Knowledge Base Completion by Learning to Rank Model

Yong Huang[1,2] and Zhichun Wang[1,2](✉)

[1] Beijing Advanced Innovation Center for Future Education, Beijing Normal University, XinJieKouWai St. 19, HaiDian District, Beijing 100875, People's Republic of China
yhuang@mail.bnu.edu.cn, zcwang@bnu.edu.cn
[2] College of Information Science and Technology, Beijing Normal University, XinJieKouWai St. 19, HaiDian District, Beijing 100875, People's Republic of China

Abstract. Knowledge base (KB) completion aims to predict new facts from the existing ones in KBs. There are many KB completion approaches, one of the state-of-art approaches is Path Ranking Algorithm (PRA), which predicts new facts based on path types connecting entities. PRA treats the relation prediction as a classification problem, and logistic regression is used as the classification model. In this work, we consider the relation prediction as a ranking problem; learning to rank model is trained on path types to predict new facts. Experiments on YAGO show that our proposed approach outperforms approaches using classification models.

Keywords: Knowledge base completion · Path ranking
Learning to rank

1 Introduction

Recent years have witnessed a rapid growth of Knowledge Bases (KBs), such as YAGO [13], DBpedia [1], Google Knowledge Vault [4] etc. Usually, large-scale KBs are built automatically by extracting from the web text or other resources. KBs contain large amount of facts about various entities, and they are very useful to many applications such as Question Answering, biomedical information. Despite their large number of facts, they are still incomplete and missing a huge number of facts. To deal with this problem, many works have been done on KB completion, which aim to fill in missing facts by using existed facts to predict unknown facts.

Symbolic approaches use rules or relation paths to infer new facts in a KB. Lao [7] introduced the Path Ranking Algorithm (PRA), which uses random walks to search through bounded length of paths connecting entity pairs of multi-relational instances. These paths are used as features in a classifier that predicts new instances of the given relation. In PRA, each relation path can be viewed as a logic rule, therefore PRA actually is a kind of discriminatively trained logical inference [6].

J. Li et al. (Eds.): CCKS 2017, CCIS 784, pp. 1–6, 2017.
https://doi.org/10.1007/978-981-10-7359-5_1

Other KB completion methods such as embedding approaches [2,15] are also used recently. Embedding views KB completion problem as matrix completion, and learn low-dimensional representations of both entities and relations in KB, this can be used to infer new facts. There are also some work trying to combine symbolic technique and embedding technique most recently, including path-based TransE [9] etc.

This paper focuses on PRA and its extensions. PRA extracts relational path features to build classification models using logistic regression. It is based on local close world assumption (LCWA) [4] to generate negative entity pairs. However, in real KB there are too many negative entity pairs that positive and negative triples are extremely imbalance. What's more, KB completion use the candidate entity pairs to fill in the missing facts, ranking these candidates are much more reasonable instead of classifying or scoring these entity pairs. Here we consider a novel KB completion method by learning to rank model, and experiments show that this model is extremely beneficial for inferring new facts.

The article is structured as follows. We introduce our learning to rank model for KB completion in Sect. 2, and we report our experiments in Sect. 3, then review some symbolic related work in Sect. 4. Our conclusion and future works are in Sect. 5

2 Learning to Rank Model

Learning to rank model for KB Completion is a two step process: (1) generating feature matrix, (2) inferring new facts for each relation. In our approach, we follow the same method in PRA to extract relational features of entity pairs. While predicting new facts, we define pairwise objective function, with the purpose of improving entity rank rather than entity pair scores. And we use lambdaMART as a learning to rank model in comparison with PRA-style models, directly minimize mean average precision loss.

Feature Computing. Given a target relation r_i in a KB, we collect a set of entity pairs that have relation r_1, R_i is a set satisfying:

$$R_i = \{(h_{ij}, t_{ij}) | (h_{ij}, r_i, t_{ij}) \in KB\}$$

and then for R_i, we generate negative entity pairs following the Local Close World Assumption. For each entity pair (h_{ij}, t_{ij}), we perform random walk with restart to collect relational path types that connect (h_{ij}, t_{ij}) with a bounded path length. After random walk, we get a set of path types $P_i = \{p_i | (h_{ij}, p_i, t_{ij}) \in KB\}$. P_i collects the existed relational path types for relation r_i. These path types are used as relational features to make prediction of new entity pairs. We binarize these path type values to compute feature vector for each entity pair.

Entity Pairs Ranking. While predicting new facts in a KB, we are inspired by learning to rank approaches [3]. If real entity pairs exist in a KB, their rank should be higher than these fake entity pairs. Hence we should learn entity rank instead of learning scores of entity pairs. We explore different learning to rank methods to build pairwise objective function in this paper. Providing a group of entity pairs, we consider $(h_{ij}, t_{ij}) = 1$ if $h_{ij}, t_{ij} \in KB$ else $(h_{ij}, t_{ij}) = -1$. If there is a query for $(h_{ij}, r_i, ?)$, we generate a list of ranked tail entity, and the positive pairs are considered to better fit the relation while the negative pairs are not.

We take a state-of-art boosting tree method called gradient boosting tree or lambdaMART as our learning to rank approach. This method takes positive and negative entity pairs as a partial order. Instead of minimizing the log loss function of logistic regression, we directly optimize MAP using gradient boosting to learning entity pairs' rank. The output of LambdaMART can be defined as

$$L(r_i|w,c) = \sum_{i=1}^{K} \alpha_i f_i(x) + \sum_{i=1}^{n} l(f(x), f(\hat{x})) + \frac{C}{2} w^T w$$

where in the first part of the formula each $f_i(x)$ is a function modeled by a single tree, α_i is the learned weight with ith tree and K is the number of trees. The second part of is pairwise loss measuring the distance between predicted values and actual values. And the third part is regularization. L2 penalty on leaf values are added to avoid over-fitting. C is constant. Stochastic gradient boosting is used to improve accuracy on each iteration of a base tree.

3 Experiments

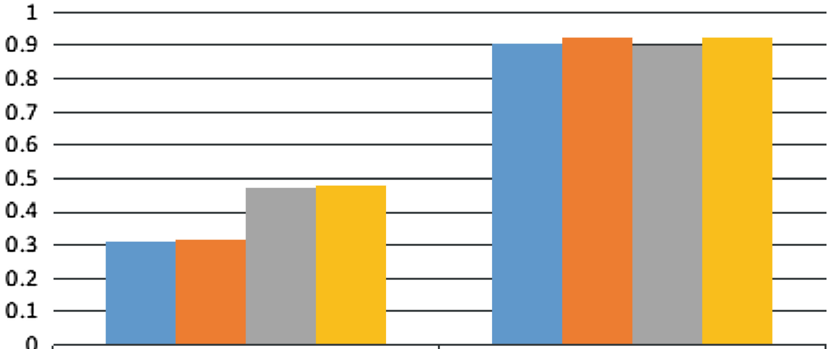
We compare our evaluation results of learning to rank methods with that of PRA and SFE [5] in YAGO KB completion task. SFE is another PRA-style KB completion method, it uses subgraph feature extraction to compute much more comprehensive path type features. The baseline model can be found here[1]. Both PRA and learning to rank method use this baseline model to extract path features.

Data. To evaluate our approach, we used the data from YAGO knowledge base. YAGO is built automatically from Wikipedia, GeoNames, and WordNet. Currently, YAGO2 contains 37 kind of relations, more than 10 million entities and more than 120 million facts about these entities. The YAGO2 data can be found here[2]. For each entity pair in KB, we randomly generate eight negative entity pairs using LCWA. For example, an entity pair$(Beijing, isCapitalOf, China)$ can generate a fake entity pair$(Shanghai, isCapitalOf, China)$ and many others. 37 relations are tested in our experiments. There are 124597 entity pairs on average for training, and 21810 for testing.

As we state above, if we use Logistic Regression in PRA and SFE, it's approximated by a learning to rank problem using binary classification. Hence we use LR as PRA and SFE's baseline with python framework[3] to train LR and learning

[1] https://github.com/matt-gardner/pra.
[2] http://www.mpi-inf.mpg.de/.
[3] http://scikit-learn.org/stable.

to rank model. We set L2 parameter in LR is a range from [0.01,1]. For learning to rank method we set lambdaMART tree size 1000. We set the same L2 parameter as LR in lambdaMART tree. We set C a default value 1. We present the optimal result as the experimental results in this paper.



	MAP	MRR
PRA	0.3083	0.9048
SFE	0.3127	0.9214
rankPRA	0.4691	0.8962
rankSFE	0.4733	0.9185

Fig. 1. Evaluation results: MAP and MRR

Result. We use mean average precision (MAP) and mean reciprocal rank (MRR) as metrics, which are commonly used in KB completion tasks [5,14]. Both metrics effect evaluate ranking methods, if the true entity pairs rank before the fake entity pairs, the MAP and MRR result will get higher. We compare PRA, SFE with LR model and learning to rank model called rankPRA and rankSFE. The results are presented in Fig. 1. We can conclude that learning entity rank outperforms PRA and SFE significantly on MAP, and get comparable MRR score. In fact, tree methods get 35 kinds of relations' AP score that are higher than PRA and SFE. Only four relations scores are lower than PRA and SFE in RR score. Since tree method is nonlinear, it's difficult to output the learned weight of relational path.

We further explore all relations' AP scores, and results show most relations get significantly promoted than linear model. More than 20 relations' scores are higher than 0.5. Our model shows all relations scores in YAGO2. We can see that most relations get satisfying scores, while some relations get poor score like *imports* and *exports*. We analyze these relations and find triples like (Bangladesh, exports, wordnet_fertilizer_14859344) are difficult for PRA to lookup discriminative path, so we should find better path types in the future work. But for most relations, PRA feature computing can find expressive path feature, and learning to rank methods improve the MAP and MRR score greatly.

4 Related Work

We review symbolic KB completion methods in this section. Symbolic KB completion methods assumes that the existence of an edge can be predicted by extracting features from observed edges in the graph. Predict the triple (Beijing, isLocatedIn, China)from the existence of path (Beijing, hasUniversity, Tsinghua_University, isLocatedIn, China). Other methods like TransE etc. low-dimensional representations approaches are excluded in this paper.

Inductive Logic Programming and Rule Mining. Inductive logic programming (ILP) uses logic programming as a uniform representation to infer facts. Rule mining methods such as Wang [16] learn rules by mining frequent predicate cycles in KB. ILP [10] was Firstly introduced in 1991, another ILP system named First Order Inductive Logic (FOIL) was proposed by Quinlan [12]. It constructs horn clause programs from training examples. For example, isMarriedTo(a,b) $\bigwedge$ hasChild(a,c) $\Rightarrow$ hasChild(b,c) can be used to infer the fact that $hasChild(DonaldTrump, IvankaTrump)$, if we have learned the rule with variables a, b, c bound to IvanaMarieTrump, DonaldTrump, IvankaTrump. These systems use open world assumption and are easily interpretable, owing to the learned rules only cover a subset of patterns.

Path Ranking Algorithm. PRA [8] uses random walks through bounded length of paths to predict links in multi-relational graphs, and discovers paths by enumerating each relation's entity pairs. The key idea of PRA is using path probabilities as features to build classification methods and predicting hidden relations. PRA can get much more precisely path types compared to ILP, and get comparable performance to embedding methods [11].

PRA Extensions. Many other symbolic methods have been proposed. Subgraph Feature Extraction (SFE) proposed by Gardner [5] is another PRA-style approach. Instead of computing path type probabilities, SFE binarize it as features. This approach can not only be more efficient, but also gain more expressive path features [6] leading better performance. Wang [14] proposed a multi-task learning strategy to cluster some highly correlated relation before using PRA, referred as CPRA. CPRA can effectively identify coherent clusters whose relations are highly correlated.

5 Conclusion

This paper proposes a novel approach to solve KBs completion problem. We follow PRA and apply learning to rank method to rank entity pairs in YAGO. Experiments show that our approach performs better than PRA and SFE. More expressive features can be added to PRA feature computing in the future work, and we will explore new ranking methods to improve symbolic KB completion performance.

Acknowledgement. The work is supported by project of Beijing Advanced Innovation Center for Future Education (BJAICFE2016IR-002).

References

1. Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S.: Dbpedia-a crystallization point for the web of data. Web Semant. Sci. Serv. Agents World Wide Web **7**(3), 154–165 (2009)
2. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in Neural Information Processing Systems, pp. 2787–2795 (2013)
3. Cao, Z., Qin, T., Liu, T.-Y., Tsai, M.-F., Li, H.: Learning to rank: from pairwise approach to listwise approach. In: Proceedings of the 24th International Conference on Machine Learning, pp. 129–136. ACM (2007)
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: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 601–610. ACM (2014)
5. Gardner, M., Mitchell, T.: Efficient and expressive knowledge base completion using subgraph feature extraction. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1488–1498 (2015)
6. Gardner, M., Talukdar, P.P., Krishnamurthy, J., Mitchell, T.: Incorporating vector space similarity in random walk inference over knowledge bases. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (2014)
7. Lao, N., Cohen, W.W.: Relational retrieval using a combination of path-constrained random walks. Mach. Learn. **81**, 53–67 (2010)
8. Lao, N., Mitchell, T., Cohen, W.W.: Random walk inference and learning in a large scale knowledge base. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, pp. 529–539. Association for Computational Linguistics, Stroudsburg (2011)
9. Lin, Y., Liu, Z., Luan, H.-B., Sun, M., Rao, S., Liu, S.: Modeling relation paths for representation learning of knowledge bases. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2015) (2015)
10. Muggleton, S.: Inductive logic programming. New Gener. Comput. **8**(4), 295–318 (1991)
11. Nickel, M., Murphy, K., Tresp, V., Gabrilovich, E.: A review of relational machine learning for knowledge graphs. Proc. IEEE **104**, 11–33 (2016)
12. Quinlan, J.R., Cameron-Jones, R.M.: FOIL: a midterm report. In: Brazdil, P.B. (ed.) ECML 1993. LNCS, vol. 667, pp. 1–20. Springer, Heidelberg (1993). https://doi.org/10.1007/3-540-56602-3_124
13. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a large ontology from wikipedia and wordnet. Web Semant. Sci. Serv. Agents World Wide Web **6**(3), 203–217 (2008)
14. Wang, Q., Liu, J., Luo, Y., Wang, B., Lin, C.-Y.: Knowledge base completion via coupled path ranking. In: ACL (2016)
15. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI 2014), pp. 1112–1119 (2014)
16. Wang, Z., Li, J.-Z.: Rdf2rules: learning rules from RDF knowledge bases by mining frequent predicate cycles. CoRR, abs/1512.07734 (2015)