Proceedings of the Eleventh International AAAI Conference on Web and Social Media (ICWSM 2017)

Querying Documents Annotated by Interconnected Entities

Shouq Sadah, Moloud Shahbazi, Vagelis Hristidis
Department of Computer Science and Engineering
University of California, Riverside, California, USA
{ssada003, mshah008, vagelis}@cs.ucr.edu

Abstract

In a large number of applications, from biomedical literature to social networks, there are collections of text documents that are annotated by interconnected entities, which are related to each other through *association graphs*. For example, social posts are related through the friendship graph of their authors, and PubMed articles are annotated by Mesh terms, which are related through ontological relationships. To effectively query such collections, in addition to the text content relevance of a document, the semantic distance between the entities of a document and the query must be taken into account.

In this paper, we propose a novel query framework, which we refer as *keyword querying on graph-annotated documents*, and query techniques to answer such queries. Our methods automatically balance the impact of the graph entities and the text content in the ranking. Our qualitative evaluation on real dataset shows that our methods improve the ranking quality compared to baseline ranking systems.

1 Introduction

Much research has studied how to query interconnected documents, such as Web pages, relational databases (tuples are the documents) or XML data (XML elements are the documents). In these settings, the assumption is generally that the user submits a keyword query and the system combines the text similarity with the graph structure to rank documents or collections of documents.

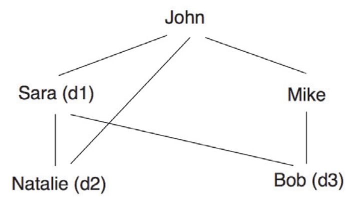
However, this paradigm misses the quite common scenario where the relationships do not exist directly between the documents, but between *graph entities* contained in the documents. As an example, consider the posts of a social network, which contain the id of their author (and possibly of the recipients too), and the users are connected through a friendship graph. In this case, the graph entity is the user who submits the query.

We specifically study the problem where the query, in addition to keywords, specifies one or more graph entities (or simply entities) of interest. In the social networks domain, the user who submits the query becomes the entity, meaning that he is more interested in documents posted by closer friends. For example, if user John in Figure 1 submits a query with keyword "birthday", he likely prefers a post related to "birthday" by his direct friend Mike than by his farther friend Bob (assuming other factors such as freshness, user influence and text similarity are the same).

We refer to the above type of queries as *keyword queries on graph-annotated documents*, to differentiate them from the more traditional keyword queries on interconnected data mentioned above. In addition to the social networks, where we focus our presentation for clarity, keyword queries on graph-annotated documents can be found in other domains. In medical literature bibliographies, each article is annotated by a set of medical concepts, connected by a medical ontology. In e-commerce, products have a description and are annotated by a product category (e.g., "SLR Camera"); the category becomes the entity. In spatial databases, each document may have a location and a graph of (e.g., roads) that defines the distances between them; the user's location becomes the entity.

A key challenge in effectively answering keyword queries on graph-annotated documents is to balance the importance of the various graph entities and the keyword terms. For example, in the above social network query by John, if there is a post by Bob that is very relevant to keyword "Obama", it is likely a better result than a post of Mike that is less relevant to Obama, if Obama is a global (not local) topic in the social network. In contrast, if John specifies keyword "birthday", he is likely interested in birthday posts of his friends only.

Figure 1: Example of social network graph showing the post IDs for users



In this work, we make the following contributions:

- We define the *keyword queries on graph-annotated documents* problem, and propose an effective framework that

Table 1: Association graphs

Association Graph	Entity	Association Edge
ontology	concept	semantic relationship
social network	user	friendship
spatial objects	object location	distance

intelligently balances graph entities relevance with keyword relevance.

- We evaluate the results quality of our proposed framework, comparing to several baseline methods through a comprehensive user study

2 Related Work

Ontology-based query expansion XontoRank (Farfan et al. 2009) exploits ontological relationships to answer keyword queries on XML documents. Precomputing all pairs of distances is infeasible in our setting as we don't have a distance threshold. Further, their query only contains keywords and not graph entities, which are necessary in our setting (e.g. to specify the user who submits a query in a social network). (Xiong and Callan 2015) use semi-structured data sources such as controlled vocabularies and knowledge bases to improve the quality of ranking – entities from external sources are used as objects connecting query and documents. Their proposed technique "EsdRank" annotates the query using related objects from external data to improve retrieved documents.

Search in social networks (Maniu and Cautis 2012) proposed a *top-k* algorithm that relies exclusively on weight of tagging systems. Given a query consisting of a user and keywords (tags in this case), the algorithm shall return *top-k* relevant documents having the highest score with respect to the keywords and the proximity (or social) scores between the users.

3 Problem Definition and Semantics

Let *D* be a collection of documents. Each document $d \in D$ is defined as a tuple *(d.w, d.u)*, where *d.w* is its textual content and *d.u* is set of graph entities.

The *entities* are related to each other through an association graph. An association graph $G = (N, E)$ consists of a set of entities (nodes) *N*, and a set of association edges *E*. The nodes and edges in various association graphs are shown in Table 1. A social network graph could be an undirected (e.g. friendship connections in Facebook) or directed graph (e.g. follower/followee connections in Twitter), where *N* is a set of users and *E* is a set of relationships between users. In the rest of the paper, for simplicity of presentation, we assume that there is only one association graph; extending the algorithms for multiple association graphs is straightforward.

Example 1: Consider social posts, which are annotated by the id of their author. In contrast to other graphs, each document here can be annotated with only one entity. Suppose we have the following posts and the social network graph shown in Figure 1.

d1: Sara: "Obama to announce $600 million in grant programs to prepare workforce for jobs"

d2: Natalie: "Michael Bloomberg Pledges Million to Push Gun Control"

d3: Bob: "Obama Supporters Don't Know Obama"

A *keyword query on graph-annotated documents* $Q = (Q.w, Q.u)$ consists of $Q.w$, a set of query keywords, and $Q.u = \{Q.u_1, \cdots, Q.u_m\}$, a set of graph entities.

A *Top-k* keyword query on graph-annotated documents returns a ranked list of the *k* most relevant documents from *D* based on a similarity function that combines both the graph entities and the textual similarity.

Example 1 (cont'd): if user "John" submits keyword query "Obama policies", the corresponding keyword query with graph entity is $Q = (\{\text{“}Obama\text{”}, \text{“}policies\text{”}\}, \{John\})$.

Our contribution in this paper is to balance the impact of the query keywords and the query entities in the ranking (Section 4).

4 Ranking Semantics

To keep the ranking model generic in terms of combining functions, we define separately the impact of the query keywords $IRScore(d.w, Q.w)$ and graph entities $Dist(d.u, Q.u)$ and combine them by a monotone aggregate function. The monotonicity for the graph entities is defined on the path length between two entities, whereas for the keywords impact is defined on the term frequencies or other text features.

The score of the document d for query Q is:

$$score(d, Q) = f(Dist(d.u, Q.u), IRScore(d.w, Q.w)) \quad (1)$$

The combining function f may include other features such as document or user popularity. We adopt a previously proposed combining function that multiplies the impact of the two components and uses a decay factor for the entities distance (Guo et al. 2003) (originally used in the context of XML documents):

$$score(d, Q) = \alpha^{Dist(d.u,Q.u)} \times IRScore(d.w, Q.w) \quad (2)$$

where $\alpha < 1.0$ is the distance decay factor in the association graph G. A key challenge, which we tackle in Section 4.1, is the computation of α.

For the purpose of the experiments, to compute *Dist(d.u, Q.u)*, we build upon previous work (Farfan et al. 2009; Rada et al. 1989), and define it as the sum of the shortest path distances between each of the query's graph entities in *Q.u* and their closest document entity in *d.u*. In other words, *Dist(d.u, Q.u)* is the sum of the number of edges between every entity in *Q.u* and its closest entity in *d.u*. Formally, *Dist(d.u, Q.u)* is defined as follows:

$$Dist(d.u, Q.u) = \sum_{q \in Q.u} G.ShortestPath(q, d.u) \quad (3)$$

where $ShortestPath$ computes the length of the shortest path in association graph between an entity q and its closest entity in set $d.u$. In the case of multiple association graphs, the score is defined as following:

$$score(d, Q) = \prod_{for\ each\ G} \alpha_G^{G.Dist(d.u,Q.u)} \times IRScore(d.w, Q.w) \quad (4)$$

where α_G is the decay factor for association graph *G*

An example of a specific text ranking function *IRScore(d.w, Q.w)* used in our experiments is as follows:

$$IRScore(d.w, Q.w) = \sum_{t \in Q.w} tf(d.w, t) \times idf(t) \quad (5)$$

4.1 Computation of α Parameter

In this subsection, we explain how α is computed to balance the relevance of the graph entity distance with the keyword similarity. We argue that the following intuition holds, which we also evaluate in Section 5: *If the documents that match the query keywords have similar content regardless of their distance to the query's graph entities, then the distance should have a smaller importance.*

Specifically, if the association graph is a social network, this means if user John specifies keyword "Obama" and his friends do not have any consistent political views (e.g., some are Republican, some Democrat, and some undecided), then John would likely be interested in posts about Obama coming from both close friends and the rest of the network. Thus, the importance of $Q.u$ is higher. In contrast, if John's friends discuss a topic about Obama (e.g., his immigration views), which is distinct from the general chatter about Obama on the whole network, then John would prefer posts from his friends rather than from the rest of the network; thus, the importance of $Q.w$ is higher. As another example, for query "birthday party", if John's friend had a party, then John would be most interested in posts about that party and not about a random party on the network.

To achieve the above intuition in the social network application, we compute the content difference between the local community and the whole network for the set of documents that match the query keywords $Q.w$. A popular measure of the difference of two sets of documents is the Kullback-Leibler (*KL*) divergence (Kullback and Leibler 1951), which measures the difference between two probability distributions, specifically the distribution of terms in the posts relevant to $Q.w$ from the user's social neighborhood, and the distribution of terms in the posts relevant to $Q.w$ in the whole network:

$$KL(R^u_Q, R_Q) = \sum_{v \in vocab} R^u_Q(v) \log \frac{R^u_Q(v)}{R_Q(v)} \quad (6)$$

where R^u_Q and R_Q are the probability distributions of the relevant posts in the neighborhood of the user u and in the whole social network (hence the latter may be precomputed as a set of ¡term, probability¿ pairs), respectively. Let D be the set of all posts in the social network, and let D_Q be the subset of D that contains at least one of the keywords in $Q.w$, and D^u_Q be the subset of D_Q posted by users with distance up to T from the query user u. Suppose we have n query terms in $Q.w$, we compute the exact value of R_Q as:

$$R_Q(v) = \frac{\sum_{d \in D_Q} tf(d, v)}{\sum_{d \in D_Q, v' \in vocab} tf(d, v')} \quad (7)$$

To compute R^u_Q, we concatenate the text of all posts that are relevant to $Q.w$ (e.g., that contain all terms in $Q.w$) posted by users with distance up to T from the query user u. We set the threshold $T = 1$ to only consider direct friends.

$$R^u_Q(v) = \frac{\sum_{d \in D^u_Q} tf(d, v)}{\sum_{d \in D^u_Q, v' \in vocab} tf(d, v')} \quad (8)$$

As an example, the word "Peter" may appear with probability $R^u_Q(Peter) = 0.001$ in the user's neighborhood and with probability $R_Q(Peter) = 0.00005$ in the whole social network. To incorporate the *KL* measure in our scoring function (Equation 2), we need first to normalize it for each query Q between $(0, 1)$ since *KL* is unbounded. Therefore, we define α using *KL* as follow:

$$\alpha = e^{-KL(R^u_Q, R_Q)} \quad (9)$$

That is, the larger *KL* means the two sets of documents are different, and hence the posts from user's neighbourhood are more preferable. The same rationale applies to several other types of association graphs, such as the ontology graph discussed above.

5 Qualitative Experiments

In this section, we present the experimental evaluation of our KL-based method. First we describe our dataset, and then we present the qualitative experiment.

Datasets: We conducted our experiments on Twitter dataset, which was obtained using the Twitter Streaming API. Since the relations between users are bidirectional, i.e. each user has followers and followings, we discarded all unidirectional relations to convert the graph to undirected one, that is, we only keep an edge between two users if they follow each other. The goal of only considering bidirectional relations is to define the local and global communities for each user.

Table 2: Description of Twitter dataset

Dataset	Number of Users	Number of Tweets
Twitter	18,492	221,643

Table 2 shows a description of Twitter dataset, including the number of the users and tweets.

We evaluate our proposed ranking method, which computes the α parameter using the KL strategy based on user's community, as we discussed in Section 4.1. To achieve this, we used 20 queries, where each query consists of a user who submits the query, and a list of keywords. For that, we selected 20 different users from Twitter dataset, where each user has at least 20 friends and 10 tweets. For each query Q, we combined the query keyword $Q.w$ with the user id as the graph entity $Q.u$, and then we computed the *top-3* results by using 6 different methods:

1. Two baselines:
   - IRscore baseline: computes documents' scores using the text similarity only, and ignore the distance to the $Q.u$

Table 3: Query keywords and number of matches per ranking method

	IRscore Baseline	Distance Baseline	α=0.1	α=0.5	α=0.9	KL-based
Obama	1	0	0	3	1	1
Forest	2	1	3	3	2	3
Tax	0	1	2	3	2	3
Wednesday	0	0	3	3	0	3
Madrid	1	0	1	3	2	3
Prince	0	2	3	3	1	3
Bundy	1	0	2	3	1	2
Michael	0	2	3	3	0	3
Sonia	3	0	0	0	3	3
Gun	3	0	2	2	3	3
Happy	1	0	3	3	3	3
Mystery	2	1	3	3	2	3
Food	0	3	3	1	0	3
Hope	0	0	3	3	1	1
Constitution	3	0	1	1	3	3
Beautiful	0	1	3	1	2	2
Photo	0	1	3	3	0	3
Law	2	0	1	2	2	2
Market	2	1	3	2	2	3
Dream	1	0	1	1	2	3
Total	22	13	43	46	32	53

- Distance baseline: orders the documents by their distance from the user; for ties orders by decreasing document freshness

2. Static α parameter, using Equation 2. We consider $\alpha =$ 0.1, 0.5, and 0.9
3. Adaptive (query-specific) α using *KL* divergence, as shown in Equation 6

After finding the *top-3* results for each of the methods, we took the union of the results and conducted a user study. We asked seven students to imagine that they are the selected Twitter users *Q.u*, and to mark the *top-3* relevant tweets for each of the 20 queries (the distance of each tweet to the user is also displayed). To help them get an idea of what their friends and the rest of the network talk about, we provided them with the following information:

1. Top 20 tweets of the local community (immediate network) that contain the query keyword
2. Top 20 tweets of the global community (all users in the network) that contain the query keyword

After the students selected the *top-3* results for each query, we took the majority voting to define the *top-3* "ground truth" results for each query, and then we compared all the methods with the students' selections in terms of accuracy, that is, how many of their top-3 results are in the ground truth top-3.

Table 3 shows the accuracy of the 20 queries. Using *KL*-based method to compute α parameter achieved accuracy of 88.33% comparing to the students' selections, which is 51.66% improvement over IRscore baseline and 66.66% improvement over Distance baseline. Our method using $\alpha = 0.5$ also achieved accuracy of 76.67%, with improvement of 43% over IRscore baseline and improvement of 55% over Distance baseline.

To intuitively explain the role of *KL* divergence in computing the α parameter, consider the query keyword "Michael", where *KL* divergence is high between the local and the global communities, specifically *KL*=1.11. The reason for the high *KL* value is that the local community for the user who submitted the query keyword "Michael" talks more about Michael Bloomberg, while the global community talks more about Michael Tsarion and Michael Donnor. Since the two communities are different, tweets from local community are preferred, and hence the α here equals 0.33 by using Equation 9. Another example is query keyword "Constitution", where *KL* divergence here equals 0.05. Thus, both local and global communities talk about the same constitution, which means the user is more likely interested in tweets from both local and global communities when selecting top relevant tweets. To avoid computing the exact *KL*, we only consider the top recent tweets in both communities (1000 tweets for local and 5000 tweets for global).

6 Conclusions

In this work, we proposed a novel query framework for querying collections of graph-annotated documents, which we refer as *keyword querying on graph-annotated documents*. Our method automatically balances the importance of the graph entities relevance and the text content relevance. Our qualitative experiments show that the KL-based method achieved an average accuracy improvement of 60% over baselines.

Acknowledgment

Partially supported by NSF grants IIS-1216007, IIS-1447826 and IIS-1619463.

References

Farfan, F.; Hristidis, V.; Ranganathan, A.; and Weiner, M. 2009. Xontorank: Ontology-aware search of electronic medical records. In *Data Engineering, 2009. ICDE'09. IEEE 25th International Conference on*, 820–831. IEEE.

Guo, L.; Shao, F.; Botev, C.; and Shanmugasundaram, J. 2003. Xrank: Ranked keyword search over xml documents. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, 16–27. ACM.

Kullback, S., and Leibler, R. A. 1951. On information and sufficiency. *The annals of mathematical statistics* 22(1):79–86.

Maniu, S., and Cautis, B. 2012. Taagle: Efficient, personalized search in collaborative tagging networks. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, 661–664. ACM.

Rada, R.; Mili, H.; Bicknell, E.; and Blettner, M. 1989. Development and application of a metric on semantic nets. *Systems, Man and Cybernetics, IEEE Transactions on* 19(1):17–30.

Xiong, C., and Callan, J. 2015. Esdrank: Connecting query and documents through external semi-structured data. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 951–960. ACM.