Collective Entity Resolution in Relational Data

Data Science Institute
COLUMBIA UNIVERSITY

Yiming Huang, Yiran Wang, Yijie Cao, Xinxin Hang, Kai Kang
Amir Rahmani, Jihan Wei
Tain Zheng, Sining Chen, Adam Kelleher

Data Science Capstone Project
with Capital One

Entity Reference Problem & Importance

The absence of identifying entities often results in data redundancy, inaccuracy in query processing and information extraction. Entity Resolution (ER) is the task of different manifestations of real-world entities in various sources by linking and grouping.

Traditionally, entities are resolved using pair-wise similarity resolution, in which entities for co-occuring references are determined jointly, rather than independently, can improve entity resolution accuracy. Here, we investigate a novel graph based relational clustering algorithm that uses both attribute and relational information for determining the underlying domain entities, and we an initial code layout. The algorithm has many application, such as web search, shopping purchase records, government and public health data.

Method, Dataset & System Design

Fig.1(a) shows our initial graph and Fig.1(b) would be our resolution graph output. The node with names are entities. The r's are references and the h's are hyper edges meaning that those references appear in the same paper.

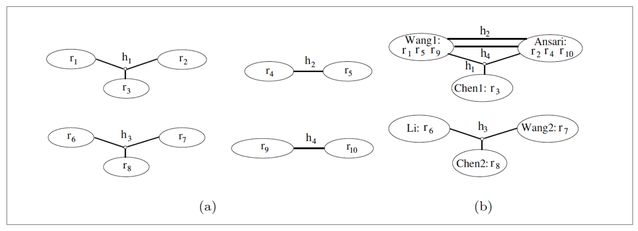

Figure 1. (a) The reference graph (b) The entity graph

We used two datasets to answer our questions. CiteSeer dataset contains 1504 documents with 2892 author references to 165 author entities. The arXiv dataset is 29555 papers in high energy physics publications with 58515 references to 9200 authors.

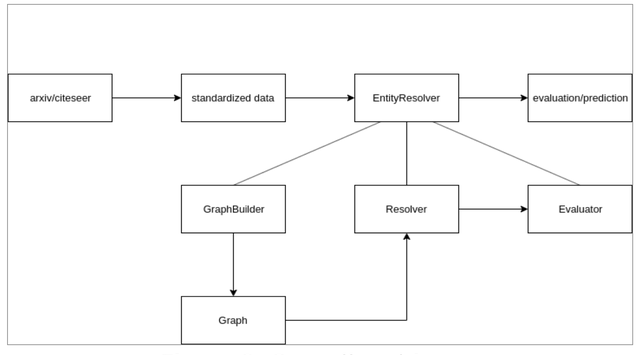

Figure 2. Overall architecture

Entity Resolver: We implemented three top parts of the algorithm, including building graph,calculating similarities and implementing clustering algorithm.

Graph Builder: Built Graph class to implement graph

Resolver: Calculated similarity metrics for clustering

Evaluator: Evaluated clustering using ground truth

Algorithm & Evaluation

Shown in Fig.3 is the pseudocode of the algorithm. For blocking, we implemented Blocking techniques invented by Hernandez and Stolfo,1995. We used 1-exact-match to choose bootstrap candidates. For similarity measure, we implemented Ada and Jaccard.

```
1.   Find similar references using blocking
2.   Initialize clusters using bootstrapping

3.   For clusters c_i, c_j such that similar(c_i, c_j)
4.       Insert ⟨sim(c_i, c_j), c_j, c_j⟩ into priority queue

5.   While priority queue not empty
6.       Extract ⟨sim(c_i, c_j), c_i, c_j⟩ from queue
7.       If sim(c_i, c_j) less than threshold, then stop
8.       Merge c_i and c_j to new cluster c_ij
9.       Remove entries for c_i and c_j from queue
10.      For each cluster c_k such that similar(c_ij, c_k)
11.          Insert ⟨sim(c_ij, c_k), c_ij, c_k⟩ into queue
12.      For each cluster c_n neighbor of c_ij
13.          For c_k such that similar(c_k, c_n)
14.              Update sim(c_k, c_n) in queue
```

Figure 3. High-level description of the relational clustering algorithm

We evaluated the precision, recall and f1. The Fig4 (a) & (b) shows the precision-recall curves using the Adar similarity and 0.5 alpha. Our model has a good performance using CiteSeer data. Our bootstrap strategy successfully initialized the clustering. When the merging of clusters, the recall value increases until reaching the similarity threshold. Our model has a high f1 score, 0.991. Similarly, the model reaches a good result on arXiv data with 0.975 f1 score.

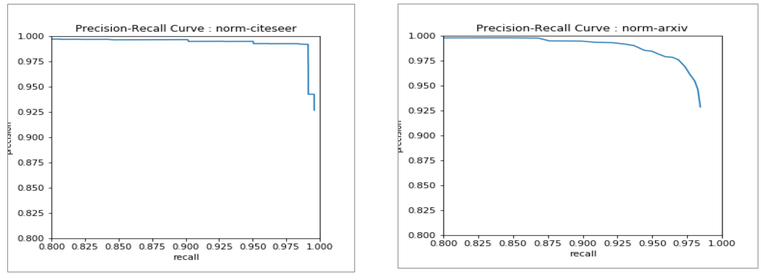

Figure 4. (a) P-R curve for Citeseer dataset (b) P-R curve for arxiv dataset

Conclusion

In order to deploy our model to solve real-life problems, we implemented the code in Python, optimized the data structure and algorithm for efficiency and wrapped into a package called Collective Entity Resolution Our package can be applied to several specific situations. First, it can be used in different attribute types. Second, the users can customize their own bootstrap and blocking strategies. Third, the users can choose different similarity function and evaluation approaches.

Acknowledgments

We express our gratitude to the academic and industry mentors (names showed above).

References

Collective Entity Resolution in Relational Data. ACM Transactions on Knowledge Discovery from Data. Volume 1 Issue 1, March 2007 Article No.5.