert both entities and relations of the knowledge graph into low-dimensional vectors [3,11,12,21,25]. Since TransE does not work well for 1-to-N, N-to-1 and N-to-N relations , other methods such as TransH [25], TransR [12] and PTransE [11] are also proposed in this stream of work. However, both tensor-based approaches and vector-based approaches rely on the training data that may be hard to achieve (for negatives). Moreover, the implicit nature of such approaches makes it hard to debug why they do not work well for entities and relations of particular domains.

In this paper, we propose a model to effectively estimate the likelihood of a triple $< s, p, o >$ according to the semantic contexts of s and o. The intuition of the proposed solution is "If s is similar with s' and $< s', p, o >$ exists, then it is likely that $< s, p, o >$ may also exist. Similarly, if o is similar with o' and $< s, p, o' >$ exists, then it is likely that $< s, p, o >$ may also exist." The problem is then how to effectively define the similarity between entities based on their semantic contexts, and how to estimate a triple based on the similarity measures of entities. The main contributions of the paper can be summarized as follows:

- We propose a similarity measure of two entities based on their semantic contexts.
- We design an effective model to estimate the likelihood of a missing triple.
- We conduct extensive experiments on two public datasets. The results show that the proposed model significantly outperforms the state-of-the-art techniques.

The rest of the paper is organized as follows: Sect. 2 gives a related work study. Section 3 introduces the solution. Experimental study is given in Sect. 4, followed by the conclusion given in Sect. 5.

2 Related Work

There are three typical representation models for knowledge graphs: graph based model, tensor based model and low-dimension vector model. The first one simply treat the whole knowledge graph as a graph. The intuition of our model is inspired by SimRank [8]. However, SimRank does not take semantic contexts into account when evaluating the similarities of entities. It merely computes the similarity score between two objects based on the topology structure of the graph. Moreover, the computational complexity of SimRank is quite expensive. Tong et al. [24] propose an efficient algorithm for SimRank. However, the relations among entities are still ignored. There are also some related studies considering the labels of edges/paths when evaluating the similarity of entities [7,13,23]. For example, Sun et al. [23] propose a measure named PathSim to evaluate the similarity between two entities under a certain semantic path. However, users should specify the paths in advance which is not suitable for knowledge graph with the abundant types of relations and entities. Another work PRA [10] employs relation paths for inference on knowledge graphs. It has beed applied in KnowledgeVault [4].

Tensor based model regard a knowledge graph as a tensor. A score for none existing triples in a given knowledge graph could be obtained through the tensor factorization algorithms in [5,6,9,17,18]. Take RESCAL [17] as an example, knowledge graph is modelled as a tensor. Two ways refer to the entities, the other way to the relations. When a triple $< s, p, o >$ holds in knowledge graph, the value corresponding to s, p, o is 1, and 0 otherwise. RESCAL decompose the tensor into a matrix and a low-dimension core tensor. They can be used to do link prediction. Unfortunately, the computational cost is too high and the memory requirements are also high especially for knowledge graph with millions of entities and thousands of relations.

Recently, with the development of representation learning, a lot of works [3,11,12,21,25] regard both entities and relations in knowledge graph as a low-dimensional vector. All these work are inspired by [14]. Take TransE [3] as an example. The learning score function of TransE is

$$f(s, p, o) = ||\mathbf{s} + \mathbf{p} - \mathbf{o}||_2^2 \tag{1}$$

To learn vector representation for the entities and relations in knowledge graph, TransE minimize $f(s, p, o)$ if the triple $< s, p, o >$ exists, and maximize otherwise. The relations could be classified into four classes. A given relation is 1-to-1 if a subject entity can map to at most one object, such as *capital*, *spouse*. 1-to-N means a subject could map to many objects. N-to-1 means many subjects could map to the same one object, such as *birthplace*. N-to-N means many subjects could map to many objects, such as *starring*. TransE works well to 1-to-1 relations but is not suitable for 1-to-N, N-to-1 and N-to-N relations. The reason is that replacing subject or object to generate negative examples is valid only for 1-to-1 relations.

Both tensor-based approaches and vector-based approaches need to construct a model using the whole graph, which becomes very expensive once the graph reaches certain size. What is more, some work such as [21] requires to build a model for each relation. Moreover, the interpretability of these methods are not strong since all the reasoning mechanism is implicit.

Another family of knowledge graph completion solutions are based on inductive logic programming. Various inductive logic programming methods such as

FOIL [19], Progol [16] and Claudien [20] can be applied. For example, FOIL learns Horn clauses which cover all positive examples but none negative ones. Due to the huge search space, the computational complexity is extremely high.

3 Our Model

3.1 The Basic Idea

An knowledge graph is a directed edge-labeled graph denoted as $\mathcal{K} = \{E, U, L, \tau\}$, where (1) E is an entity set, (2) $U \subseteq E \times E$ is a set of directed edges, (3) L is a set of edge labels (predicates), and (4) $\tau : U \rightarrow L$ is a mapping function defines the mappings from the edges to the labels. Each label represents a relation between two entities. For example, $\tau(s, e) \rightarrow l$ (also can be represented as an triple $< s, l, e >$), where $s \in E$, $e \in E$, $l \in L$, $< s, e > \in U$, means that there is a relation, the predicate l, between entities s and e. We use l^{-1} to represent the reverse relation of the label l. For example, a triple $< s, l, e >$ can also be represented as another triple $< e, l^{-1}, s >$.

To evaluate the likelihood of a triple $< s, p, o >$, the basic idea is to use local semantic contexts determined by $< s, p, o >$. Specifically, to evaluate whether s satisfies a pattern $< x, p, o >$, we will compute the similarity of entities satisfying pattern $< x, p, o >$ with the entity s. The higher the similarity, the more likelihood that the fact $< s, p, o >$ holds. Similarly, to evaluate whether o satisfies a pattern $< s, p, x >$, we will compute the similarity of entities satisfying pattern $< s, p, x >$ with the entity o. To be more specific, in the running example of Fig. 1, to estimate the probability of fact $< Catch\_Me\_If\_You\_Can, starring, Tom\_Hanks >$, we compare the similarities between $Catch\_Me\_If\_You\_Can$ and the other movies where Tom_Hanks played in. On the other way, we also compare Tom_Hanks with other actors played in $Catch\_Me\_If\_You\_Can$. The greater the similarity, the more likely Tom_Hanks played in $Catch\_Me\_If\_You\_Can$.

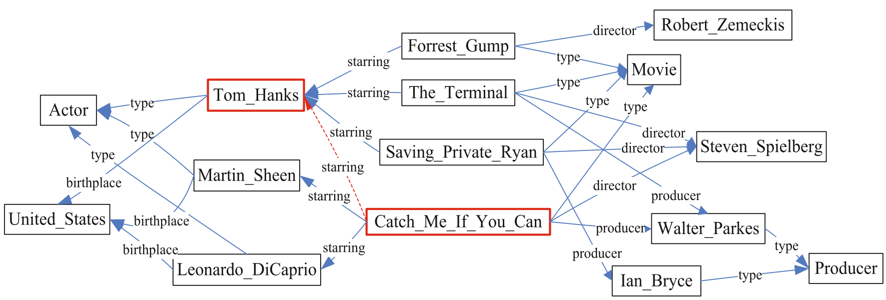

Fig. 1. A running example

To facilitate the introduction of our solution, we first give some relevant definitions as follows.

3.2 Definitions of Common Semantic Pattern

Given two entities, if they are similar, there must be a few of common features shared between them. For example, *Forrest_Gump* and *The_Terminal*, we say they are similar because they are films in which *Tom_Hanks* played. We apply the concept *semantic pattern* to define such common features.

Definition 1 (Semantic Pattern). *A semantic pattern in a knowledge graph* $\mathcal{K}$ *is composed of an anchor entity node* e_a*, and a predicate* p*. It is denoted as* $\pi = e_a : p$*.*

A semantic pattern (SP) is used to represent a set of target entities (defined in the Definition 2) having the relation p with the same anchor entity e_a. For example, to express the movies where Tom_Hanks played a role, we can utilize the semantic pattern $\pi_1 = Tom\_Hanks : starring^{-1}$, where Tom_Hanks is the anchor entity, and the predicate $starring^{-1}$ is the relation between the target entities of π_1 and the anchor entity Tom_Hanks. Note that $^{-1}$ indicates the direction of relation (predicate) where the anchor entity serves as an object. Another semantic pattern $\pi_2 = Movie : type^{-1}$ means a set of entities whose *type* is $Movie$. A semantic pattern exhibits the common feature of the target entities.

Definition 2 (Target entity). *If an entity* e *has a relation* p *with the anchor entity* e_a*. We say* e *is a target entity of* $\pi = e_a : p$ *which is denoted as* $e \models \pi$*.*

The set of target entities of a semantic pattern $\pi = e_a : p$ is denoted as $E(\pi) = \{e|e \models \pi\}$. For example, the set $E(\pi_1)$=$\{Catch\_Me\_If\_You\_Can$, $Saving\_Private\_Ryan$, $Forrest\_Gump\}$ in the running example of Fig. 1.

Actually, each entity in knowledge graphs may satisfy many semantic patterns. For example, in the Fig. 1, $Tom\_Hanks \models \pi_3$, $Tom\_Hanks \models \pi_4$, $Tom\_Hanks \models \pi_5$, $Tom\_Hanks \models \pi_6$, $Tom\_Hanks \models \pi_7$ where $\pi_3 = Forrest\_Gump : starring, \pi_4 = Actor : type^{-1}, \pi_5 = United\_States : birthplace^{-1}, \pi_6 = The\_Terminal : starring, \pi_7 = Saving\_Private\_Ryan : starring$. We use $\Phi(e)$ to represent the set of semantic patterns where the entity e satisifies and $\Phi(Tom\_Hanks) = \{\pi_3, \pi_4, \pi_5, \pi_6, \pi_7\}$.

If two entities satisfy the same semantic pattern, they will be similar with each other to some extent. However, if they do not have any common semantic pattern, they may also be similar with each other. For example, $Steven\_Spielberg$ and $Robert\_Zemeckis$ are similar because $Robert\_Zemeckis$ directed $Forrest\_Gump$ and $Steven\_Spielberg$ directed $The\_Terminal$, and the two movies have the same actor Tom_Hanks. That is to say, $Forrest\_Gump$ and $The\_Terminal$ are similar and the similarity propagate to $Steven\_Spielberg$ and $Robert\_Zemeckis$.

With the above definitions, an important issue is then how to define the similarity between entities, based on the semantic contexts determined by common semantic patterns.

To facilitate the understanding of the concepts and solution, we show some frequently used notations in Table 1.

Table 1. Frequently used notations

Notation	Description
$\mathcal{K} = \{E, U, L, \tau\}$	A knowledge graph
$\pi = e_a : p$	A semantic pattern where e_a is an anchor entity and p is a (directed) predicate
$E(\pi) = \{e \mid e \models \pi\}$	The set of target entities satisfying the semantic pattern π
$\Phi(e)$	The set of semantic patterns where the entity e satisifies
$P(e)$	The set of predicates of the entity e
$sim_s(a, b)$	The direct similarity score of two entities determined by SP
$sim_p(a, b)$	The propagated similarity score of two entities

3.3 Similarity of Entities

The semantic similarity of two entities consists of two parts: one is acquired by their common semantic patterns denoted as $sim_s(a, b)$; and the other is propagated through the same relation from the similar entities denoted as $sim_p(a, b)$.

With respect to $sim_s(a, b)$, the more common semantic patterns shared by entity a and b, the more similarity they will have. However, the weight of each common semantic pattern is different. Borrowing the idea of inverse document frequency in information retrieval, we term it as inverse semantic pattern frequency. The idea is that general semantic patterns are not as useful as non-frequent semantic patterns in computing the similarities of entities. We define the weight of an semantic pattern π as $\frac{1}{log(1+|E(\pi)|)}$ where $|E(\pi)|$ is the number of target entities of π. Note that for each semantic patterns π in the knowledge graphs, $|E(\pi)| > 1$.

$$sim_s(a, b) = \frac{\sum\limits_{\pi \in (\Phi(a) \cap \Phi(b))} \frac{1}{log(1+|E(\pi)|)}}{\sqrt{|\Phi(a)||\Phi(b)|}} \tag{2}$$

Besides the direct similarity determined by local semantic contexts of entities, we need also measure the propagated similarity of entities, which is defined as:

$$sim_p(a, b) = \sum_{p \in (P(a) \cap P(b))} \frac{\sum\limits_{a' \models a:p} \sum\limits_{b' \models b:p} sim_s(a', b')}{|E(a : p)||E(b : p)|} \tag{3}$$

where $P(a)$ and $P(b)$ are sets of predicates of a and b respectively. According to the definition, the propagated similarity is aggregated from pairs of entities that share the same predicate to the two entities a and b.

Finally, the similarity between two entities is evaluated as the weighted sum of the two parts:

$$sim(a, b) = \alpha \cdot sim_s(a, b) + (1 - \alpha) \cdot sim_p(a, b) \tag{4}$$

where α is the parameter tuning the weight.

3.4 The Overall Solution

Our goal is to obtain the likelihood of a triple $< s, p, o >$. The intuition of our method is to calculate the similarity between entity s and other entities s' who has a relation p with o, which means $s' \models o : p^{-1}$. We also take into account of the similarity between o and o' which has a relation p with s, which means $o' \models s : p$. Finally, we define the probability from two directions of a triple, by considering the likelihood of $s \models o : p^{-1}$, as well as that of $o \models s : p$. For evaluating the probability score of a triple $< s, p, o >$, we optimistically choose the maximal similarity score computed from the two directions.

$$score(< s, p, o >) = \beta \cdot \max_{o' \in E(s:p)} sim(o', o) + (1 - \beta) \cdot \max_{s' \in E(o:p^{-1})} sim(s', s) \quad (5)$$

which is also a weighted sum (determined by the parameter β) of the two parts. Note that the likelihood of a triplet is not normalized, which does not affect the effectiveness of triple prediction because it is a relative measure. We denote our method as LSCS for short.

4 Experiments

Our model is evaluated on two widely used knowledge graphs: WordNet [15] and Freebase [2]. We adopt three datasets (their statistics are given in Table 2) and conduct three tasks to evaluate our model.

Table 2. Statistics of the data sets

Dataset	#Relation	#Entities	#Train	#Test
FB15K	1,345	14,951	483,142	59,071
FB13	13	75,043	316,232	23,733
WN11	11	38,696	112,581	10,544

4.1 Experimental Setup

Data Sets Description

Wordnet. This knowledge graph can be seen as a combination of dictionary and thesaurus. The entities (called synsets) correspond to word senses, and the relations between entities represent lexical relations between them, such as hypernym, hyponym and meronym. We utilize WN11 used in [12,21] which contains 11 relation types.

Freebase. Freebase is a large and growing collaborative knowledge base which provides general facts of the world. There are currently around 3.1

billion facts (triplets) and more than 80 million entities. For instance, $< albert\_einstein, spouse, mileva\_maric >$ means the relation between entity *albert_einstein* and entity *mileva_maric* is *spouse*. We apply FB13 used in [12,21] and FB15K used in [3,11,12] as two data sets. For FB13, all the subjects are from people domain and 13 relations are extracted while only 7 appear in the testing data.

Baselines

We apply 5 baselines for comparison with our model. Among them, we use the code provided by the authors for TransR, TransE and PTransE. All of them consider the knowledge graph as a continuous vector space meanwhile the relations and entities are transformed into low-dimension vectors. We utilize the best configuration supplied by the paper [3,11,12]. Simultaneously, we compare with other entity similarity model including SimRank [8] and PathSim [23]. For SimRank we use a decay factor $C = 0.8$ and for PathSim we explore all possible paths.

Metrics

The metrics adopted for evaluation include: accuracy for the triple classification task, the mean rank and hits@10 for entity prediction. We consider hits@1 for relation prediction since hits@10 for approaches exceeds 95%. All these metrics are also used in [3,11,12,21].

For each test triplet, the subject of the triplet is replaced by other entities with predicate in triplet and rank these entities in descending order of scores calculated by score function in Eq. 5. We also apply the same procedure for the object of the triplet. We exhibit the mean of those predicted ranks and the hits@10 which is the proportion of correct entities ranked in the top 10. Country to expectation, the above metrics may under-estimate when some triplets already exist in the knowledge graph. In this case, before ranking we may filter out those triplets because they are true. We denote the first setting as “Raw” and the latter one as “Filter”

4.2 Experimental Results

Triple Classification

Our goal of this task is to choose the correct triple in the form of (s, p, t) in the testing set. In the testing sets of WN11 and FB13, there are triple pairs with the same subject and predicate and different objects. For each pair, the first one is positive while the other is negative. This is a binary classification task which has been investigated in [3,12,25]. All algorithms compute a score for each triple, the score could be used to determine the likelihood of each possible triple. For each pair, if the first one’s score is greater, then we deem it as the positive one. The metric adopted for evaluation is *accuracy*.

Evaluation results on both WN11 and FB13 are displayed in Table 3. From the table we observe that: (1) Overall, the results on WN11 is better than FB13.

Table 3. Evaluation results of triple classification(%)

Dataset	WN11	FB13
TransE	80.24	81.09
PTransE	79.24	73.32
TransR	80.12	80.20
SimRank	98.86	82.65
PathSim	98.99	80.65
LSCS	**99.02**	**88.17**

This is because of the characteristics of data sets. Since all the subjects in FB13 are from *people* domain, we can not obtain the similarity between objects from other contexts. For instance, $Paris$ and $Marseilles$ are similar because both of them are the cities of $France$. Unfortunately, we could not obtain this information in FB13. In contrast, the knowledge is more comprehensive in WN11. (2) None of TransE, TransR and PTransE can outperform the other kind of methods based on entity similarity such as Pathsim on the WN11 data set. The reason behind the phenomenon maybe that methods based on entity similarity only consider the local knowledge of a triple are more precise than the model learned from the whole knowledge graph. (3) On FB13, our model significantly outperforms baseline methods. There are two reasons for this result. Firstly, local knowledge of a triple is more powerful. Secondly, our entity semantic similarity metric is more effective than SimRank and Pathsim. We not only take into account their common semantic patterns, but consider their different weights.

Figure 2 shows the accuracy of different relations on FB13 of our model. The accuracy ranges from 70.86 % ($gender$) to 97.94 % ($nationality$). We observe that using local knowledge about subjects and objects based on semantic context similarity is difficult to predict the $gender$ even though there are only two values for $gender$. It is difficult to infer the $gender$ from a person's local information (such as $nationality$, $cause\_of\_death$, $profession$, $ethnicity$) in intuition.

Table 4. Results of entity prediction

Metric	Mean rank		Hits@10(%)	
	Raw	Filter	Raw	Filter
TransE	147	48	52.80	74.40
PTransE	168	18	53.29	91.44
TransR	178	81	52.14	79.72
SimRank	**114**	90	55.29	70.35
PathSim	157	90	51.59	77.96
LSCS	154	**16**	**59.74**	**92.21**

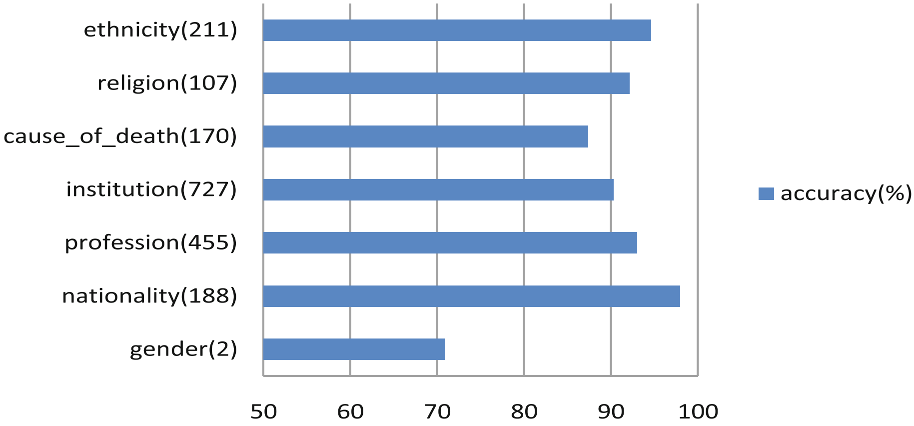

Fig. 2. Comparison of accuracy of different relations on FB13. The number in the bracket means the size of possible answer set

Entity Prediction

The goal of entity prediction is to predict the missing subject or object for a triple $< s, p, o >$ which employed in [3,11,12]. This task requires a set of candidate entities from the knowledge graph for the position of subject or object which is missing rather than only giving one best result.

In this paper, we utilize the FB15K dataset for this task because of the abundant relations contained by FB15K. In the testing phase, for each test triple $< s, p, o >$, we replace the subject and object entity by those which have the predicate p but not all entities in the knowledge graph. For example, to predict $< Catch\_Me\_If\_You\_Can, starring, Tom\_Hanks >$, we replace the subject $Catch\_Me\_If\_You\_Can$ with those subject entities who have a predicate $starring$, such as, $Saving\_Private\_Ryan, The\_Terminal, Forrest\_Gump$, we do not use such as $United\_States$ to replace $Catch\_Me\_If\_You\_Can$ because $United\_States$ does not have a relation $starring$ with any entity. Rank all generated triples in descending order by the prediction scores. We follow [3,11,12] and use two measures as our evaluation metric: mean rank and hits@10. As described in [3], the metrics are affected by the triples in knowledge graph. In other word, we also calculate a score for triples that already exist in the knowledge graph. The metrics will be under-estimated. Therefore, we filter out all those triples already hold in knowledge graph before rank and named as "Filter". The other one is "Raw". We calculate the metrics individually.

Table 4 shows the results of entity prediction. According to the results, we can observe that our model outperforms the other baselines. It implies that local information provides adequate evidences to infer a triple while the semantic context similarity is also effective.

Relation Prediction

Relation prediction aims to predict the relation between two entities. We also conduct experiment on the FB15K dataset for evaluation. We replace all of the predicates in the knowledge graph for the given two entities and then rank them. We adopt mean rank and hits@1 as measures to compare the algorithms. Experimental results of relation prediction are shown in Table 5.

Table 5. Results of relation prediction

Metric	Mean rank		Hits@1(%)	
	Raw	Filter	Raw	Filter
TransE	85	85	47.82	58.60
PTransE	2.38	2.13	68.42	**93.92**
TransR	101	100	34.71	42.35
SimRank	28	7	49.48	53.27
PathSim	31	10	52.77	59.54
LSCS	**1.86**	**1.77**	**70.42**	90.31

Generally, relation prediction is easier than entity prediction because the number of possible relations is less. Compared with the results of entity prediction in Table 4, the results of relation prediction in Table 5 are better. From the result we observe that: both PTransE and our model performs the best. Because the local knowledge plays an import role for prediction and the entities similarity metric also works of our model. Meanwhile the performance of PTransE is also very good. That is because PTransE takes into account different paths play a role for triple prediction.

Parameters Impact of LSCS

LSCS has two parameters that may affect its performance on knowledge graph completion. One is the parameter α used to balance the weight of common semantic patterns and the similarity conveyed from neighbors. Intuitively, the contribution of common semantic pattern is even greater. The other is the parameter β used to represent different weight of two directions which means one direction is to compare s with s' when $< s', p, o >$ exists in knowledge graph, the other is to compare o and o' when $< s, p, o' >$ holds in knowledge graph. We test the impacts of these two parameters on triple classification task on both WN11 and FB13 dataset. According to the results in Figs. 3 and 4, we may observe that the parameter α and β affects the performance on the two datasets differently. This is also reasonable because the characteristics are different for the two datasets.

In Fig. 3, when $\alpha = 0.95$ the performance reaches the best on FB13. When $\alpha = 1$, we notice that the performance is worse than $\alpha = 0.95$. This prove that

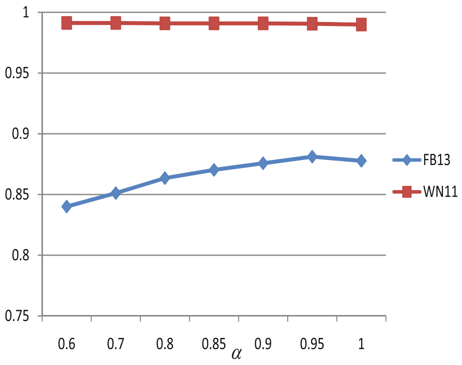

Fig. 3. The impacts of the parameter α

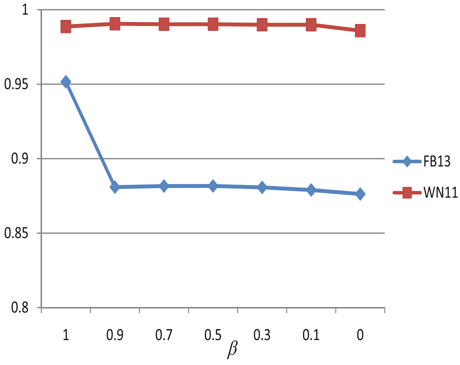

Fig. 4. The impacts of the parameter β

the similarity conveyed from the similar neighbors also plays a little role. For WN11 Dataset, the value of α has little influence on the performance.

From the Fig. 4, we observe that the larger of β the performance is better for FB13 dataset. Since the subject entities in FB13 are only from *people* domain, and all the predicates are about *people*, such as $gender, nationality$, $cause\_of\_death$. The contribution of similarity between s and s' when $< s', p, o >$ holds is more important. For WN11 dataset, the value of β has little influence on the performance. Generally speaking, two directions are essential to the performance while the direction weight distribution has little effect. To sum up, the weight of two directions is equal, that is $\beta = 0.5$.

4.3 Example of Reasoning

We have observed that our model achieves good performance for knowledge graph completion through the experiments. In this part, we exhibit one example of our model.

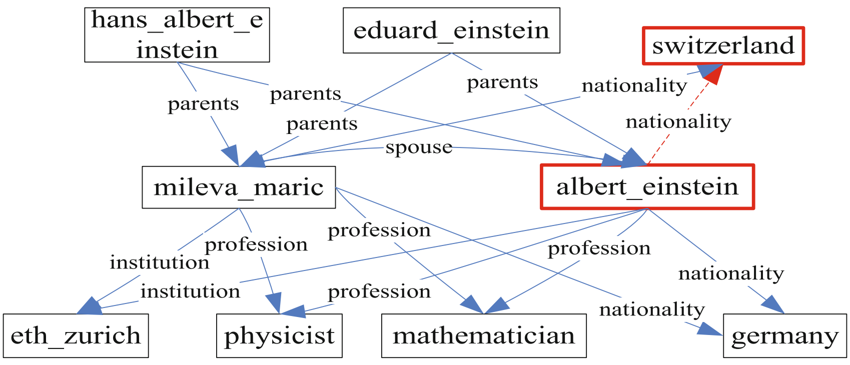

Fig. 5. A reasoning example in FB13

As shown in Fig. 5, to predict $< Albert\_einstein, nationality, Switzerland >$, we notice that *mileva_maric* whose *nationality* is *switzerland* has a strong similarity with *albert_einstein* which increase the likelihood of the triple. We can regard *mileva_maric* as an evidence to infer the likelihood of the triple $< Albert\_einstein, nationality, Switzerland >$ exists in knowledge graph. The more similar between *albert_einstein* and *mileva_maric*, the more likelihood of the triple.

5 Conclusion

In this paper, we propose an approach to complete knowledge graphs, which only considers the local knowledge related to the subjects and objects in the knowledge graph. Unlike existing methods, our model is based on the similarity determined by the semantic contexts between entities to infer the likelihood of a triple. In experiments, we evaluate our models on three tasks including triple classification, entity prediction and relation prediction. Experiment results demonstrate that our model achieves significant improvements compared to TransE, TransR and PTransE as well as other entity similarity measures such as SimRank and Pathsim.

Acknowledgements. This work is supported by National Basic Research Program of China (973 Program) No. 2012CB316205, the National Science Foundation of China under grant (No. 61472426, 61170010, 61432006), the Fundamental Research Funds for the Central Universities, the Research Funds of Renmin University of China No. 14XNLQ06, and a gift by Tencent.

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.G.: DBpedia: a nucleus for a web of open data. In: Aberer, K., et al. (eds.) ASWC 2007 and ISWC 2007. LNCS, vol. 4825, pp. 722–735. Springer, Heidelberg (2007)
2. Bollacker, K.D., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, 10–12 June 2008, pp. 1247–1250 (2008)
3. Bordes, A., Usunier, N., García-Durán, A., Weston, J., Yakhnenko, O.: 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 5–8 December 2013, Lake Tahoe, Nevada, United States, pp. 2787–2795 (2013)
4. Dong, X., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmann, T., Sun, S., Zhang, W.: Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In: The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2014, New York, NY, USA, 24–27 August 2014, pp. 601–610 (2014)

5. Drumond, L., Rendle, S., Schmidt-Thieme, L.: Predicting RDF triples in incomplete knowledge bases with tensor factorization. In: Proceedings of the ACM Symposium on Applied Computing, SAC 2012, Riva, Trento, Italy, 26–30 March 2012, pp. 326–331 (2012)
6. Franz, T., Schultz, A., Sizov, S., Staab, S.: TripleRank: ranking semantic web data by tensor decomposition. In: Bernstein, A., Karger, D.R., Heath, T., Feigenbaum, L., Maynard, D., Motta, E., Thirunarayan, K. (eds.) ISWC 2009. LNCS, vol. 5823, pp. 213–228. Springer, Heidelberg (2009)
7. Gangemi, A., Presutti, V.: Towards a pattern science for the semantic web. Semantic Web **1**(1–2), 61–68 (2010)
8. Jeh, G., Widom, J.: Simrank: a measure of structural-context similarity. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 23–26 July 2002, Edmonton, Alberta, Canada, pp. 538–543 (2002)
9. Krompass, D., Nickel, M., Tresp, V.: Large-scale factorization of type-constrained multi-relational data. In: International Conference on Data Science and Advanced Analytics, DSAA 2014, Shanghai, China, 30 October–1 November 2014, pp. 18–24 (2014)
10. Lao, N., Mitchell, T.M., Cohen, W.W.: Random walk inference and learning in a large scale knowledge base. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, 27–31 July 2011, John McIntyre Conference Centre, Edinburgh, UK, A meeting of SIGDAT, a Special Interest Group of the ACL, pp. 529–539 (2011)
11. Lin, Y., Liu, Z., Luan, H., Sun, M., Rao, S., Liu, S.: Modeling relation paths for representation learning of knowledge bases. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, 17–21 September 2015, pp. 705–714(2015)
12. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 25–30 January 2015, Austin, Texas, USA, pp. 2181–2187 (2015)
13. Maccatrozzo, V., Ceolin, D., Aroyo, L., Groth, P.: A semantic pattern-based recommender. In: Presutti, V., et al. (eds.) SemWebEval 2014. CCIS, vol. 475, pp. 182–187. Springer, Heidelberg (2014)
14. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. 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, pp. 3111–3119 (2013)
15. Miller, G.A.: Wordnet: a lexical database for English. Commun. ACM **38**(11), 39–41 (1995)
16. Muggleton, S.: Inverse entailment and progol. New Gener. Comput. **13**(3&4), 245–286 (1995)
17. Nickel, M., Tresp, V., Kriegel, H.: A three-way model for collective learning on multi-relational data. In: Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, 28 June–2 July 2011, pp. 809–816 (2011)
18. Nickel, M., Tresp, V., Kriegel, H.: Factorizing YAGO: scalable machine learning for linked data. In: Proceedings of the 21st World Wide Web Conference 2012, WWW 2012, Lyon, France, 16–20 April 2012, pp. 271–280 (2012)

19. Quinlan, J.R., Cameron-Jones, R.M.: FOIL: a midterm report. In: Machine Learning: ECML-93, European Conference on Machine Learning, Vienna, Austria, 5–7 April 1993, pp. 3–20 (1993)
20. Raedt, L.D., Dehaspe, L.: Clausal discovery. Mach. Learn. **26**(2–3), 99–146 (1997)
21. Socher, R., Chen, D., Manning, C.D., Ng, A.Y.: Reasoning with neural tensor networks for knowledge base completion. In: Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held 5–8 December 2013, Lake Tahoe, Nevada, United States, pp. 926–934 (2013)
22. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, 8–12 May 2007, pp. 697–706 (2007)
23. Sun, Y., Han, J., Yan, X., Yu, P.S., Wu, T.: Pathsim: meta path-based top-k similarity search in heterogeneous information networks. PVLDB **4**(11), 992–1003 (2011)
24. Tong, H., Faloutsos, C., Pan, J.: Fast random walk with restart and its applications. In: Proceedings of the 6th IEEE International Conference on Data Mining (ICDM 2006), 18–22 December 2006, Hong Kong, China, pp. 613–622 (2006)
25. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, 27–31 July 2014, Québec City, Québec, Canada, pp. 1112–1119 (2014)