Abstract

As knowledge graphs become a hot trend in managing and exploiting information, there are lots of research conducted by both researchers and companies in order to build knowledge graphs from data. However, most of the research work lies in the area of link prediction and knowledge graphs completion, which usually requires given information from domain experts in the form of existing ontologies or uncompleted knowledge bases. In this work, our goal is to exploit the tremendous amount of web data to build the knowledge graphs in an unsupervised manner. Instead of focusing on fully unstructured data, we plan to extract the semi-structure information of web tables to reduce the uncertainty and noisiness of the data.

1 Graph embeddings

In this section, we explain three most common methods for knowledge graphs embeddings: TransE (Bordes et al., 2013), TransH (Wang et al., 2014) and TransR (Lin et al., 2015).

1.1 TransE

Bordes et al (Bordes et al., 2013) proposed one of the earliest embedding methods for knowledge graphs. All triple components (s, p, o) are embedded in a same vector space. The main idea of TransE is that relation p in a triple can be understood as a translation of the subject s to the object o. Therefore, TransE minimizes the distance between (s + p) and o in the vector space. The overall objective function of TransE is shown as below:

\[ L = \sum_{(s, p, o) \in S} \sum_{(s', p, o') \in S'} [d(s + p, o) - d(s', p, o')] \]

As we can see, TransH can be considered as an extension of TransE, which provides flexibility for entity representations based on the involving relations. However, the main idea is not very different to TransE and both of these approaches have re-
lations as the key element in their objective functions, which we point out the weaknesses in ??.

1.3 TransR

In TransR, instead of mapping both entities and relations to a same vector space, Lin et al (Lin et al., 2015) proposed an embedding approach where entities and relations are mapped into two separate vector spaces. Since entities and relations are two different concepts, TransR claims that this separation would allow users to extend the capabilities of knowledge graphs embedding. The objective function is computed as follows with $s_M$ and $o_M$ are the projections of $s$ and $o$ to another vector space using the the projection $M_p$:

$$L = \sum_{(s,p,o) \in S} \sum_{(s',p,o') \in S'} \left[ d(s_{\perp} + p, o_{\perp}) - d(s'_{\perp}, p, o'_{\perp}) \right]$$

As we can see here, even though TransR has been developed to address the issues to TransE and TransH, all three approaches rely heavily on relations to learn the embedding and thus cannot be applied in problems where the relation sets are not given or are given in a noisy and messy data.

2 Table Classification

Web tables can be divided into three different categories: relational tables, entity tables and matrix tables. Relational tables describe a set of entities with some of its attributes. Each row in a relational table represents an entity while each column contains values of one attribute. On the other hand, entity tables show information of one entity where each individual row shows different attribute value. The least common type in the three categories, matrix tables, is mostly used in statistical analysis. Examples of these three table types are shown in Figure ??.

Since our project focuses on the building knowledge graphs from web tables, we only consider relational tables as our target table types. The main reason for our choice is that relational tables represent the relationships between entities and their attributes, which are similar to the relations between elements of triples in knowledge graphs: subjects, predicates and objects. Entity tables also contain these types of information. However, since entity tables are usually semi-structured, it is harder to extract information. Therefore, we do not use entity tables in our project.

2.1 Problem Definition

In the problem of building knowledge graphs from web tables, table clustering is a very important task. Web tables are available in most of internet web pages and it becomes extremely difficult to decide whether two tables have similar topic due to the high level of diversity. In the table clustering step, our system aims to group tables that have similar topics together. Based on the clustering results, we can choose a cluster of similar tables and build knowledge graphs from these sets of tables.

The formal definition of table definition is given below:

**Definition 2.1** Given a set of tables $T = \{T_1, T_2, \ldots, T_n\}$, find a set of clusters $C = \{C_1, C_2, \ldots, C_m\}$ from $T$, where $C_i$ is a collection of tables that contains similar information or has a same topic.

2.2 Previous Work

According to our research, there is no previous work that focus on the problem of table clustering or table classification. However, in the field of document clustering, there are a tremendous amount of research that have been conduct in recent years. One of the main approaches for document clustering is using vector space and bag-of-words and tfidf indexing are usually used to convert documents into vectors. However, these methods have a weakness that they only consider similarities in lexical values instead of their semantics, which are not ideal, especially in textual contexts. Latent Dirichlet Allocation (Blei et al., 2003) is another common method in text clustering as it focuses on topics of words and thus provide a more semantic result compared with tfidf and bags of words.

2.3 Our current approach

In the current stage of the project, we propose a simple approach based on the foundation of text clustering for our table clustering. Our approach is described as follows:

- Remove numeric values from the tables.
- Convert columns into documents by concatenating their values.
The distances between two documents is computed by the average distance of their columns.

Based on the distance between these documents, we group documents of which distances are less than a given threshold.

In later stages, we plan to apply a previous approach on short-text clustering (Xu et al., 2015) into our table clustering problem.

3 Experiments

Our experiments focus on two parts, running and analyzing the results of various knowledge graph embedding approaches on benchmark datasets, and creating our own domain-specific datasets based on a corpus of web tables.

3.1 Graph Embeddings

In previous sections, we have mentioned several graph embedding approaches which have achieved good performance on some benchmark datasets. But because of their inconsistent implementations, it is hard to compare them. To solve the problem, (Han et al., 2018) released an open toolkit for knowledge embeddings, which provides a unified framework and various state-of-art models.

Our experiments concentrate on two benchmark datasets FB15K and WN18, both of them are widely used as benchmark datasets for link prediction. FB15K is a subgraph from Freebase and WN18 is a subgraph from WordNet.

Our experiments run on CPU with 2.9 GHz Intel Core i5 processor. The parameter settings for all experiments are listed in Table 1. The experimental results (MeanRank, MeanRank with filter, Hit@10 and Hit@10 with filter) of link prediction on benchmark datasets FB15K and WN18 are shown in Table 2. These evaluation metrics are selected in order to compare with the results in (Lin et al., 2015).

The experimental results show that for the both benchmark datasets, TransH performs slightly better than TransE in hit@10 and classification accuracy but worse in MeanRank. However, the results of TransR are worse than the results of other models.

Compared to the results provided in (Lin et al., 2015), our results of TransE and TransH are better possibly because of the improved implementation of OpenKE and different running environment. The results of TransR are worse because of the parameter settings and the machine itself. For the parameter settings, we want to get a baseline results of various models, thus we use the default settings for all models, which could be optimized in the next phase of our project. In addition, because of the less efficient computational performance of our machine, compared to the running time mentioned in (Han et al., 2018), it takes much longer time to run all models especially TransR, which has higher computational complexity.

In the next phase of this project, our plan for experiments is as follows

- Run all models with various parameter settings.
- Optimize the parameter settings and run our own datasets on these models.
- Analyze the performance of different models on our own datasets.
- Find the weakness of these models and make improvements.
<table>
<thead>
<tr>
<th>threads</th>
<th>epochs</th>
<th>number of batches</th>
<th>learning rate</th>
<th>margin</th>
<th>dimension</th>
<th>optimization method</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>500</td>
<td>100</td>
<td>0.001</td>
<td>1.0</td>
<td>100</td>
<td>SGD</td>
</tr>
</tbody>
</table>

Table 1: Parameter settings for all models

<table>
<thead>
<tr>
<th>Models</th>
<th>dataset</th>
<th>MeanRank</th>
<th>MeanRank(filter)</th>
<th>hit@10</th>
<th>hit@10(filter)</th>
<th>classification accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>TransE</td>
<td>FB15K</td>
<td>177</td>
<td>33</td>
<td>0.50</td>
<td>0.74</td>
<td>0.85</td>
</tr>
<tr>
<td></td>
<td>WN18</td>
<td>264</td>
<td>247</td>
<td>0.65</td>
<td>0.74</td>
<td>0.88</td>
</tr>
<tr>
<td>TransH</td>
<td>FB15K</td>
<td>184</td>
<td>33</td>
<td>0.50</td>
<td>0.75</td>
<td>0.88</td>
</tr>
<tr>
<td></td>
<td>WN18</td>
<td>316</td>
<td>302</td>
<td>0.79</td>
<td>0.91</td>
<td>0.96</td>
</tr>
<tr>
<td>TransR</td>
<td>FB15K</td>
<td>227</td>
<td>84</td>
<td>0.34</td>
<td>0.48</td>
<td>0.78</td>
</tr>
<tr>
<td></td>
<td>WN18</td>
<td>603</td>
<td>590</td>
<td>0.15</td>
<td>0.15</td>
<td>0.88</td>
</tr>
</tbody>
</table>

Table 2: Experimental results of different models on different datasets.

3.2 Web Tables

For web table data, we use web tables from WDC Web Table Corpus 2. The statistics of the dataset are shown as follows:

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of tables</td>
<td>460943228</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of attributes</td>
<td>460943228</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of entities</td>
<td>1271404585</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Web table statistics

Since it is hard to build a groundtruth dataset for our table clustering problem, we plan to evaluate our table clustering method after building the knowledge graph. The procedure of table clustering evaluation is illustrated as below:

- Cluster web tables using our table clustering method
- Build knowledge graph using these web tables
- Compute the ratio between the number of tables connected in the knowledge graphs and the total number of tables. This ratio reflects the connectivity between these tables and can be used as a measurement for our table clustering methods.

References


Xu Han, Shulin Cao, Lv Xin, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Juanzi Li. 2018. Openke: An open toolkit for knowledge embedding. In *Proceedings of EMNLP*.


Jiaming Xu, Peng Wang, Guanhua Tian, Bo Xu, Jun Zhao, Fangyuan Wang, and Hongwei Hao. 2015.

2http://websdatacommons.org/webtables/