Linking DBLP and OAG

Zhengxiong Yuan
zyuan@iastate.edu

Abstract

Linking entities from different sources is a fundamental task in building open knowledge graphs. Open academic graph (OAG) has already linked two academic graphs, Microsoft Academic Graph and Aminer. In this project, we are trying to link OAG and DBLP. There are at least three parts to link, venues, papers and authors. In this article, I am going to show you how to link the venues from these two knowledge graphs.

1 Introduction

Linking entities from different sources is a fundamental task in building open knowledge graphs. Despite much research conducted in related fields, the challenges of linking large-scale heterogeneous entity graphs are far from resolved. Fanjin Zhang et al.(2019) proposed a framework called LinKG to link two billion-scale academic entity graphs (Microsoft Academic Graph and AMiner). The linked graph is called Open academic graph (OAG). In this project, we are going to link DBLP and OAG. There are at least three parts to link, venues, papers and authors. I will show you how to link the venues with LinKG framework in this article.

The rest of the article is organized as follows. In Section 2, we introduce the two knowledge graphs that we are going to link. In Section 3, we present the venue linking framework. In section 4, we describe the datasets that we used in our experiments. In section 5, we present and analyze the experiment results. Finally, we give our conclusion and future work in Section 6.

2 Knowledge Graphs

In this project, we are trying to link two knowledge graphs, DBLP and OAG. DBLP is a computer science bibliography which provides open bibliographic information on major computer science journals and proceedings. Open Academic Graph (OAG) is a large knowledge graph unifying two billion-scale academic graphs: Microsoft Academic Graph (MAG) and AMiner.

2.1 DBLP

The dblp computer science bibliography is the on-line reference for bibliographic information on major computer science publications. It has evolved from an early small experimental web server to a popular open-data service for the whole computer science community. As of January 2019, dblp indexes over 4.4 million publications, published by more than 2.2 million authors. To this end, dblp indexes about than 40,000 journal volumes, more than 39,000 conference and workshop proceedings, and more than 80,000 monographs.

2.2 OAG

Open Academic Graph (OAG) is a large knowledge graph unifying two billion-scale academic graphs: Microsoft Academic Graph (MAG) and AMiner. In Jan. 2019, they published OAG v2, which contains 208,915,369 papers from MAG and 172,209,563 papers from AMiner and generated 91,137,597 linking (matching) relations between the two graphs. The statistics of OAG v2 is listed as the three tables below. The two large graphs are both evolving and they take MAG November 2018 snapshot and AMiner July 2018 or January 2019 snapshot for this version.

Dataset	#Pairs/Venues
linking relations	29,841
AMiner venues	69,397
MAG venues	52,678

Table 1: Statistics of OAG venue data.

Dataset	#Pairs/Papers
linking relations	91,137,597
AMiner papers	172,209,563
MAG papers	208,915,369

Table 2: Statistics of OAG paper data.

Dataset	#Pairs/Authors
linking relations	1,717,680
AMiner authors	113,171,945
MAG authors	253,144,301

Table 3: Statistics of OAG author data.

3 Venue Linking

Venues in academic graphs are word-sequence dependent entities. Given two venue names, one from OAG, the other from DBLP, we need to tell if they are the same venue. If the answer is yes, then we link these two venues together.

Pre-processing. Before linking, we firstly need to convert all of the venue names to lowercase.

Direct Name Matching. Assume that there are no two venues sharing the same name. Then, we could say that, if two venue names are exactly same, they must be the same venue.

Abbreviations. It turns out that many words of the venue names are represented by their abbreviations. For example, "transactions" is represented by "trans.", "communications" is represented by "commu.". To handle these cases, we need to identify if one word is the abbreviation of the other one. To do this, we firstly remove all "'s from the venue names. Then if one word starts with the other word, we will say that they represent the same word. For example, "acm computing surveys" and "acm comput. sur." are the same venue.

LSTM. For the venues that cannot be linked by direct name matching, we can use LSTM (Long Short-Term Memory) network. The

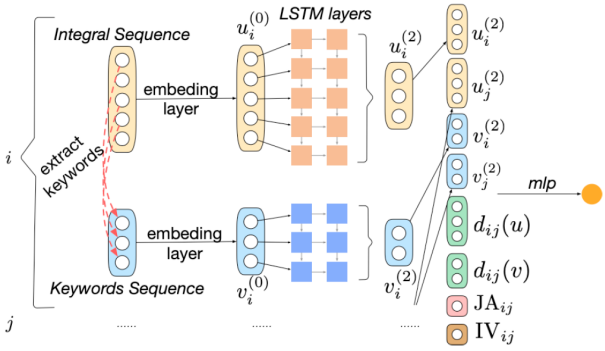

Figure 1: The LSTM model for modeling sequential dependency in venue full names.

reason is that the relative word or keyword sequences in the full name is generally preserved. Therefore, we propose to model both the Integral Sequence and Keyword Sequence in venue names. Integral Sequence is the raw word sequence from venue names, and Keyword Sequence is derived from the keywords extracted from the Integral Sequence. We present an enhanced long short-term memory networks (LSTM) to address the issues. The LSTM network is shown in Figure 1. Given a venue i in one graph, let $u_i^{(0)}$ be the input Integral Sequence and $v_i^{(0)}$ be the Keyword Sequence, we can have:

$$\begin{aligned} u_i^{(2)} &= lstm_{u2}(lstm_{u1}(u_i^{(0)})), \\ v_i^{(2)} &= lstm_{v2}(lstm_{v1}(v_i^{(0)})), \end{aligned} \tag{1}$$

where $lstm$ denotes the LSTM layers. Then we calculate the differences d_{ij} with venue j from the other graph:

$$\begin{aligned} d_{ij}(u) &= u_i^{(2)} - u_j^{(2)}, \\ d_{ij}(v) &= v_i^{(2)} - v_j^{(2)}, \end{aligned} \tag{2}$$

Finally, we concatenate all of them with Jaccard Index (JA_{ij}) and the number of inversion pairs in Keyword Sequence (IV_{ij}), and employ fully-connected layers to calculate the similarity between the two venues:

$$\begin{aligned} s_{ij} &= [u_i^{(2)}, v_i^{(2)}, u_j^{(2)}, v_j^{(2)}, d_{ij}(u), d_{ij}(v), JA_{ij}, IV_{ij}], \\ y_{ij} &= fc(s_{ij}), \end{aligned} \tag{3}$$

where fc denotes the fully-connected layers and y is the similarity between venue i and j. In training, we use venues from labeled candidate pairs to learn parameters in the LSTM

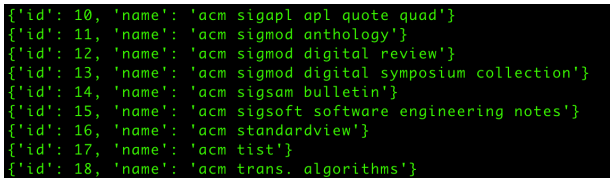

Figure 2: Extracted venue samples from DBLP.

Figure 3: Extracted venue samples from DBLP.

layers and fc layer; and in matching, we use the learned parameters to predict the similarity between two venues.

4 Experiments

4.1 Venue Extraction

The first step is to extract the venue names from the original data sets.

For OAG, we can download the venue data set from its official website. The venues are stored in a text file. Each venue is represented in a json format, including its id, journal id, full name, and normalized name. Note that not all of the venues from MAG and Aminer are linked in OAG. There is a file which contains the id's of the linked venues. So we can store the linked id's in a set, and scan MAG or AMiner linearly. If the venue exists in the set, which means that it belongs to OAG, then we extract this venue and store the id and its normalized name in file "oag_venues.txt".

For DBLP, the whole data set is stored in a huge XML file. Fortunately, its official site provides a Java package to parse the XML file. With the Java package, I wrote a Java program called "Parser.java" to extract all of the venue names. In addition, to identify different venues, I assigned an id for each venue.

The extracted venue samples are shown in Figure 2 and Figure 3.

In the end, we have extracted 27,599 venues from OAG and 1,437 venues from DBLP.

4.2 Direct Name Matching

In this step, we directly compare the full names of two venues, one from OAG, the other from DBLP. If the two venues have exactly the same name, then we link them together. For example, "{'id': 5, 'name': 'acm journal of computer documentation'}" from DBLP and "{'id': '5bf5740a1c5a1dcdd96f203c', 'name': 'acm journal of computer documentation'}" from OAG will be linked together at this step.

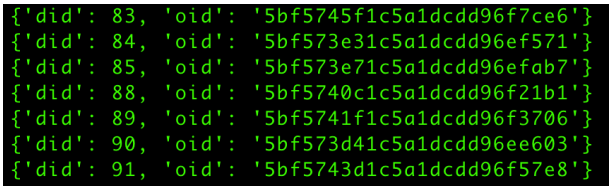

Figure 4: Samples of linked pairs

After this step, we got 312 linked pairs from the extracted two data sets.

4.3 Abbreviations

We notice that many words of the venue names are represented by their abbreviations. For example, "transactions" is represented by "trans.", "communications" is represented by "commu.". To handle these cases, we need to identify if one word is the abbreviation of the other one. To do this, we firstly remove all 's from the venue names. Then if one word starts with the other word, we will say that they represent the same word. For example, "{'id': '5bf5741c1c5a1dcdd96f340b', 'name': 'acm computing surveys'}" and "{'id': 3, 'name': 'acm comput. surv.'}" represent the same venue. So we will link them together.

After this step, we got 421 linked paris in total. The linked pair samples is shown in Figure 4.

4.4 LSTM

At this step, we use the LSTM network shown in 1. We firstly extract the keyword sequence and concatenate it with the original inegegral sequence. Then we go through two LSTM layers. Finally, we use a fully-connected layers to get the probability of the venues being the same one.

For this step, we directly used the pre-trained model from OAG, which was trained with about 1,000 manually labeled pairs.

For the testing, we manually chose 10 pairs from the two data sets, 5 positive paris and 5 negative pairs. The test paris is shown in Figure 5.

The result is shown in Figure 6.

As we can see from the result, the accuracy is terrible, which means that the pre-trained

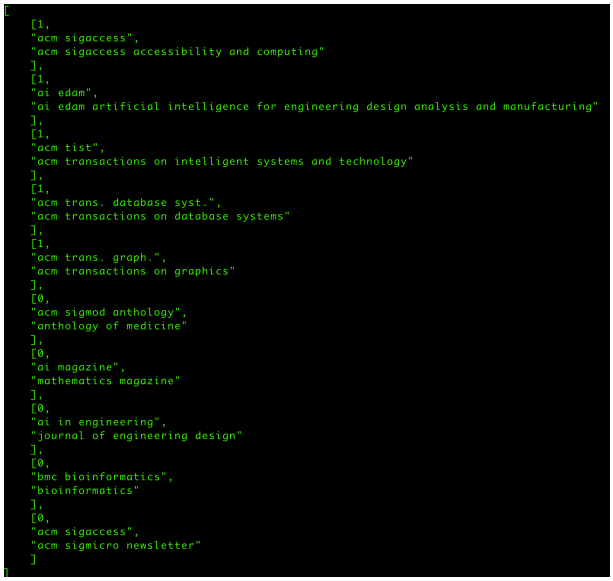

Figure 5: Test pairs

Figure 6: Test result

model doesn't suit our new data set. Creating a new test data set manually is very time-consuming, and due to limited time, I didn't have enough time to do that.

However, I looked into the test data and found that there are at least three issues which we can improve. The first one is abbreviation. Although we have linked some venues with abbreviations, there are many other cases with abbreviations. For example, "acm tist" and "acm transactions on intelligent systems and technology" are actually the same venue, but they are not identified as the same one. The second issue is the preposition words. For example, "acm trans. graph." and "acm transactions on graphics" are actually the same venue, but they are not identified as the same one due to the "on". The third issue is that keywords play too much role in short venue names. For example, both "bmc bioinformatics" and "bioinformatics" have the keyword "bioinformatics", so that they are very easily to be identified as the same venue.

5 Conclusion and Future Work

In summary, if we only use direct name matching and abbreviation pre-processing, we can only link 421 pairs out of 1,437 venues (intuitively, the actual pairs should be just a little bit less than 1,437, because OAG is much bigger than DBLP). In the future, we can design a new training set for linking DBLP. We can add more venue names with abbreviations to the training set. Also, we can add more negative short venue pairs that have the same set of keywords. The code for this project is published in my github. Instructions and further improvements will be added soon.

References

Fanjin Zhang, Xiao Liu, Jie Tang, Yuxiao Dong, Peiran Yao, Jie Zhang, Xiaotao Gu, Yan Wang, Bin Shao, Rui Li, and Kuansan Wang. 2019. OAG: Toward linking large-scale heterogeneous entity graphs. The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '19).