Path Ranking with Path Difference Sets for Maintaining Knowledge Base Integrity

Po-Cheng Huang, Hen-Hsen Huang, and Hsin-Hsi Chen
Department of Computer Science and Information Engineering
National Taiwan University, Taipei, Taiwan
pchuang@nlg.csie.ntu.edu.tw, hhhuang@nlg.csie.ntu.edu.tw, hhchen@ntu.edu.tw

ABSTRACT

Knowledge base completion (KBC) involves in discovering missing facts. However, knowledge changes over time. Some facts need to be removed from knowledge base (KB) to keep knowledge base integrity (KBI) while new facts are inserted or old facts are deleted. This paper proposes a path-based learning model to learn the dependency of dynamic relations automatically. In this way, we can eliminate the conflicting facts and keep KB clean. That would be a significant benefit for KBC and other tasks using KB.

CCS CONCEPTS

• **Information systems→ Information retrieval→ Retrieval tasks and goals→ Information extraction**

KEYWORDS

Knowledge base completion; knowledge base integrity; path ranking

1 INTRODUCTION

Large-scale knowledge bases (KB) like YAGO, DBpedia, and Wikidata provide useful structured information to many NLP tasks such as question answering and relation extraction. Even though KBs contain large collection of facts, they suffer from two major challenges: incompleteness and noisy. In recent years, a number of researches have shown to discover missing facts from existing KB itself, known as the knowledge base completion (KBC). Approaches to this task can be roughly divided into two categories: (i) path-based method such as path ranking algorithm (PRA) [3]; (ii) embedding-based method such as TransE [1].

ACM Reference format:

P.C. Huang, H.H. Huang, and H.H. Chen. 2018. Path Ranking with Path Difference Sets for Maintaining Knowledge Base Integrity. In *The 2018 Web Conference Companion (WWW 2018), April 23-27, 2018, Lyon, France*, ACM, New York, NY, 2 pages. DOI: https://doi.org/10.1145/3184558.3186932

WWW '18 Companion April 23-27, 2018, Lyon, France.

ACM ISBN 978-1-4503-5640-4/18/04.
DOI: https://doi.org/10.1145/3184558.3186932

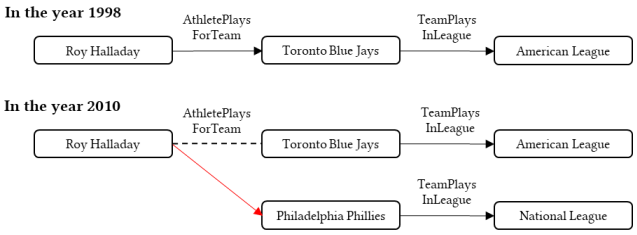

Figure 1: An example of dynamic relation.

Despite the high attention, KBs still contain noise such as mistaken facts or out-of-date facts. Noise affect the performance of KBC task and other applications. Takaku et al. [4] categorize relation types from two perspectives: (i) unique or non-unique; (ii) dynamic or static. These two perspectives give us some inspirations to remove noise and keep knowledge base integrity (KBI). Consider an example shown in Fig. 1. KB in 1998 contains two facts (*Roy Halladay, AthletePlaysForTeam, Toronto Blue Jays*) and (*Toronto Blue Jays, TeamPlaysInLeague, American League*). From them, we can infer (*Roy Halladay, AthletePlaysInLeague, American League*) and add this missing fact to the KB. In 2010, *Halladay* moves to another team. His belonging league is implicitly changed accordingly, i.e., National League. Because it was not mentioned in the real world, the out-of-date fact remains in the KB and confuses the KBC approach. Thus, removing the dependent dynamic relations to keep the KB clean is indispensable.

2 KNOWLEDGE BASE INTEGRITY

2.1 Problem Specification

A KB contains a collection of facts in the form of triples $\mathcal{T} = \{(h, r, t)\}$. Each triple is composed of a head entity (subject) $h \in \mathcal{E}$, a tail entity (object) $t \in \mathcal{E}$ and a relation (property) $r \in \mathcal{R}$, where $\mathcal{E}$ is an entity set and $\mathcal{R}$ is a relation set. KBI is an inverse operation of KBC. Instead of inferring and adding missing facts, we identify the modifications of dynamic relations in KB and remove the conflicting facts.

We dump multiple versions of KBs at different times, and collect the deletion triples $\mathcal{D} = \mathcal{T}_0 - \mathcal{T}_1$ and insertion triples $\mathcal{I} = \mathcal{T}_1 - \mathcal{T}_0$ where $\mathcal{T}_1$ is the same KB released after $\mathcal{T}_0$. The deletion triples with relation r are used to train a relation-specific classifier for r to decide whether a triple $(h, r, t) \in \mathcal{T}$ should be removed.

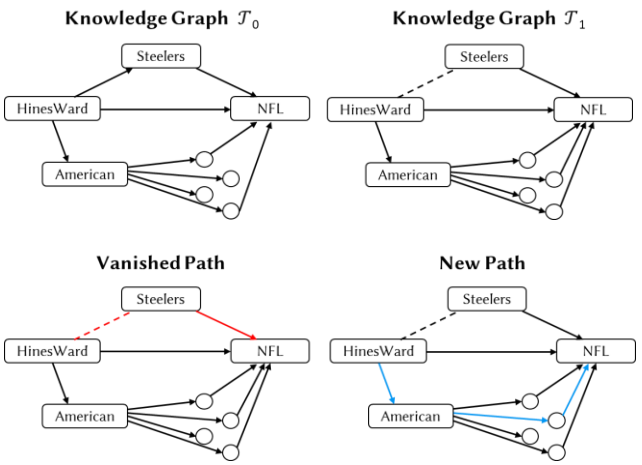

Figure 2: Path difference between two versions of KB.

2.2 Path Ranking with Path Difference Sets

The key idea of the PRA for KBC [3] is to enumerate all the paths between each entity pair with relation r in a KB and then use these paths as features to train a classifier to discover the missing relation r between an entity pair. PRA deals with the KBC task on a snapshot of a KB. In contrast, we focus on the differences between two versions (say, $\mathcal{T}_0$ and $\mathcal{T}_1$) of a KB in KBI task.

Given a relation r, we collect all the paths with maximum length 3 from $\mathcal{T}_0$ and $\mathcal{T}_1$ and place them into $\mathcal{P}_{1r}$ and $\mathcal{P}_{2r}$, respectively. We get two path difference sets: vanished path set $\mathcal{V}_r = \mathcal{P}_{1r} - \mathcal{P}_{2r}$ and new path set $\mathcal{N}_r = \mathcal{P}_{2r} - \mathcal{P}_{1r}$. These two types of paths are encoded as the features for our path-ranking model.

Consider the example in Fig. 2. The fact removed from $\mathcal{T}_1$ is shown in dashed line. Some paths connected between the target entity pair, i.e., (HinesWard, NFL), are interrupted or connected due to the change of the relation. The vanished paths and the new paths are shown in red and in blue, respectively.

3 EXPERIMENTS

3.1 Dataset

The KB in this work is Wikidata, a collaboratively edited KB containing over 40 million entities. We collect the wikidata dumps, and extract the facts in wikidata into triples. We exclude the facts that have the qualifier "*end time*" because this label indicates a fact is out of date. We keep those facts whose entities or relations are mentioned at least 20 times in Wikidata. Besides, we only collect the triple that has at least one discrepant path passing through its entity pair for training and testing. That results in 6 editions from 2016/11 to 2017/07. Total 5,925 training triples and 2,936 test triples in 4 different dynamic relations are selected for testing.

3.2 Experimental Results

We train a binary classifier for each relation using the two path difference sets and logistic regression with 5-fold cross validation. We compare our model with TransE, which is a simple and powerful baseline and often applied to the KBC task. We use two measures of TransE@k, k = 1 and 10, which means the relation ranked in the top k positions is added to the KB. On the other hand, the relations after the k positions should be removed from the KB.

Table 1: Results of relation classification on 4 relations. The metrics are Precision (P), Recall (R), and F-score (F).

	Our method			TransE@1			TransE@10		
	P	R	F	P	R	F	P	R	F
League	0.80	0.52	0.63	0.00	0.00	0.00	0.00	0.00	0.00
member of	0.84	0.82	0.83	0.26	0.28	0.27	0.19	0.16	0.17
spouse	0.71	0.59	0.65	0.17	0.88	0.29	0.25	0.71	0.37
position held	0.48	0.40	0.43	0.34	0.24	0.28	0.45	0.13	0.20
average	0.63	0.56	0.59	0.28	0.26	0.27	0.32	0.16	0.21

Table 2: Top weighted paths.

Spouse	
Vanished	spouse^{-1}
New	unmarried_partner
League	
New	league → league_level_below
New	league → subclass_of → instance_of^{-1}

With the experimental setup, we consider TransE model as a relation classifier. Table 1 shows that our method outperforms TransE in terms of F-score. TransE achieves a higher recall on the relation *spouse*. Because the scoring functions of TransE: h + r ≈ t, TransE is unable to distinguish reflexive relations and just removes them. Our model can learn the reflexive relation as the most important feature because they must appear in pairs (h, r, t) and (t, r, h).

Other relation path examples are shown in Table 2. Our model learns that relation *unmarried_partner* conflicts with relation *spouse*. Moreover, when a person is moved to another level of league, his original league should be removed. We also evaluate the original PRA on the same dataset. PRA achieves an MAP of 0.49, while our model achieves a superior MAP of 0.68.

4 CONCLUSIONS

This paper introduces a new concept of KBI. The issue of KBI introduced by dynamic relations affects the performance of KBC and other applications. The proposed path ranking method with path difference sets can handle the chain reaction resulting from dynamic relations and keep KB clean. The conflicting relations such as *spouse* and *unmarried_partner* are ranked in higher feature weights in our model. Besides, we also show that the discrepant paths have the capability to simulate the KB refinement process made by machines or collaborators.

ACKNOWLEDGMENTS

This research was partially supported by Ministry of Science and Technology, Taiwan, under grants MOST-105-2221-E-002-154-MY3 and MOST-107-2634-F-002-011-.

REFERENCES

[1] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. In *NIPS*, 2013.
[2] M. Dragoni, C. Ghidini. Ontology evolution with semantic wikis. In *CAiSE*, 2012.
[3] N. Lao, T. Mitchell, and W. Cohen. Random walk inference and learning in a large scale knowledge base. In *EMNLP*, 2011.
[4] Y. Takaku, N. Kaji, N. Yoshinaga, and M. Toyoda. Identifying constant and unique relations by using time-series text. In *EMNLP*, 2012.