INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 13(1), 71-82, February 2018.

A Knowledge Base Completion Model Based on Path Feature Learning

X. Lin, Y. Liang, L. Wang, X. Wang, M. Yang, R. Guan

Xixun Lin, Xu Wang
Key Laboratory for Symbol Computation and
Knowledge Engineering of National Education Ministry,
College of Computer Science and Technology,
Jilin University, Changchun 130012, China

Limin Wang
School of Management Science and Information Engineering,
Jilin Province Key Laboratory of Internet Finance,
Jilin University of Finance and Economics, Changchun 130117, China

Mary Qu Yang
MidSouth Bioinformatics Center and Joint Bioinformatics Ph.D. Program,
University of Arkansas at Little Rock and
University of Arkansas for Medical Sciences, 2801 S.
University Avenue, Little Rock, Arkansas 72204, USA

Yanchun Liang, Renchu Guan*
Key Laboratory for Symbol Computation and
Knowledge Engineering of National Education Ministry,
College of Computer Science and Technology,
Jilin University, Changchun 130012, China
Zhuhai Laboratory of Key Laboratory of Symbolic Computation and
Knowledge Engineering of Ministry of Education,
Zhuhai College of Jilin University, Zhuhai 519041, China
*Corresponding author: guanrenchu@jlu.edu.cn

Abstract: Large-scale knowledge bases, as the foundations for promoting the development of artificial intelligence, have attracted increasing attention in recent years. These knowledge bases contain billions of facts in triple format; yet, they suffer from sparse relations between entities. Researchers proposed the path ranking algorithm (PRA) to solve this fatal problem. To improve the scalability of knowledge inference, PRA exploits random walks to find Horn clauses with chain structures to predict new relations given existing facts. This method can be regarded as a statistical classification issue for statistical relational learning (SRL). However, large-scale knowledge base completion demands superior accuracy and scalability. In this paper, we propose the path feature learning model (PFLM) to achieve this urgent task. More precisely, we define a two-stage model: the first stage aims to learn path features from the existing knowledge base and extra parsed corpus; the second stage uses these path features to predict new relations. The experimental results demonstrate that the PFLM can learn meaningful features and can achieve significant and consistent improvements compared with previous work.

Keywords: knowledge base completion, random walks, path features, extreme learning machine.

1 Introduction

Large-scale knowledge bases (KBs), such as Never-Ending Language Learning (NELL) [6], Freebase [3], Yago [35], and DBpedia [11], construct their own ontologies derived from facts that are manually or automatically extracted from databases, such as Wikipedia or other unannotated web pages. These KBs usually contain billions of facts, and each fact can be organized as a triple $R_k(a_i, b_j)$, such as AthletePlaysForTeam (Messi, Barcelona) and Professions (van Gogh, Artist). The variables a and b represent entities or attributes in the real world, and R represents the binary relationship between them. Billions of these facts constitute a large complicated knowledge network. The construction of such large-scale KBs is significant for many types of natural language processing research, e.g., question answering [2], semantic analysis [1], and information retrieval [13].

However, existing facts stored in KBs are not comprehensive compared with real-world knowledge. Many important entity-relationships are missing due to improper manual operations or the drawbacks of fact-extraction models [38]. Fortunately, most of the information can be inferred from existing facts in KBs. Thus, the task of knowledge base completion (KBC) has aroused intense attention in both academic and industry research [27, 36].

Graph-based approaches for KBC is an important subfield of statistical relational learning (SRL) [10]. A classic learning method in SRL is Markov logic network (MLN) [32], which combines the probabilistic graphical model with first-order logic for knowledge inference. Although MLN is very powerful, it needs to explore all derivation trees and combine them to calculate even a single ground fact; therefore, MLN does not perform well when the amount of data is large. Inductive logic programming (ILP), such as the first-order inductive learner (FOIL) and its variants, is another type of SRL [18, 23, 31]. FOIL deploys the separate-and-conquer strategy to learn the first-order logic rule set; however, similar to MLN, FOIL cannot handle large-scale KBs. ProPPR [37] is a new direction that attempts to improve the scalability of FOIL.

The path ranking algorithm (PRA) [19] aims to reason new facts directly from facts observed in the KBs and achieves state-of-the-art performance compared with previous work. PRA attempts to encode the entire KB as a large edge-labeled directed graph and exploits random walks to extract informative path features. Each path feature can be viewed as the most frequent sequence of relations in the graph. However, PRA is deficient in terms of addressing long-tail data distributions because some rare entities lack common path features. Another direction emphasizes the extraction of relations from a large text corpus and combination with current KBs to enhance the performance of PRA [7–9, 20, 21]. The extracted relations are added to the original directed graph as new edges to compensate for the lack of sufficient facts. However, directly adding relations with similar semantics to the KB results in severe path explosion and feature sparsity.

In addition to the aforementioned models, many approaches focus on latent feature models [27], which aim to learn latent representations of entities and relations by minimizing a reconstruction loss or a margin-based ranking loss [4, 12, 25, 28, 29]. Latent feature models are effective to encode knowledge representations, but when the KB tensor constructed from data has a higher rank, it is more difficult to obtain meaningful embeddings.

In this paper, motivated by PRA, we propose a more general framework called the path feature learning model (PFLM). The PFLM is a two-stage model: In the first stage, two types of path features (directed relation paths (DRPs) and supplement relation paths (SRPs)) are generated from different target entity pairs by random walks; in the second stage, we incorporate the learned features into the kernel extreme learning machine (KELM) [14, 15] for KBC. In addition to the advantages of good generalization and fast learning speed [16], KELM is robust in terms of triple classification, which is illustrated in our experiments. The main highlights of this paper are summarized as follows:

1. With the stage of path feature learning, plenty of classification algorithms (in our case, we use ELM and KELM) can be easily incorporated into our framework. Using the new path

features and the single-hidden layer feedforward neural network, the PFLM can perform KBC effectively and efficiently.

2. The PFLM is an extensible scheme. In addition to the DRPs introduced by PRA, the proposed SRPs and other expected path information (e.g., immediate nodes) can be collected in the path feature learning stage.

3. The results of experiments show that our model achieves significant and consistent improvements compared with baseline models, such as PRA [19] and CPRA [7].

The rest of our paper is organized as follows: Section 2 introduces the background of basic model setting. In Section 3, we introduce the detailed implementation strategies for our model. Experimental details and discussions are provided in Section 4. The last section draws our conclusions and identifies future work.

2 Background

In this section, we first give a brief overview of PRA and KELM. PRA is a classic algorithm of KBC. KELM is an efficient neural network architecture that we employ to implement triple classification. Triple classification [33] is a standard way to evaluate KBC.

2.1 Path ranking algorithm (PRA)

PRA leverages multiple random walks to reach tail entity b from head entity a for each entity pair (a,b). The filtered paths that connect entity pair (a,b) serve as different path features. Then, the random probability of head entity a reaching tail entity b through path-constrained random walks is calculated as the feature value. Finally, PRA adopts logistic regression based on the limited-memory Broyden-Fletcher-Goldfarb-Shanno(L-BFGS) to perform triple classification. The effectiveness of PRA depends on the power of the Horn clause rules to obtain the entity constraints. A relation path is generated by the conjunction of a sequence of triples, for example, given WriterCreatedRole (Hemingway, Santiago) → RoleDescribledInBook (Santiago, The Old Man and the Sea), the predicative information WriterWroteBook (Hemingway, The Old Man and the Sea) can be inferred. By exploiting the implicit relation paths, novel facts that do not originally exist in the KBs are produced.

The more detailed procedures are as follows: PRA encodes the whole KB as a directed edge-labeled graph $G(N, T, E)$. N is the set of entities in KB, E is the set of edges connecting entity pairs, and T is the collection of edge types representing first-order logic rules. A path P is the ordered sequence of edges P=$\{R_1$,$\ldots$,$R_n\}$. PRA applies multiple random walks, starting from the head entity a, to obtain the common path set $S(P) = \{P_1, \ldots, P_k\}$ for the common tail entity b. $S(P)$ is relevant to a specific relation R. The most frequent paths from $S(P)$ are selected as path features. Each path feature value $V_{a,P}(b)$ is calculated using the recursive formulas defined as follows.

If P is an empty path:

$$V_{a,P}(b) = 1 \quad \text{if} \quad a = b \tag{1}$$

$$V_{a,P}(b) = 0 \quad \text{otherwise.} \tag{2}$$

If P is not an empty path, $P' = R_1, \ldots, R_{n-1}$:

$$V_{a,P}(b) = \sum_{b' \in range(P')} V_{a,P'}(b') \cdot P(b|b'; R_n), \tag{3}$$

where $P(b|b'; R_n)$ is the probability of reaching target node b from b' with one edge labeled R_n, and $range(P')$ is the set of target nodes where the path P' ends.

2.2 Kernel extreme learning machine (KELM)

Extreme learning machine is a learning algorithm aims to train single-hidden layer feedforward neural networks. Huang provided strict theoretical proof that the standard single-hidden layer feedforward neural networks training process can be considered as finding a least-squares solution β' for the linear system $H\beta = T$ that allows the hidden node parameters w_i and b_i to be randomly generated. H is the hidden layer's output matrix, β is the output weights and T represents the training sample outputs. In most cases, when the number of hidden nodes is much less than the number of distinct training samples, $\beta' = H^{\dagger}T$ can be calculated with zero error to approximate these training examples, where $H^{\dagger}$ is the Moore-Penrose generalized inverse of H. ELM shows good generalizability and learning speed; it performs better than the conventional learning algorithm in many areas, such as face recognition [40], text classification [39], image classification [5], and medical diagnosis [24].

Kernel extreme learning machine (KELM), in contrast to traditional ELM, conducts classification and regression with kernel functions and kernel parameters (C, γ) instead of the number of hidden-layer nodes and path feature mappings $h(x)$. The reason we choose KELM as our classifier is its robustness, which will be discussed in our experiments. The kernel matrix is defined as

$$\Omega_{ELM} = HH^T \tag{4}$$

$$\Omega_{ELM}(i, j) = h(x_i) \cdot h(x_j) = K(x_i, x_j), \tag{5}$$

where $K(x_i, x_j)$ is the kernel function.

The output weights and output function are expressed as follows:

$$\beta = H^T(\frac{I}{\lambda} + HH^T)^{-1}T \tag{6}$$

$$f(x) = h(x)\beta = \begin{bmatrix} K(x, x_1) \\ \cdot \\ \cdot \\ \cdot \\ K(x, x_N) \end{bmatrix}^T (\frac{I}{\lambda} + \Omega_{ELM})^{-1}T. \tag{7}$$

3 Proposed method

Motivated by PRA, we propose a novel model called the path feature learning model (PFLM), which combines path feature learning with KELM to achieve triple classification. Fig. 1 illustrates the framework for our model. The inputs of our model are the knowledge base and the large parsed corpus. The path feature learning aims at providing a large real-valued matrix for KELM. The final output of PFLM is predicting whether the entity pair contains the specific relation, which is a binary classification problem. The main reason for combining these operations is that the first stage of our model assumes that all background knowledge and samples are ground facts, which makes it more suitable for machine learning models. On the premise of representing the KB as a large directed heterogeneous graph with encodable Horn clause rules, random walk inference outperforms traditional searching methods, such as the breadth-first search algorithm, in the process of the feature searching. The retrieved first-order Horn clause rule sets, which are represented as paths, can be recognized as candidate path features. We first introduce the

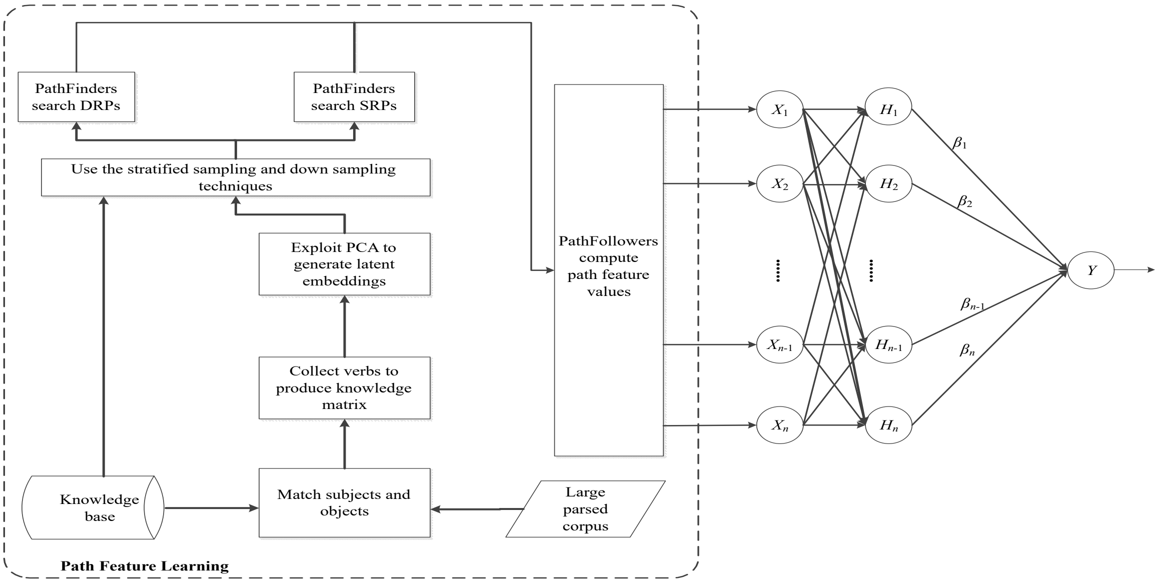
PathFinders search DRPs
PathFinders search SRPs
Use the stratified sampling and down sampling techniques
Exploit PCA to generate latent embeddings
Collect verbs to produce knowledge matrix
Match subjects and objects
Knowledge base
Large parsed corpus
PathFollowers compute path feature values
Path Feature Learning
X_1
X_2
X_{n-1}
X_n
H_1
H_2
H_{n-1}
H_n
β_1
β_2
β_{n-1}
β_n
Y

Figure 1: Flowchart of the path feature learning model.

concept of path feature learning, which includes DRPs and SRPs. Then, we introduce two practical techniques in subsection 3.2 for efficient implementation.

3.1 Path feature learning

In the path feature learning stage, we define two path types according to their components. DRPs are the existing chains of the Horn clause rules that are directly searched by PRA. SRPs are the sequences of relations with some latent embeddings that are learned from factorization of the KB matrix M. Specifically, each row of the matrix is a tuple $t = (subject, object)$, while *subject* and *object* are entities that exist in the KB and also occur in 500 million dependency-parsed Web documents [7, 21]. Each column of the matrix is the verb in t. The elements of the matrix are the frequencies of (t, *verb*) in the extra corpus. After row normalization and centering, we apply PCA to M to obtain the latent embeddings of the *verb*s. As shown in the Fig. 1, both DRPs and SRPs can be searched and computed by PRA according to the equations 1, 2 and 3.

We consider a specific example to explicitly describe the two path types. An important reasoning path for the triple AthletePlaysInLeague (LeBron James, NBA) can be expressed by DRP as AthletePlaysForTeam (LeBron James, Cleveland Cavaliers) → TeamBelongedToLeague (Cleveland Cavaliers, NBA). Unfortunately, the knowledge repository may miss the valuable relational edge AthletePlaysForTeam (LeBron James, Cleveland Cavaliers), so it seems impossible to infer the fact AthletePlaysInLeague (LeBron James, NBA), which is known to us but disappears in the KB. This feature sparsity problem leads to severe over-fitting for many isolated entity pairs, which limits the performance of PRA.

In our method, the proposed SRPs combine the extracted subject-verb-object (SVO) information that expresses similar semantics of AthletePlaysForTeam from the extra text corpus. The method exploits verb clustering techniques to map the lexicalized edge labels (e.g., 'works for' , 'plays for', 'leads to') to some latent embeddings, forming a new edge type: LatentEM1 (LeBorn James, Cleveland Cavaliers). The probability of extracting important path features is increased by using both DRPs and SRPs compared with that of PRA, which only considers DRPs in

KBs. The path features are quite logical for our understanding and effective for the large feature space. After the entity pair path information has been taken into account for all datasets, the most frequent m candidate path features are selected as the final path features. Each path feature is composed of relations or latent embeddings or mixtures of both. It is reasonable to regard the random probability values as path feature values, even if the computation requires enormous running time. The larger the probability is, the more likely a specific path feature will be selected for our target relation. This attribute cannot be reflected by binary features. Finally, the abstract graph information is mapped to relevant feature vectors, and the KELM completes the task of triple classification.

It is noted that the PFLM is a general scheme. We chose KELM as our second stage classification model, in contrast to the logistic regression model adopted by PRA. Different kernels and parameters can be selected for different relation decisions. In general, in the stage of path feature learning, in addition to DRPs, we propose SRPs to explictly identify path features from the KBs and extra corpus; in the second stage, the feature matrices are transferred to the kernel extreme learning machine classifier to complete the task of triple classification. Furthermore, each feature matrix is computed by random walks following the concrete relations indicated by DRPs or SRPs. Missing relations can be added to the KBs based on accurate classification results.

3.2 Two practical sampling techniques

Due to the knowledge bias in KBs, sampling techniques must be employed to balance the datasets. When adopting random walks to obtain positive and negative samples, two points must be considered. First, the relation types contained in KBs are probably uneven, and this phenomenon may affect the overall model's feature distribution. Therefore, we consider [21], which employs stratified sampling [32] to obtain identical sample numbers for different types of relations when possible. Secondly, PRA holds the closed-world assumption. In addition to the small portion of positive samples already existing in knowledge graph G, most of the samples are negative ones produced by random walks, which may lead to a serious distribution imbalance. We use a downward sampling technique to solve this problem. To control the positive and negative samples within a reasonable proportion (in our case, the ratio is 1:10), the PFLM selects the common relation paths in accordance with the sampling numbers. For example, in a large-scale KB, we present thousands of entity pairs for AthletePlaysInLeague. In addition to the path AthletePlaysForTeam → TeamBelongedToLeague, many other related paths, such as AthletePlaysSport → AthletePlaysSport^{-1} → LeaguePlayers^{-1}, AthletePlaysSport→StadiumHomeToSport^{-1}→ LeagueStaduim^{-1}, and AthletePlaysForTeam → AthletePlaysForTeam^{-1} →LeaguePlayers^{-1} are contained in the set. We denote R^{-1} as the inverse of relation R (i.e., WriterWroteBook^{-1} is equivalent to BookWritenByWriter). After sampling and feature computing, the PFLM provides the final feature matrices composed of multiple entity pairs with their feature vectors for the target relation AthletePlaysInLeague to the KELM classifier.

4 Experiments and discussion

We use the Never-Ending Language Learning (NELL) dataset to evaluate our model. NELL is a large-scale KB whose contents are learned by reading from the web over time. The dataset we employ is a benchmark and can be downloaded from `http://rtw.ml.cmu.edu/emnlp2013_pra/`. This dataset contains 15 relations, and each relation is split into two parts: 10% test data and 90% training data. To rigorously compare and decrease the extensive feature computing, we follow the same rules as CPRA [7] and set the number of each sample's path features as 750.

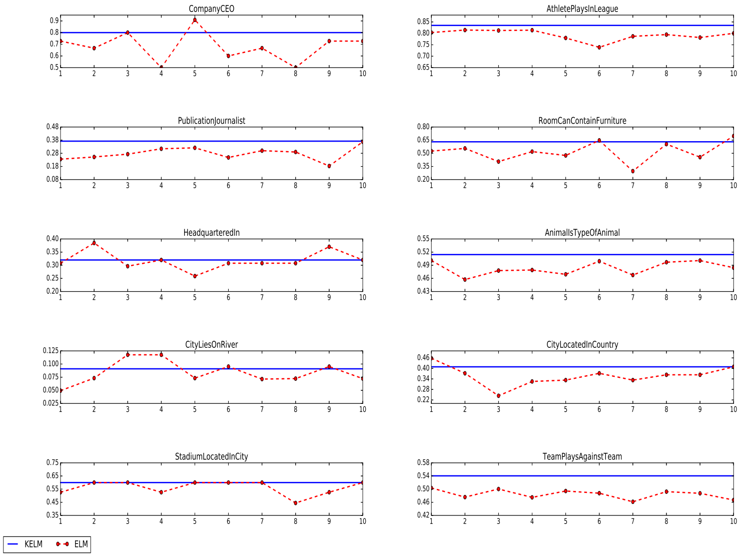

Figure 2: Comparison of 10 relations' data with ELM and KELM based on RBF kernel function. The horizontal axis represents the experiment times, and the vertical axis represents the F1-measure.

Table 1: The comparisons of four different kernels for KELM

	Macro-precision	**Macro-recall**	**Macro-accuracy**	**Macro-F1**
Linear kernel	0.5249	0.3798	0.8939	0.4130
Wavelet kernel	0.5974	0.3862	0.8960	0.4332
RBF kernel	**0.8825**	**0.4172**	**0.9488**	**0.5279**
Polynomial kernel	0.8021	0.4034	0.9446	0.4986

We first compare the robustness of ELM and KELM for triple classification on ten diverse relation datasets; the corresponding path features are the same and are both provided by path feature learning. The results are shown in Fig. 2. From Fig. 2, it can be seen that when the number of samples is small, the ELM's F-measure oscillates between several fixed values, and the vibration is drastic in some relation datasets, such as StadiumLocatedInCity. When the number of samples is larger, we intend to implement more hidden layer nodes to improve the network generation, and ELM fluctuates more severely, thus causing a decline in accuracy and stability, such as the relation RoomCanContainFurniture. By contrast, when we choose KELM and fix the corresponding kernel parameters, KELM is more robust and performs better than ELM. Therefore, we chose KELM as our classifier instead of ELM based on the experimental results.

Moreover, we compare the performance of different kernel functions on the same task. Table 1 presents the experimental results. The best performances are indicated in bold for all tables in this section. We employ the grid-search strategy to select the best parameters C and γ for different kernel functions. The parameters are both tuned in $\{2^{-5}, 2^{-3}, \ldots, 2^{15}\}$. We evaluate the linear, wavelet, RBF and polynomial kernels on four measurements among 15 relation datasets. From the results, we can observe that the effects of different kernels vary considerably, and the RBF kernel achieves the best performance. The most notable result is that the RBF kernel is 35.8%, 3.8%, 5.5%, and 11.5% higher than the Linear kernel, which indicates that the dot

Table 2: Running time comparison for PRA, CPRA and PFLM

	PRA	**CPRA**	**PFLM**
Time(min)	1313	1160	1191

Table 3: Detailed comparison of the F1-measure for PRA, CPRA, and PFLM

	PRA	**CPRA**	**PFLM**
AnimalIsTypeOfAnimal	0.5214	0.5270	**0.6069**
AthletePlaysForTeam	0.2156	0.6387	**0.6667**
AthletePlaysInLeague	0.8099	0.7402	**0.8370**
CityLiesOnRiver	0.0493	0.3076	**0.5484**
CityLocatedInCountry	0.1538	0.5454	**0.5778**
CompanyCEO	0.2857	0.3529	**0.7273**
CountryHasCompanyOffice	0.0000	0.0000	**0.1481**
DrugHasSideEffect	**0.9629**	0.9427	0.9474
HeadquarteredIn	0.3076	**0.6382**	0.6047
locationLocatedwithinLocation	0.3950	0.4147	**0.4286**
PublicationJournalist	0.0967	0.1594	**0.5444**
RoomCanContainFurniture	0.7206	0.7320	**0.7985**
StadiumLocatedInCity	0.5263	0.6666	**0.7143**
TeamPlaysAgainstTeam	0.4736	0.2086	**0.5800**
WriterWroteBook	0.5911	0.8000	**0.8218**

product in an infinite-dimensional space is more suitable for our problem. Therefore, we choose the RBF kernel as our kernel function in the following experiments. In Table 2 we compare the running times of PRA, CPRA, and PFLM. From Table 2 we can conclude the following. 1) PRA adopts logistic regression with L2 regularization optimized by L-BFGS to complete the triple classification. It calculates the path feature values with dozens of iterations, which means that its convergence is slow. 2) CPRA reduces the number of relation paths that random walkers need to search, therefore it is the fastest method. 3) Although the computation and selection for the path feature learning of the DRPs and SRPs consume substantial running time, the PFLM is not as time-consuming as PRA.

In Table 4 we compare PRA, CPRA and PFLM on 15 relations. The experimental results show that the PFLM achieves significant and consistent improvement. For example, the PFLM is 12% and 23% higher than CPRA and PRA on the macro F1 criterion, respectively. We believe the PFLM outperforms CPRA because the PFLM can better incorporate expressive path information to address the problem of feature sparsity, which limits the performance of CPRA and PRA. We reproduce the experiments in [30], and Table 3 shows a more detailed comparison of the F1 measurement for the three models on 15 NELL relations. From Table 3, we can conclude that the PFLM shows significant improvement in the triple classification of 13

Table 4: Macro measurement comparison for PRA, CPRA and PFLM (%)

	Macro-precision	**Macro-recall**	**Macro-F1**	**Macro-accuracy**
PRA	0.7458	0.3443	0.4073	0.9373
CPRA	0.8094	0.4241	0.5116	0.8708
PFLM	**0.9009**	**0.5152**	**0.6367**	**0.9439**

Table 5: Most impressive path features in the three relations

	Path Type	Element of path
AtheletePlaysForTeam	DRPs	-athleteledsportsteam-
		-atheltéledssportsteam-teamhomestadium-teamhome stadium^{-1}-
	SRPs	-LE1^{-1}-LE2-
		-LE1^{-1}-LE2-teamhomestadium-teamhomestadium^{-1}-
CityLiesOnRiver	DRPs	-proximityfor-subpartof^{-1}-riverflowsthroughcity^{-1}-
		-statecontainscity^{-1}-atlocation-riverflowsthrough city^{-1}-
	SRPs	-LE1LE5^{-1}-
		-LE1LE5^{-1}-riverflowsthroughcity-riverflowsthrough city^{-1}-
CompanyCEO	DRPs	-organizationleadbyperson-
		-worksfor-
	SRPs	-LE1LE5- LE1LE5^{-1}-agentcollaborateswithagent-
		-LE1^{-1}-LE2-LE1LE2-agentcollaboratesswithagent-

relations, with the largest increases of 463.0 % and 241.5 % compared with PRA and CPRA on the relation PublicationJournalist. Table 5 displays the impressive path features (DRPs and SRPs) that are searched by path feature learning on three different relations. These paths are common to all entity pairs, and their importance in triple classification are reflected by their own path feature values. For each path type, we present two examples, and we can observe that the highest-weighted DRPs and SRPs have similar semantics with the target relations.

5 Conclusions and future work

In this paper, we propose the PFLM to solve the problem of large-scale KBC. The PFLM extracts DRPs and SRPs during the path feature learning stage and sends these features to the kernel extreme learning machine. The PFLM shows superior classification ability compared with the original algorithm and its variants on the benchmark datasets. For our future work, we plan to 1) incorporate more path information into the path feature learning schemes, e.g., the path-type limitation and immediate nodes; 2) enhance the efficiency for computing path feature values by exploiting parallel or distributed computing.

Acknowledgements

The authors are grateful for the support of the National Natural Science Foundation of China (No. 61572228, No. 61472158, No. 61300147, No. 61602207 and No. 61572225), the United States National Institutes of Health (NIH) Academic Research Enhancement Award (No.1R15GM114739), the National Institute of General Medical Sciences (NIH/NIGMS) (No.5P20GM103429), the Science Technology Development Project from Jilin Province (No. 20160101247JC), the Premier-Discipline Enhancement Scheme supported by the Zhuhai Government and the Premier Key-Discipline Enhancement Scheme supported by Guangdong Government Funds.

Bibliography

[1] Agirre, E.; Lacalle, O.; Soroa, A. (2014); Random walks for knowledge-based word sense disambiguation, *Computational Linguistics*, 40, 57–84, 2014.

[2] Berant, J.; Chou, A.; Frostig, R.; Liang, P. (2013); Semantic parsing on Freebase from question-answer pairs, *Proceedings of EMNLP*, 1533–1544, 2013.

[3] Bollacker, K.; Evans C.; Paritosh, P.; Sturge, T.; Taylor, J. (2008); Freebase: a collaboratively created graph database for structuring human knowledge, *Proceedings of KDD*, 1247–1250, 2008.

[4] Bordes, A.; Usunier, N.; García-Durán, A.; Weston, J.; Yakhnenko O. (2013); Translating embeddings for modeling multi-relational data, *Proceedings of NIPS*, 2787–2795, 2013.

[5] Cao, F.; Liu, B.; Park, D. (2013); Image classification based on effective extreme learning machine, *Neurocomputing*, 102, 90–97, 2013.

[6] Carlson, A.; Betteridge, J.; Kisiel, B.; Settles, B.; Hruschka, E.; Mitchell T. (2010); Toward an architecture for never-ending language learning, *Proceedings of AAAI*, 1306–1313, 2010.

[7] Gardner, M.; Talukdar, P.; Kisiel, B.; Mitchell, T. (2013); Improving learning and inference in a large knowledge-base using latent syntactic cues, *Proceedings of EMNLP*, 833–838, 2013.

[8] Gardner, M.; Talukdar, P.; Krishnamurthy, J.; Mitchell, T. (2014); Incorporating vector space similarity in random walk inference over knowledge bases, *Proceedings of EMNLP*, 833–838, 2014.

[9] Gardner, M.; Mitchell, T. (2015); Efficient and expressive knowledge base completion using subgraph feature extraction, *Proceedings of EMNLP*, 1488–1498, 2015.

[10] Getoor, L.; Taskar, B. (2007); Introduction to statistical relational learning, MIT press, 2007.

[11] Glassick, C.E.; Huber, M.T.; Maeroff, G.I. (2015); DBpedia - A large-scale, multilingual knowledge base extracted from Wikipedia, *Semantic Web*, 6, 167–195, 2015.

[12] Guo, S.; Wang, Q.; Wang, B.; Wang, L.; Guo, L. (2015); Semantically smooth knowledge graph embedding, *Proceedings of ACL*, 84–94, 2015.

[13] Hoffmann, R.; Zhang, C.; Ling, X.; Zettlemoyer, L.; Weld, D. (2011); Knowledge-based weak supervision for information extraction of overlapping relations, *Proceedings of ACL*, 541–550, 2011.

[14] Huang, G.; Wang, D.; Lan, Y. (2011); Extreme learning machines: a survey, *International Journal of Machine Learning and Cybernetics*, 2, 107–122, 2011.

[15] Huang, G.; Zhou, H.; Ding, X.; Zhang, R. (2012); Extreme learning machine for regression and multiclass classification, *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42, 513–529, 2012.

[16] Huang, G.; Zhu, Q.; Siew, C. (2006); Extreme learning machine: theory and applications, *Neurocomputing*, 70, 489–501, 2006.

[17] Lanckriet, G.; Cristianini, N.; Bartlett, P.; Ghaoui, L.; Jordan, M. (2004); Learning the kernel matrix with semidefinite programming, *Journal of Machine Learning Research*, 5, 27–72, 2004.

[18] Landwehr, N.; Kersting, K.; Raedt, L. (2005); nFOIL: Integrating naıve bayes and FOIL, *Proceedings of AAAI*, 795–800, 2005.

[19] Lao, N.; Mitchell, T.; Cohen, W. (2011); Random walk inference and learning in a large scale knowledge base, *Proceedings of EMNLP*, 529–539, 2011.

[20] Lao, N.; Minkov, E.; Cohen, W. (2015); Learning relational features with backward random walks, *Proceedings of ACL*, 666–675, 2015.

[21] Lao, N.; Subramanya, A.; Pereira, F.; Cohen, W. (2012); Reading the web with learned syntactic-semantic inference rules, *Proceedings of EMNLP*, 1017–1026, 2012.

[22] Lao, N.; Mitamura, T.; Mitchell, T.; Zuo, W. (2012); Efficient random walk inference with knowledge bases, *PhD Thesis*, 2012.

[23] Lavrac, N.; Dzeroski, S. (1994), Inductive logic programming, *Proceedings of Workshop on Logic Programming*, 146–160, 1994.

[24] Lee, K.; Man, Z.; Wang, D.; Cao, Z. (2013); Classification of bioinformatics dataset using finite impulse response extreme learning machine for cancer diagnosis, *Neural Computing and Applications*, 22, 457–468, 2013.

[25] Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; Zhu, X. (2015); Learning entity and relation embeddings for knowledge graph completion, *Proceedings of AAAI*, 2181–2187, 2015.

[26] Ma, C.; OuYang J.; Chen, H.; Ji, J. (2016); A novel kernel extreme learning machine algorithm based on self-adaptive artificial bee colony optimisation strategy, *International Journal of Systems Science*, 47, 1342–1357, 2016.

[27] Nickel, M.; Murphy, K.; Tresp, V.; Gabrilovich, E. (2015); A review of relational machine learning for knowledge graphs, *Proceedings of IEEE*, 104, 11-33, 2015.

[28] Nickel, M.; Tresp, V.; Kriegel, H. (2011); A three-way model for collective learning on multi-relational data, *Proceedings of ICML*, 809–816, 2011.

[29] Nickel, M.; Rosasco, L.; Poggio, T. (2016); Holographic embeddings of knowledge graphs, *Proceedings of AAAI*, 1955–1961, 2016.

[30] Niu, F.; Ré C.; Doan, A.; Shavlik, J. (2011); Tuffy: Scaling up statistical inference in markov logic networks using an rdbms, *Proceedings of the VLDB Endowment*, 4, 373–384, 2011.

[31] Quinlan, J. (1990); Learning logical definitions from relations, *Machine Learning*, 5, 239–266, 1990.

[32] Richardson, M.; Domingos, P. (2006); Markov logic networks, *Machine Learning*, 62, 107–136, 2006.

[33] Socher, R.; Chen, D.; Manning, C.; Ng, A. (2013); Reasoning with neural tensor networks for knowledge base completion, *Proceedings of NIPS*, 926–934, 2013.

[34] Su, L.; Yao, M. (2013); Extreme learning machine with multiple kernels, *Proceedings of ICCA*, 424–429, 2013.

[35] Suchanek, F.; Kasneci, G.; Weikum, G. (2007); Yago: a core of semantic knowledge, *Proceedings of WWW*, 697–706, 2007.

[36] Wang, Q.; Mao, Z. Wang, B.; Guo, L. (2017); Knowledge graph embedding: a Survey of approaches and applications, *IEEE Transactions on Knowledge and Data Engineering*, 2724–2743, 2017.

[37] Wang, W.; Mazaitis, K.; Cohen, W. (2013); Programming with personalized pagerank: a locally groundable first-order probabilistic logic, *Proceedings of CIKM*, 2129–2138, 2013.

[38] West, R.; Gabrilovich, E.; Murphy, K.; Sun, S.; Gupta, R.; Lin, D. (2014); Knowledge base completion via search-based question answering, *Proceedings of WWW*, 515–526, 2014.

[39] Zheng, W.; Qian, Y.; Lu, H. (2013); Text categorization based on regularization extreme learning machine, *Neural Computing and Applications*, 22, 447–456, 2013.

[40] Zong, W.; Huang, G. (2011); Face recognition based on extreme learning machine, *Neurocomputing*, 74, 2541–2551, 2011.