Computing and Informatics, Vol. 38, 2019, 1403–1417, doi: 10.31577/cai_2019_5_1403

FUZZY KNOWLEDGE INFERENCE: QUICKLY ESTIMATE EVIDENCE VIA FORMULA EMBEDDING

Xiao Zhang

School of Mechanical Electronic and Information Engineering
China University of Mining and Technology, Beijing
No. 11 College Road, Haidian District, Beijing China
e-mail: zhangxiao@cumtb.edu.cn

Abstract. Inference on Knowledge Bases (KBs) is an important way to construct more complete KBs and answer KB questions. Inference can be viewed as a process from evidence to conclusion following specific formulas. Traditional methods usually search on the KB to collect evidence, which cannot apply to large-scale KBs, because the running time of searching increases radically as the scale of KBs increases. What is worse, evidence cannot be found if one fact in it is missing, which may result in the failure of inference. To this end, we propose a fuzzy method of estimating evidence, which replaces searching by estimating the existence of evidence by constructing formula embeddings, and then we merge these estimations into a probabilistic model to infer conclusions. This method can apply to large-scale KBs, because estimating evidence is very fast and is irrelevant to the KB scale. Estimating evidence can also be viewed as fuzzy matching, so this method can handle the situation where facts are missing. We evaluate this method on the knowledge base completion task, and it achieves a better performance than state-of-the-art methods and has a shorter running time.

Keywords: Fuzzy logic, fast inference, knowledge base completion, formula mining

1 INTRODUCTION

The inference is a process from evidence to a conclusion. A typical inference on KBs is usually to predict the missing element in a tuple, e.g., *Relation (Head Entity, ?)* or *Relation (?, Tail Entity)*, so the inference is an important way to complete KBs and answer KB questions. If we view a KB as direct graphs (Figure 1 a)), inference essentially means to take usage of the graph structures to predict missing links, e.g. *Nationality(Cristiano Ronaldo*[1]*, ?).* To each specific type of queries, we can extract several frequent structures from graphs as a priori knowledge, and they are called formulas. For example, for *Nationality* queries, $\mathit{Father}(x_1, x_2) \wedge \mathit{Nationality}(x_2, x_3)$ is a frequent structure which has a probability of supporting $\mathit{Nationality}(x_1, x_3)$, and it is called as a formula, where x_i denotes an entity variable. Some other formulas are shown in Figure 1 b).

Traditional inference methods usually search on the KB to collect formula instances as evidence, e.g., we can find *Cristiano Ronaldo* $\xrightarrow{Father}$ *Jose Dinis Aveiro* $\xrightarrow{Nationality}$ *Portugal*, to support *Nationality(Cristiano Ronaldo, Portugal).* Searching evidence cannot apply to large-scale KBs, because its computation complexity is $O(n^l)$, where n is the average degree of nodes and l is the maximal length of formula, and it may take a long time when the graph is large or dense. Another drawback of searching is that its matching condition is too strict. When one or two facts of evidence are missing in the KB, the evidence cannot be found by searching method, which may result in incorrect inference result. For example, *Son(Cristiano Ronaldo, Santos)* is missing in Figure 1 a), so the evidence, *Cristiano Ronaldo* $\xrightarrow{Son}$ *Santos* $\xrightarrow{Nationality}$ *Portugal*, cannot be found by searching, and further result in the invalidation of the formula, $\mathit{Son}(x_1, x_2) \wedge \mathit{Nationality}(x_2, x_3) \Rightarrow \mathit{Nationality}(x_1, x_3)$.

To accelerate the inference computation process, knowledge graph embedding methods are used in inference on KBs, such as TransE [3], TransH [15] and TransR [10]. TransE is translating embedding. TransH is translating embedding on hyperplanes. Trans means performing translation in relation-specific entity space. PTransE is translating embedding for relation paths. Their basic intuition is to represent entities and relations in KBs as low-rank real vectors, and almost all of them expect that entities in one fact are close in a space specific to the relation. However, these embedding-based methods lack the explicit logic constrains and enough evidence, which makes these methods to be more like modeling KBs than performing inference. Therefore, they are prone to be damaged by unexpected noises and lead to unsatisfying results. Compositional training (COMP) [6] and PTransE [9] have a preliminary attempt at merging paths into KB embedding model, but they have not solved the above problems. COMP uses paths to improve learning KB embeddings rather than to infer missing facts, so its performance is still unsatisfied. PTransE still needs to search paths on KBs, so its running time is as long as the traditional inference methods.

We consider that collecting evidence to infer the conclusion is a good tradition, but searching on KBs is not necessary. KB embeddings provide a possibility to estimate evidence without searching. We exploit the embedding strategy and propose an approach which can realize quick fuzzy inference. This method represents

[1] Cristiano Ronaldo is a Portuguese professional footballer of the Juventus F.C.

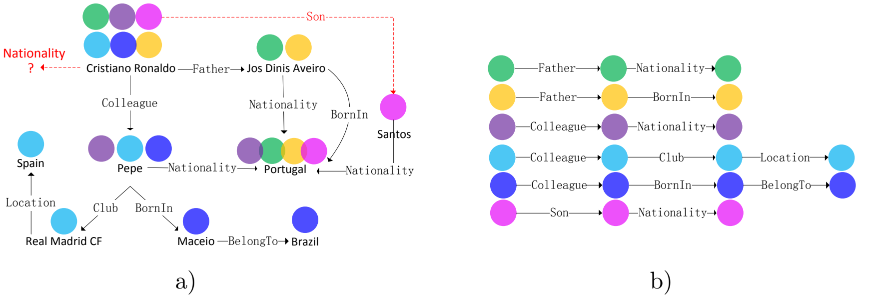


Figure 1. An example of inference on the KB: a) is the knowledge graph, and the query is what is the nationality of Cristiano Ronaldo, where Son(Cristiano Ronaldo, Santos) is a missing fact; b) is a set of formula operators, which are embedded and used to estimate whether this evidence is occurring between Cristiano Ronaldo and one nation.

formulas as computable operators by using pre-trained KB embeddings, and these formula operators are used to measure the distances from holding evidence. For example, for the formula $Father(x_1, x_2) \wedge Nationality(x_2, x_3) \Rightarrow Nationality(x_1, x_3)$, we use embeddings of *Father* and *Nationality* to represent the formula operator, and then it is applied on *Cristiano Ronaldo* and *Portugal* to estimate the existence of the evidence, *Cristiano Ronaldo* $\xrightarrow{Father}$ *Somebody* $\xrightarrow{Nationality}$ *Portugal*. This method treats the evidence as a whole and only cares about whether it exists. After that, we merge the probabilities of evidences obtained from estimation into the MLN framework and infer the probability of *Nationality(Cristiano Ronaldo, Portugal)* is true. To realize evidence fuzzy matching and speed up the estimation, we ignore middle variables, e.g., the middle variable, *Jose Dinis Aveiro*, is replaced with *Somebody*. Therefore, this method is insensitive to missing facts and can find evidence under difficult situations where facts are missing in the evidence. When estimate one evidence, the computation complexity of this method is $O(1)$, and paths with any length can be estimated by one step simple calculation. Therefore, our approach can apply to large-scale KBs.

2 RELATED WORK

In general, according to the process of inference on KB, there are three types of approaches:

1. probabilistic logic inference;
2. knowledge graph embedding;
3. formula embedding.

Especially, the third type can be viewed as the combination of logic and embedding.

2.1 Probabilistic Logic Inference

This type of methods focused on mining frequent substructure on the KB as evidence, and it models the relationship between evidence and the query in different ways. MLN [14], PSL [5] and PRA [8] are all typical probabilistic logic models. These methods need searching or performing random walks to collecting evidence, and then they use a probabilistic model to combine evidence to infer conclusions. We take MLN as the example to explain, and its manner of combining evidence is similar to our approach.

Markov Logic Network (MLN) can be viewed as a probabilistic extension of first-order logic by attaching weights to formulas. Especially, for its discriminative version, higher weight indicates greater reward to the query that satisfies the formula. To predict the query Y, it mines a set of formulas F_Y and counts these formula groundings on the KB, which is the typical practice of search-based inferences. Then MLN calculates Y's conditional probability as follows:

$$P_w(Y = y|X = x) = \frac{1}{Z_y}\exp\left(\sum_{f_i \in F_Y} w_i n_i(x, y)\right) \tag{1}$$

where $n_i(x, y)$ denotes the number of true groundings of formula f_i and w_i is f_i's weight. $Z_y = \sum_{y' \in Y} exp(\sum_{f_i \in F} w_i n_i(x, y'))$ is the normalizing term.

2.2 Knowledge Graph Embedding

Many knowledge graph embedding models are proposed in recent years, such as RESCAL [13], SE [4], SME[2], LFM [7], TransE [3], TransH [15], TransR [10]. Almost all embedding-based models embed entities into a relatively low (e.g., 50) dimensional embedding vector space R^k while representing relations in different ways. These models expect that correlative elements of the KB are close in the embedding space, so they can be used to complete KBs. To predict the query $r(h, t)$, most of the embedding-based models calculate the similarity between E_h and E_t under E_r, noted as $f_s(E_h, E_t, E_r)$, where E_h, E_t and E_r represent the embeddings of h, t and r, respectively. This type of methods can be used to learning KB embeddings before estimate evidence in our approach, but they do not represent explicit evidence. We employ TransE as an example to explain how the embedding model performs inference, and we use TransE to learn KB embeddings in our experiments.

TransE represents relations as translations in the entity vector space R^k, and assumes that if $r(h, t)$ holds, then the embedding of the tail entity t should be close to the embedding of the head entity h plus some vector that depends on the relationship r. TransE's similarity function is:

$$f_{r(h,t)} = -\|E_h + E_r - E_t\|_2^2. \tag{2}$$

TransE employs a margin-based ranking loss to learn the KB embeddings. TransE assumes that $f_{r(h,t)}$ is larger than other $f_{r(h,t')}$ or $f_{r(h',t)}$, where $r(h,t)$ exists in the KB but $r(h,t^{'})$ or $r(h^{'},t)$ does not and designs its loss function $\mathcal{L}$ as follows:

$$\mathcal{L} = \sum_{f_s \in KB} \sum_{f_s' \in KB'} [\gamma + f_s^{'} - f_s]_+ \quad (3)$$

where KB and $KB^{'}$ represent a true fact set and a false fact set according the KB, respectively. The symbol $[\ldots]_+$ means the final results always bigger than 0. When the number in [] is less than zero, the final result is 0. When the number in [] is bigger than 0, the final result is the number.

2.3 Formula Embedding

Embedding formulas is latest research subject, and researchers focus on combining logic and embedding which is also our purpose. Compositional training (COMP) [6], PTransE [9], SePLi [17] and RNN model [12] are all representative models, and they are related to our method of embedding paths.

COMP wants to learn KB embeddings by using facts and paths, simultaneously. COMP represents paths in two ways:

1. RESCAL based: a path is represented as a matrix, which equals the produce of relations in it;
2. TransE based: a path is represented as a vector, which equals the sum of relations in it.

COMP constructs a dataset of paths by performing random walks on the KB, and then the path dataset and the original KB are used to learn KB embeddings.

PTransE has a similar method to represent paths, which is also based on TransE. PTransE designs its loss function by combining both original query loss and path loss, and designs its loss function as follows:

$$L = L(h,r,t) + \frac{1}{Z} \sum_{p \in P(h,t)} R(p|h,t) L(p,r) \quad (4)$$

where $L(h,r,t)$ is the query loss and the second term is the path loss. $P(h,t)$ is the set of paths from h to t, and $R(p|h,t)$ is a kind of probability of path p. PTransE uses the similarity between the path p and the relation r to define the path loss. However, $L(p,r)$ is irrelevant with entities in the query, so it is like the weight of p rather than its loss.

3 OUR APPROACH

We first describe our fuzzy inference model. We gradually propose three types of evidence under three assumptions and explain why we finally employ formula operators as evidence. Then we describe the method of estimating evidence by KB embeddings.

3.1 Evidence-Based Inference Model

A typical inference method usually designs a model to combine evidence to infer the conclusion, and they define, collect and filter evidence according to their own models. How to define and acquire evidence is of crucial importance for correct inference. When we infer a query, *Relation(Head Entity, Tail Entity)*, evidence can be all visible information which is related to the query. For the above example in Figure 1, the part of knowledge graph connected to *Cristiano Ronaldo* or *Portugal* is the whole evidence. We propose the first assumption as follows:

Assumption 1. A connected subgraph including both Head Entity and Tail Entity is the whole evidence for predicting any relation between these two entities. If another subgraph is also evidence, there must be missing facts between these two subgraphs.

The assumption can be proved easily, because any irrelevant entity or fact in the KB is useless for predicting the query. When another subgraph affects the inference result, We can always add a link under a relation type to connect entities in these two subgraph. The assumption give the first glance of of the importance of missing facts. Under Assumption 1, an inference model is shown as:

$$f_{r(h,t),s} = W(r(h,t), s) \cdot (1 - D(s)) \tag{5}$$

where $r(h,t)$ is the query, s is the subgraph evidence, and $W(r(h,t), s)$ is the weight of inferring $r(h,t)$ from s. D function represents the distance from finding evidence, and it is defined by the specific method of acquiring evidence. When the method fully finds s, $D(s) = 0$, and when the method never find s, $D(s) = 1$.

Although the connected subgraph contains complete and original information, the whole structure is too sparse and difficult to used by learning models. To solve the problem of sparse feature, We split the evidence graph into independent parts. The most intuitive and convenient substructure is path (where path is a general path which may contain reverse edges), so we make the following assumption.

Assumption 2. All paths connecting Head Entity and Tail Entity are the whole evidence for predicting any relation between these two entities. If an unconnected path is evidence, it always can link Head and Tail Entity by adding facts to it.

This assumption splits the connected subgraph in Assumption 1 into several paths between head entity and tail entity, and these paths are used in the subsequent inference model independently. Similar to Assumption 1, when an unconnected path may have a contribution on inferring the query, we can make it connected by adding links under one specific relation. For example, the unconnected path

Santos $\xrightarrow{Nationality}$ *Portugal* in Figure 1 does not connect to *Cristiano Ronaldo*, and then we can add a link from *Cristiano Ronaldo* to *Santos* under *Son* relation to make it connected. Such situations of missing facts are common in KBs. Under Assumption 2, we also give an inference model, as:

$$f_{r(h,t),P} = \frac{1}{Z} \sum_{p_i \in P} w_i \cdot (1 - D(p_i)) \tag{6}$$

where P is the set of paths connecting h and t, w_i is the weight of p_i, and $Z = \sum_{p_i \in P} w_i$ is the normalizing constant. $D(p_i)$ is the distance from finding p_i, and the range of $D(p_i)$ is also [0,1]. For traditional search-based methods, $D(p_i)$ in Equation (6) has only two values: 1 or 0. When a fact is missing from p_i, $D(p_i)$ is always 1. Therefore, search-based methods cannot handle all paths in Assumption 2.

To hold paths with missing facts, we find that estimating whether some paths exist between entities is easier than searching them exactly. Therefore, we extract relation sequences from paths, and treat them as formula operators to estimate paths between entities. For example, we can obtain the relation sequence *Colleague* $\rightarrow$ *Club* $\rightarrow$ *Location* as an operator from the path *Cristiano Ronaldo* $\xrightarrow{Colleague}$ *Pepe* $\xrightarrow{Club}$ *Real Madrid CF* $\xrightarrow{Location}$ *Spain*, and we can use the formula operator to estimate whether such paths exist between any other two entities. Therefore, we propose the third assumption.

Assumption 3. All types of formula operators existing between Head Entity and Tail Entity are the whole evidence for predicting any relation between these two entities.

This assumption takes a formula operator as a whole by ignoring middle entities and only cares about whether a type of paths occurs. According to Assumption 2, if one middle entity is useful, we can always construct another path which is from Head Entity to Tail Entity and passes the middle entity. There are two advantages of Assumption 3:

1. we can use one distance to estimate the existence of one path type and ignore the number of paths.
2. Paths under the same relation sequence share weights, which reduces the feature sparsity.

We change the inference model in Equation (6) as follows:

$$f_{r(h,t),F_r} = \frac{1}{Z} \sum_{f_{r_i} \in F_r} w_{r_i} \cdot (1 - D(f_{r_i})) \tag{7}$$

where F_r is the set of formula operators for inferring relation r, and $D(f_{r_i})$ is the distance from formula operator f_{r_i} occurring. A formula operator is not a path but a set of paths, and the distance $D(f_{r_i})$ is used to estimate whether there is at least one path under this formula operator.

These three assumptions are layers of the progress. They define three forms of evidence: connected subgraph, path, and formula operator, and these three types of evidence should contain all information required by corresponding inference model. Inference models treat these evidence as features and employ a linear model to merge evidence by attaching weights to them. The simple linear model requires the representation of evidence containing dependency and interactions between elements in KBs, and the evidence representation decides how to acquire evidence and how to define the distance function D.

3.2 Estimate Evidence

We propose our approach under Assumption 3 and represent formula operators as computable real vectors to estimate evidence and quickly calculate distance function $D(f_{r_i})$ in Equation (7). To make formula operators computable, we propose to represent them by KB embeddings.

Embedding means representing each entity in KB as a low-dimension numeric vector, and different dimensions of the vector may implicitly represent different aspects of an entity. Relations in KB usually have relevant representations, such as vectors, matrixes and tensors. Entities interact under a specific relation by performing arithmetical operations between entity embeddings and relation's representation.

To make embedding of a formula operator contain its relations' information and share information with other related formula operators, we exploit the idea of TransE [3], COMB [6] and PTransE [9]. TransE represents relations as translations in the entity vector space R^k, and assumes $E_h + E_r = E_t$ when $r(h,t)$ holds. COMB and PTransE treat a path as a normal relation, and have $E_h + E_p = E_t$ when $p(h,t)$ holds. For a path $h \xrightarrow{r1} x \xrightarrow{r2} t$, we can get $E_h + E_{r_1} = E_x$ and $E_x + E_{r_2} = E_t$. We rewrite the second equation as $E_x = E_t - E_{r_2}$ and use it to eliminate the middle variable x, and then we get $E_h + E_{r_1} + E_{r_2} = E_t$. $E_{r_1} + E_{r_2}$ is naturally used as the representation of the formula operator $\rightarrow r1 \rightarrow r2 \rightarrow$. More generally, we define one formula operator as $E_f = \sum_{r_i \in p} E_{r_i}$.

We also treat a formula operator as a translation. When a formula operator f occurs between h and t, we expect E_h to be close to E_t under E_f. Therefore, we define the $D(f_{r_i})$ in Equation (7) as $D(f_{r_i}) = |E_h + E_{f_{r_i}} - E_t|$, and use it to estimate the probability of existing at least one path under f_{r_i} between h and t. For example, if we want to know whether there is at least one path $\xrightarrow{Father} x \xrightarrow{Nationality}$ between *Cristiano Ronaldo* and *Portugal*, we just calculate distance by $\|E_{C.Ronaldo} + E_{father} + E_{nationality} - E_{Portugal}\|_2^2$ and ignore who *Cristiano Ronaldo*'s father is. When the grounding path *Cristiano Ronaldo* $\xrightarrow{Father}$ *José Dinis Aveiro* $\xrightarrow{Nationality}$ *Portugal* exists in the KB, the distance should be close to 0.

We prove the correctness of the estimating algorithm. When we find a path $p = e1 \xrightarrow{r_1} e2 \xrightarrow{r_2} \ldots \xrightarrow{r_n} e_{n+1}$ by search on the KB graph, if all facts $r_i(e_i, e_{i+1}) \in p$

exist in the KB, we can declare the path p exist. This criterion can be viewed as a necessary and sufficient condition, and we add up each fact's distance to existence as the path distance.

$$\begin{aligned} D(p) &= \sum_{r_i(e_i,e_{i+1})\in p} \|E_{e_i} + E_{r_i} - E_{e_{i+1}}\| \\ &\geq \| \sum_{r_i(e_i,e_{i+1})\in p} E_{e_i} + E_{r_i} - E_{e_{i+1}}\| \\ &= \|E_{e_1} + \sum_{r_i\in f} E_{r_i} - E_{e_{i+1}}\| \\ &= \|E_{e_1} + E_f - E_{e_{i+1}}\| \end{aligned} \tag{8}$$

where $\|\cdot\|$ can be any norm function as a fact distance. According to triangle inequality for norms, the sum of norms is greater than or equal to the norm of the sum, so we can get the second line. The middle entities can be eliminated and there are only relations left as the third line, and the sequence of relations can be viewed as a formula operator. Coincidentally, we have $E_f = \sum_{r_i\in f} E_{r_i}$ and finally get the forth line. The path p's real distance $D(p)$ must be greater than or equal to the distance of the formula operator $D(f) = \|E_{e_1} + E_f - E_{e_{i+1}}\|$. When $D(f) = 0$, we can get all facts $\|E_{e_i} + E_{r_i} - E_{e_{i+1}}\| = 0$, what indicates that the path p exists. Therefore, it is reasonable that we employ $D(f) = \|E_{e_1} + E_p - E_{e_{i+1}}\|$ to estimate whether there is at least one path under formula operator f.

Dataset	Relation	Entity	Train	Valid	Test
WN18	18	40 943	141 442	5 000	5 000
FB15K	1 345	14 951	483 142	50 000	59 071

Table 1. Statistics of WN18 and FB15K

Before estimating formula operators, this approach needs firstly mining a set of formula operators for each relation type. We employ the Depth-First-Search (DFS) algorithm to traverse the KB graph several times to count frequent relation sequence, and limit the maximum length of paths. Then we simply rank the formula operators by occurrence number, and choose top-K as the set of formula operators. How to mine formula operators is not a focus of this paper, so we avoid missing useful formula operators just by choosing a large K.

3.3 Objective Formalization

We rewrite Equation (7) as the score function for a triplet $r(h,t)$, as:

$$f_{r(h,t),F^r} = 1 - \frac{1}{Z} \sum_{f_i^r\in F_r} w_i^r \cdot \|E_h + E_{f_i^r} - E_t\|_2^2 \tag{9}$$

	Dataset	WN18				FB15K			
	Metric	Mean Rank		Hits@10(%)		Mean Rank		Hits@10(%)	
		Raw	Filt	Raw	Filt	Raw	Filt	Raw	Filt
2.a	INS*(MLN)	329	319	55.5	66.33	242	226	49.2	60.3
2.b	RESCAL	1 180	1 163	37.2	52.8	828	683	28.4	44.1
	TransE	263	251	75.4	89.2	243	125	34.9	47.1
	TransH	401	388	73.0	82.3	212	87	45.7	64.4
	TransR	238	225	79.8	92.0	**198**	77	48.2	**68.7**
2.c	COMB	504	491	78.1	90.7	212	92	39.9	52.6
2.d	FIEE	**133**	**127**	**93.5**	**96.9**	221	**76**	**49.8**	68.6

Table 2. Knowledge base completion results

where w_i^r measures the correlation between formula operator f_i^r and the query relation r, and $w_i^r > 0$ represents positive correlation to predicting the query and $w_i^r < 0$ represents negative correlation. Especially, to take advantage of the implicit relationship between entities as TransE does, we treat r itself as a special formula operator. In Equation (9), only w is parameter and needs to be learnt by training model, while we use embedding E pre-trained by TransE and never change them. We employ a margin-based rank loss to training model, as:

$$\mathcal{L} = \sum_{f_s \in KB} \sum_{f_s' \in KB'} [\gamma + \sum_{p_i^r \in P_r} w_i^r \cdot (D_{f_s'}(p_i^r) - D_{f_s}(p_i^r))]_+ \tag{10}$$

where $D_{f_s}(p_i^r)$ and $D_{f_s'}(p_i^r)$ represent path distances of true and false queries, respectively.

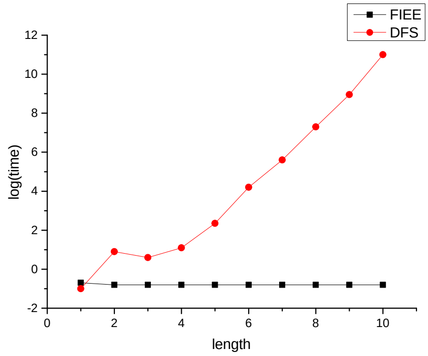
4 EXPERIMENTS AND ANALYSIS

We have compared our approach with several state-of-the-art methods on KB completion (KBC) task. We predict the missing h or t for a fact $r(h,t)$ in the test set. The detail evaluation method is to replace t in $r(h,t)$ by all entities in the KB to form left testing set Q_{left}, and methods need to rank the right answer at the top of the list. Similarly, we produce the right testing set Q_{right}. We perform the experiments on both WN18 and FB15K datasets which were subsets sampled from WordNet[11] and Freebase [1], respectively, and Table 1 shows statistics of them. We report the mean of those true answer ranks and the Hits@10[2] under both 'raw' and 'filter' as TransE does.

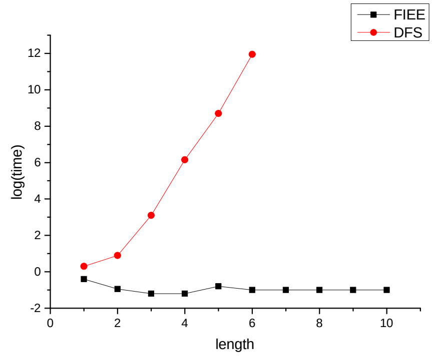
For comparison, we employ 3 types of methods as baselines:

1. search-based method: MLN [14];
2. embedding-based methods: RESCAL [13], TransE [3], TransH [15], TransR [10];
3. embedding and path method: COMB [6].

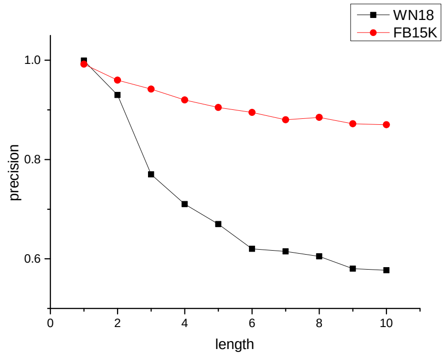
[2] The proportion of correct entities ranked in the top 10.



a) Runtime on WN18



b) Runtime on FB15K



c) Precision

Figure 2. The running time and performance of estimating evidence

We employ INS* [16] as the implement of MLN model. For embedding methods, we use reported results directly since the evaluation datasets are identical. We also implement COMB algorithm, and generate its path query dataset randomly.

We call our approach as FIEE (Fuzzy Inference by Estimate Evidence). For our approach, we select different parameters for two datasets. We use stochastic gradient descent (SGD) to learn embeddings and weights, and we employ validate set to select parameters. We select the learning rate λ among {0.001, 1e-4, 1e-5, 1e-6}, the margin γ among {0.125, 0.25, 0.5, 1}, the dimensionality of entity and relation k among {50,100}, the L_2 regularization coefficient among {0.1, 1, 10} and the maximum length of paths l_{max} among {2,3,4,5,6,7} for WN18 and {2,3,4} for FB15K. Optimal configurations are: $\lambda = 0.001$, $\gamma = 0.5$, $k = 50$, $L_2 = 10$ and $l_{max} = 4$ for WN18; $\lambda = 0.001$, $\gamma = 1.0$, $k = 100$, $L_2 = 0.1$ and $l_{max} = 3$ for FB15K.

4.1 Our Method vs. State-of-the-Art Methods

Table 2 shows the KBC results on both WN18 and FB15K, and we can obtain the following observations.

1. FIEE achieves good performances on both WN18 and FB15K. For WN18, our approach outperforms all state-of-the-art methods on all metrics. It indicates that our approach is effective for knowledge inference.
2. Comparing FIEE with INS in Table 2 a), FIEE outperforms INS on all metrics. It indicates our fuzzy inference is more robust to missing facts than search-based inference. Noise paths can weaken INS seriously and its performance gets worse as the number of candidates grows, which is described in [16]. On the other hand, FIEE treats all entities in the KB as candidates, which implies fuzzy inference can weaken the negative effect of noise paths.
3. Comparing FIEE with embedding-based methods in Table 2 b), FIEE outperforms them on metrics except Mean Rank(raw) and Hits@10(filt) on FB15K dataset. It indicates that introducing explicit logic constrains into the embedding model can improve the KBC performance, and it shows that knowledge graph embedding models have limited inference ability.
4. Comparing FIEE with COMB in Table 2 c), FIEE outperforms COMB. It indicates that paths used as evidence have more effect than used as additional training data, though both of them have the same way of representing paths. Comparing TransE, COMB has no obvious advantage, which also indicates that COMB cannot utilize structure information completely.
5. The best performance of our approach occurs when the maximum path length is 4 for WN18 and 3 for FB15K. This implies there would be more noise when the path length increases in both estimating evidence and the KBC task.
6. Our approach's performance on FB15k is not as good as on WN18. We think the reason may be that FB15K has much more relation types than WN18, which means there would be huge size of formula operators in FB15K. Therefore, it is hard to cover all correlative and meaningful formula operators, and some of ones we selected may be noise.

4.2 Comparison on Running Time

To prove our approach's absolute predominance on running time, we design an experiment to compare the running time of our estimating algorithm and Depth-First-Search algorithm.

We construct a dataset which contains 10 000 various true or false (1:1) path instances with length from 1 to 10, and run FIEE 100 times on it to estimate paths. As comparison, DFS algorithm searches same path instances 100 times, and we compare their running times.

Figures 2 a) and 2 b) show methods' running times on WN18 and FB15K, respectively. We show logarithms of running time and find the running time of DFS increasing exponentially with the growth of path length. Especially on FB15K, DFS cannot finish in 72 hours when path length reaches 6. However, our approach's running time almost remains unchanged with path length increasing and always is under 1 second. It shows our approach's victory by great superiority on running time and indicates that our approach is high-efficient for estimating evidence no matter how long it is.

4.3 Estimate Evidence

The performance of estimating formula operators has an important effect on the KBC result, so we take an experiment to evaluate our approach's performance on it. We still employ the dataset in running time experiment and use estimating path existence to approximate formula operators. Figure 2 c) shows the precision on both WN18 and FB15k, and we can obtain the following observations:

1. Our approach achieves a good performance on estimating paths, which proves our approach can replace searching paths on the large-scale graph.
2. The precision is reducing with the growth of path length both in WN18 and FB15k. For WN18, when the length reached 5, the precision was less than 70 %. There are two reasons of this phenomenon:

   (a) there are cascading errors which are mentioned in [6], and the longer the path is, the larger the cascading error is;
   (b) there may be missing facts in paths, and the longer the path is, the more facts are missing.

   The interesting thing is that the first reason has adverse effect on KBC task, while the second reason is reverse. There is an optimum maximum length of paths for each specific KB.
3. The precision of estimating path existence on FB15K is higher than the precision on WN18, but the KBC performance on WN18 is better. It further proved that our approach can handle paths with missing facts, and these paths can improve KBC performance.

5 CONCLUSION AND FUTURE WORK

This paper presents a novel method, called FIEE, to perform fuzzy inference by estimating evidence. FIEE treats relation sequences as computable formula operators to estimate path existence between entities. FIEE never searches on KBs, so its running time is short and it can be applied to large-scale KBs. Estimating evidence can also be viewed as fuzzy matching, so this method can handle the situation where facts are missing. We evaluate our approach on the KBC task, and it achieves good performances on both WN18 and FB15K datasets.

In future, we will explore the following research directions:

1. The precision of estimation is not very high, and we think the reason is that the method of representing formula operator is not good enough, so we need explore better representation of formula operators.
2. In this paper, we still need to get formula types before learning the inference model, and then we want to deal with them simultaneously.

REFERENCES

[1] Bollacker, K.—Evans, C.—Paritosh, P.—Sturge, T.—Taylor, J.: Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data (SIGMOD '08), 2008, pp. 1247–1250, doi: 10.1145/1376616.1376746.

[2] Bordes, A.—Glorot, X.—Weston, J.—Bengio, Y.: A Semantic Matching Energy Function for Learning with Multi-Relational Data. Machine Learning, Vol. 94, 2014, No. 2, pp. 233–259, doi: 10.1007/s10994-013-5363-6.

[3] Bordes, A.—Usunier, N.—Garcia-Durán, A.—Weston, J.—Yakhnenko, O.: Translating Embeddings for Modeling Multi-Relational Data. In: Burges, C. J. C. et al. (Eds.): Advances in Neural Information Processing Systems 26 (NIPS 2013), Vol. 2, 2013, pp. 2787–2795.

[4] Bordes, A.—Weston, J.—Collobert, R.—Bengio, Y.: Learning Structured Embeddings of Knowledge Bases. Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence (AAAI '11), 2011, pp. 301–306.

[5] Brocheler, M.—Mihalkova, L.—Getoor, L.: Probabilistic Similarity Logic. Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence (UAI 2010), 2010, pp. 73–82.

[6] Guu, K.—Miller, J.—Liang, P.: Traversing Knowledge Graphs in Vector Space. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2015, pp. 318–327, doi: 10.18653/v1/D15-1038.

[7] Jenatton, R.—Le Roux, N.—Bordes, A.—Obozinski, G. R.: A Latent Factor Model for Highly Multi-Relational Data. In: Pereira, F. et al. (Eds.): Advances in Neural Information Processing Systems 25 (NIPS 2012), 2012, pp. 3167–3175.

[8] Lao, N.—Mitchell, T.—Cohen, W. W.: Random Walk Inference and Learning in a Large Scale Knowledge Base. Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP), ACL, 2011, pp. 529–539.

[9] Lin, Y.—Liu, Z.—Luan, H.—Sun, M.—Rao, S.—Liu, S.: Modeling Relation Paths for Representation Learning of Knowledge Bases. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2015, pp. 705–714.

[10] Lin, Y.—Liu, Z.—Sun, M.—Liu, Y.—Zhu, X.: Learning Entity and Relation Embeddings for Knowledge Graph Completion. Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI '15), 2015, pp. 2181–2187.

[11] Miller, G. A.: Wordnet: A Lexical Database for English. Communications of the ACM, Vol. 38, 1995, No. 11, pp. 39–41, doi: 10.1145/219717.219748.

[12] Neelakantan, A.—Roth, B.—McCallum, A.: Compositional Vector Space Models for Knowledge Base Completion. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2015, pp. 156–166, doi: 10.3115/v1/P15-1016.

[13] Nickel, M.—Tresp, V.—Kriegel, H.-P.: A Three-Way Model for Collective Learning on Multi-Relational Data. Proceedings of the 28th International Conference on Machine Learning (ICML '11), 2011, pp. 809–816.

[14] Richardson, M.—Domingos, P.: Markov Logic Networks. Machine Learning, Vol. 62, 2006, No. 1-2, pp. 107–136, doi: 10.1007/s10994-006-5833-1.

[15] Wang, Z.—Zhang, J.—Feng, J.—Chen, Z.: Knowledge Graph Embedding by Translating on Hyperplanes. Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI '14), 2014, pp. 1112–1119.

[16] Wei, Z.—Zhao, J.—Liu, K.—Qi, Z.—Sun, Z.—Tian, G.: Large-Scale Knowledge Base Completion: Inferring via Grounding Network Sampling over Selected Instances. Proceedings of the 24th ACM International Conference on Information and Knowledge Management (CIKM '15), 2015, pp. 1331–1340, doi: 10.1145/2806416.2806513.

[17] Wu, F.—Song, J.—Yang, Y.—Li, X.—Zhang, Z.—Zhuang, Y.: Structured Embedding via Pairwise Relations and Long-Range Interactions in Knowledge Base. Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI '15), 2015, pp. 1663–1670.

Xiao Zhang is a teacher in the China University of Mining and Technology, Beijing. She received her Ph.D. degree in computer software and theory from the Institute of Software, Chinese Academy of Sciences, Beijing in 2018. She received her B.Sc. degree in computer science and technology from the Ji Lin University in 2012. Her research interests include operating systems and distributed systems.