Searching and ranking in RDF documents and social networks

Sibel Adalı
Rensselaer Polytechnic Inst.
Troy, NY
sibel@cs.rpi.edu

Alvaro Graves
Rensselaer Polytechnic Inst.
Troy, NY
agraves@cs.rpi.edu

Konstantin Mertsalov
Rensselaer Polytechnic Inst.
Troy, NY
mertsk2@cs.rpi.edu

1. MOTIVATION

As semantic web based applications are gaining popularity, very large RDF documents are becoming common. SPARQL is the *de-facto* standard in querying RDF data and research on efficient implementations of SPARQL interfaces for very large RDF graphs has attracted a great deal of interest in the recent years. However, in large datasets, the user faces the problem that the result set for her queries can be large. In this situation there is no clear for the user, from where to start looking at the results, since all of them are equally valid. Moreover, given the result of a SPARQL query, the only possible order is lexicographical which doesn't help the user to distinguish which of the returned values should she look first. In this sense, it would be desirable to have a notion of "relevance" of nodes. A related problem is that of analyzing social network data. Most social network analysis concentrates heavily on finding social groups and finding the importance of individuals in a social network. However, this work generally considers the social network as a graph with a single type of connection, edges representing the existence of social communication or friendship for example. There are not many methods developed for social networks with many different types of semantic connections. As a result, there is very little work on querying of semantically rich social network data.

In this paper, we consider the problem of clustering and ranking of RDF nodes. Our strategy consists of using graph metrics to calculate the relevance of different nodes in a specific RDF graph. In particular, for ranking we propose an iterative ranking algorithm based on closeness centrality where distances are related to the importance of different paths and we base the clustering on the similarity of different nodes based on their predicates in the RDF graph. Our methods equally apply to rich social network data that can be represented as RDF where certain nodes are considered of type "people". We conduct extensive experiments and validation for our ranking algorithm using the DBLP data set for publications and the DBPEDIA version of wikipedia.

Semsearch 2009

2. RANKING

In our work, we assume that the importance of a node is based on the structure of the graph. In particular, we use *closeness centrality* as the main indicator for our ranking. this metric can be described as the mean geodesic distance between a node u and all the others. Thus, the most central nodes will be the ones which smaller mean shortest distance to the rest of the nodes. In order to obtain meaningful results for all the nodes, we consider the graph undirected. We have found that this metric can give very meaningful results for ranking different kinds of information[1]. However, in this case we are not considering all the nodes the same: Our purpose is to rank specific types of nodes. Moreover, our algorithm makes use of the ranking of certain type of nodes to improve the ranking of other types. This leads to the following iterative algorithm.

2.1 Iterative ranking of nodes

In an RDF graph, connections between certain types of nodes will be through other concepts and relationships. The task becomes determining the importance of the paths using specific types of nodes and via different types of predicates. To accomplish both tasks, we introduce an iterative algorithm. In the first iteration of the algorithm, we create a new graph using only the nodes of a certain type. These nodes are now connected based on the paths found on the original graph. We assume that a domain expert can help identify the important path types for a data set or they can be determined from user queries by finding frequent paths in user queries. We note that the more connected two nodes are in the graph through paths, the closer the relationship between them and thus we assign weights to the edges between two nodes proportional to the number of paths between them. Using this graph, we compute the centrality of the nodes for this specific type (say c_1). Next, we find the importance of other nodes in the original graph based on the average centrality of the nodes of type c_1 that links to them. Given now these new values, we recreate the graph for nodes of type c_1. The value of a path is now dependent on the value of the nodes it passes through. We recompute centrality based on this new graph and continue to iterate in this fashion. In our method, nodes of type c_1 are important if they are central and are connected through important relationships and other nodes are important if they link to central nodes of type c_1. While similar to approaches based on pagerank [5, 6] or Kleinberg's method [3], our method introduces a new iterative approach and uses centrality as the main criteria. Unlike other approaches [4, 6], we use no adjustable values in our algorithm. All weights are determined from the graph structure.

3. CLUSTERING

There are situations when we favor local ranking over the global ranking of nodes in the whole graph. Taking the example of DBLP, we may be interested in people related to a specific field: in that case it is better to consider only the cluster corresponding to that field for ranking. In particular, we cluster the conferences into overlapping groups corresponding to different areas of the science. The clusters are determined based on the strength of the relationships between the conference nodes. For this purpose, we generate the relationship graph $G = (V, E)$ from a given RDF dataset in much the same way as for ranking - G contains one vertex $v_i \in V$ for every conference node in the dataset and the edges are placed to represent the relationship between corresponding nodes (for example, edge and weights may represent the number of authors that published in both conferences). After that we use local optimization strategy to find densely interconnected sets of vertices with respect to some density $D(S)$. The density is defined over a subset of vertices in the graph. We used ratio of sum of weights of edges between nodes in the subset to the total weight of all edges attached to at least one vertex in the dataset. Optimization algorithm starts with initial seed set S and examines every set $S_{updated}$ which can be obtained from S by removing any $v_k \in S$ or by adding any $v_m \notin S, v_m \in V$ to find $S_{updated}$ which maximizes $D(S_{updated}) - D(S)$. The algorithm continues for a predefined number of steps, we found that 1000 - 5000 steps are sufficient to obtain dense clusters.

3.1 Applying clusters to ranking

We can cluster the RDF database with respect to some predicate that can organize the nodes into meaningful groups or societies. For example, consider the DBLP dataset containing researchers, the papers they publish and the venues for these papers. It is possible to cluster this database set with respect to all authors or all conferences. All authors would give groupings of collaborations. This clustering is useful for result summarization. Another way to cluster the DBLP dataset is by grouping together the conferences that are related to each other due to commonality of the authors that publish in them. These conferences correspond to specific areas of research. These clusters can be used in multiple ways. For example, we can find the ranking of nodes in each cluster separately. An author may not be universally famous or well-connected to the whole research community, but she/he may be well-known in a specific research area.

4. EXPERIMENTAL RESULTS

In order to validate our results we compared our results with the h-index[2], a well-known metric of scientific impact. We have seen that our iterative method improves the ranking, compared to the h-index. Here we show some results obtained after a few iterations. Using SwetoDBLP we present the results for one of the clusters that include conference and journals in the area of artificial intelligence. We use an algorithm based on the frequency of the words in the title (similar to a tf/idf ranking) and choose the top keywords. Based on this algorithm, the label attached to this cluster was: "learning, based, using, neural, knowledge, model, genetic, approach, algorithm, agent, robot." We find all authors who published at least one paper in a venue in this cluster and then use our iterative algorithm. Note that this cluster does not necessarily include all authors in the AI field. The ranking of the venues in this cluster is shown in 1. In table 2 we show the ranking of the top authors in this cluster and their associated h-index. Note that we see a correlation between h-index and our ranking and see that the iterative methods improves the results in terms of the h-index.

Table 1: Top 10 AI venues based on author rankings.

Position	Venue	Position	Venue
1	ICRA	6	ECAI
2	IJCAI	7	AAMAS
3	NIPS	8	ICALT
4	WebNet	9	ICML
5	GECCO	10	UAI

Table 2: Top 5 AI researchers using clustering and their h-index.

Pos	Author	h-index
1	Manuela M. Veloso	46
2	Peter Stone	37
3	Hiroaki Kitano	39
4	Minoru Asada	30
5	Satinder P. Singh	23

5. CONCLUSIONS AND FUTURE WORK

We have shown several methods for ranking nodes in RDF and social networks, based on topological properties of them. In particular we have shown an iterative method based on closeness centrality that improves such ranking based on the relation of different types of nodes in the graph. Our results show that by simply considering the graph theoretic information, we are able to extract very significant information from graphs.

6. REFERENCES

[1] A. Graves, S. Adalı and J. Hendler, "A method to rank nodes in an RDF graph (poster)", in *Proc. of the Intl Confe on Semantic Web*, 2008

[2] J.E. Hirsch, "An index to quantify an individual's scientific research output", in *Proc. of the National Academy of Sciences*, pp. 16569–16572, 46(102), 2005.

[3] R. Kleinberg, "Authoritative Sources in a Hyperlinked Environment", Journal of the ACM, 46(5), pp. 604-632, 1999.

[4] B. Bamba and S. Mukherjea. "Utilizing resource importance for ranking semantic web query results". In *Proc. of 2nd Intl Workshop on Semantic Web and Databases*, 2004, pp. 185–198, 2004.

[5] A. Damian, W. Nejdl, and R. Paiu. "Peer-sensitive objectrank - valuing contextual information in social networks." In *Proc. Intl Conf on Web Information Systems Engineering*, pages 512–519, 2005.

[6] H. Hwang, V. Hristidis, and Y. Papakonstantinou. "Objectrank: a system for authority-based search on databases". In *Proc. of the ACM SIGMOD Conf*, pages 796–798, 2006.