JOINT INSTITUTE
交大密西根学院

VE444 • Networks
Group 7 Course Project

Named Entity Disambiguation in Enterprise Knowledge Graphs

Team Members: Zhanpeng Zhou, Yuchuan Tian, Haoxiang Jin

Instructor: Prof. Yifei Zhu

Overview

- Knowledge Graphs (KG) in enterprise field are powerful in investment analysis.
- Named ambiguation problem existed in KG impacts the precision of investment analysis.
- Our algorithm based on Strongly Connected Components (SCC), Node Embedding, and Graph Neural Networks (GNN) tackles the disambiguation problem.

Motivation

Knowledge graphs (KG) in enterprise field are effective in solving financial problems.

Fig.1 Knowledge Graphs in Enterprise Field

But when constructing an enterprise knowledge graph based on raw data, ambiguity, namely, two entities sharing the same name, hinders analysis precision. Hence we try to explore an efficient name disambiguation technique.

Problem Statement

Named entity disambiguation is formulated as finding the similarity between nodes in the enterprise KG.

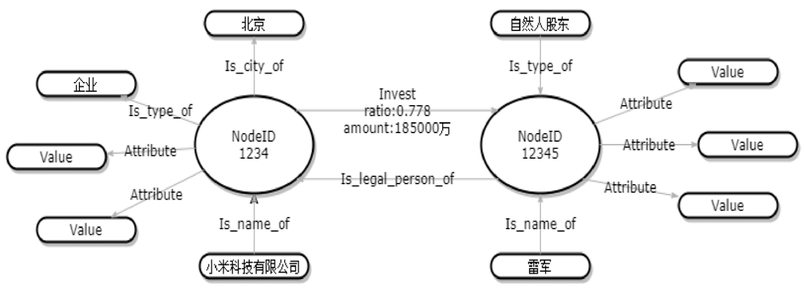

Fig.2 A Simple Example of Enterprise KG

1. Construct a directed Enterprise KG.

$$\mathcal{G} = (C, I, E)$$
$$C = \{a | a \in \text{companies}\}$$
$$I = \{b | b \in \text{investors}\}$$
$$E = \{(b, a), (a, c) | a, c \in C, b \in I\}$$

2. The named entity disambiguation problem then becomes the binary classification problem.

Classifying the entities with the same name into same/different entity group is based on the binary classifier:

$$node\_classifier(a, b) \text{ where a, b} \in \text{I}$$

This classifier returns true when two nodes are judged as the same node.

Algorithm Design

Our algorithm include two baseline methods, and two advanced methods (**Node2Vec, GNN**) with respect to them.

- **Baseline Classification**:

1. Always True: classify entity with the same name into the same entity group;
2. SCC Algorithm: divide all ties in the KG into strong/weak ties, and classify same-name entities within same group as "same entity".

- **Node2Vec Classification**: convert nodes into low-dimensional vectors based on Random Walk, which can extract local and global graph structure, and then classify entities based on simple similarity functions.

$$node\_classifier(a, b) \rightarrow a.name == b.name \;\&\; cos(Node2Vec(a), Node2Vec(b)) > \theta$$

- **GNN Classification**: learn a hidden state $\mathbf{h_v}$ of node **v** containing the info of **v**'s neighborhood via gradient descent and cross entropy loss, and output the classification result o_{ij} for node pair $i\&j$.

Denote:

y -- the actual binary classification

p -- the anticipated probability of the sample belonging to class "1"

$\mathbf{F}$ -- a fully connected layer outputting binary values

f -- a local transition function $\mathbf{x_v}$ -- the features of node **v**

$\mathbf{x_{\{co[v]\}}}$ -- the features of **v**'s edges

$\mathbf{h_{\{ne[v]\}}}, \mathbf{x_{\{ne[v]\}}}$-- the states, features of the nodes in the neighborhood of **v**

Then:

$$Cross\_Entropy\_Loss = -[y \cdot \log(p) + (1 - y) \cdot \log(1 - p)]$$
$$\mathbf{F}(\|\mathbf{h}_i - \mathbf{h}_j\|_2) = Binary\_classification\_output$$
$$\mathbf{h}_v = f(\mathbf{x}_v, \mathbf{x}_{co[v]}, \mathbf{h}_{ne[v]}, \mathbf{x}_{ne[v]})$$
$$o_{ij} = g(\mathbf{h}_i, \mathbf{h}_j, \mathbf{x}_i, \mathbf{x}_j)$$

Results

Evaluation Metrics: given

- **TP:** # of pairs of same entities judged as "pair of same entities";
- **FP:** # of pairs of different entities with same names judged as "pair of same entities";
- **FN:** # of pairs of different entities with same names judged as "pair of different entities".

Then precision & recall, defined as

$$precision = \frac{TP}{TP + FP}$$
$$recall = \frac{TP}{TP + FN}$$

are our evaluation metrics.

Evaluation Results:

Evaluation results of all classification methods are shown in **Fig. 3**:

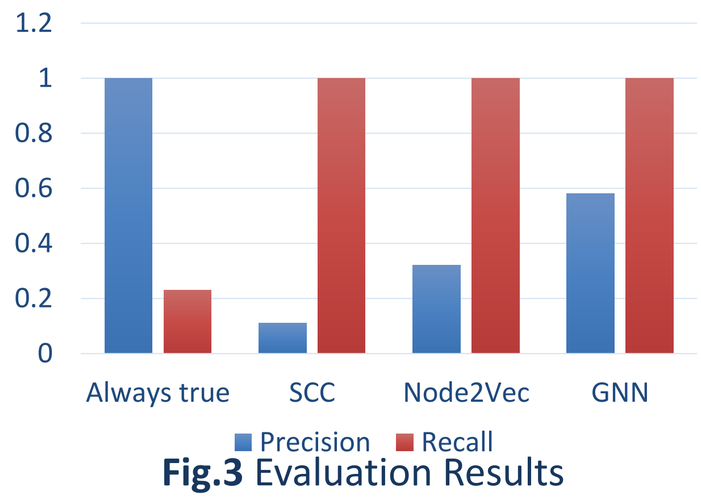

Fig.3 Evaluation Results

Conclusion:

- After applying different algorithms in this project, we find that graph structures could improve disambiguation results
- GNN gives the state of the art results, which leverage not only the graph structure info, but extra features.

References

1. Grover, Aditya, and Jure Leskovec. "node2vec: Scalable feature learning for networks." Proceedings of the 22nd ACM SIGKDD, 2016.

Contact

If you have any questions, please contact yifei.zhu@sjtu.edu.cn.