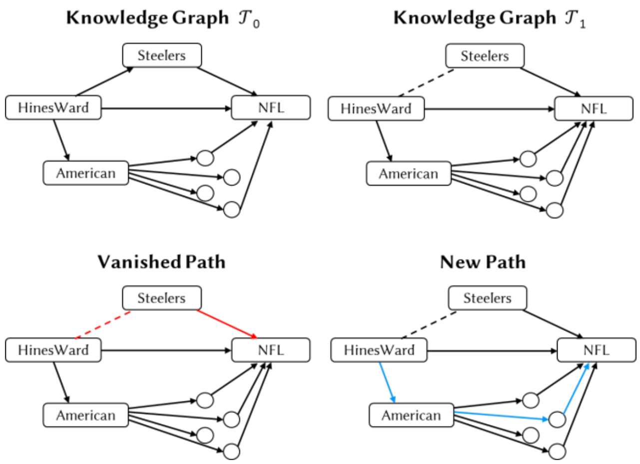

Figure 2: Path difference between two versions of KB.

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 We exclude the facts that have the qualifier "*end time*" because this label indicates a fact is out of date.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: $h + r \approx t$, 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.