Exploiting Paths for Entity Search in RDF Graphs

Minsuk Kahng and Sang-goo Lee
Department of Computer Science and Engineering
Seoul National University
Seoul, South Korea
{minsuk, sglee}@europa.snu.ac.kr

ABSTRACT

The field of entity search using Semantic Web (RDF) data has gained more interest recently. In this paper, we propose a probabilistic entity retrieval model for RDF graphs using paths in the graph. Unlike previous work which assumes that all descriptions of an entity are directly linked to the entity node, we assume that an entity can be described with any node that can be reached from the entity node by following paths in the RDF graph. Our retrieval model simulates the generation process of query terms from an entity node by traversing the graph. We evaluate our approach using a standard evaluation framework for entity search.

Categories and Subject Descriptors: H.3.3 [Information Search and Retrieval]: Retrieval models

General Terms: Algorithms, Experimentation

Keywords: Entity Search, RDF Graphs, Object Retrieval, Semantic Search, Structured Data, Semantic Web

1. INTRODUCTION

As the amount of Semantic Web (RDF) data published in the form of Linked Data[1] has been increasing recently, many researchers have been exploiting the RDF data for retrieval tasks [1]. We address the problem of retrieving entities relevant to a keyword query by using only RDF data [8]. This area has gained popularity recently, leading to the consideration of several retrieval models [2, 6, 4, 3].

Unfortunately, existing retrieval models do not fully reflect the characteristics of RDF data. RDF triples can be represented by a graph[2], thus the nodes in the RDF graph are somewhat related to each other even if they are not directly linked through a single triple (distance=1). For example, given a movie entity node `<../movie/35>`, we can find the node with the name of its director by following two triples (`<../movie/35>, <../director>, <../person/928>`) and (`<../person/928>, <../name>, "James Cameron"`) as depicted in Table 1 and Figure 1. However, most existing models do not capture this kind of case because they assume that the descriptions of an entity exist only at the nodes that are directly linked to the entity node. Therefore, the entity `<../movie/35>` cannot be described by the name of its director. So, if a user issues a query 'movie james cameron' to find movies that are directed by James Cameron, they fail to retrieve relevant movies. This issue can be partially tackled by tokenizing URIs of resources itself [3, 2, 4]. However, it becomes problematic when URIs are made up of meaningless strings like `<../928>`, rather than `<../James_Cameron>`.

Table 1: Examples of RDF triples

Subject	Predicate	Object
`<..//a.b/movie/35>`	`<../title>`	`"Avatar"`
`<..//a.b/movie/35>`	`<../director>`	`<..//a.b/person/928>`
`<..//a.b/person/928>`	`<../name>`	`"James Cameron"`
`<..//a.b/movie/19>`	`<../director>`	`<..//a.b/person/928>`

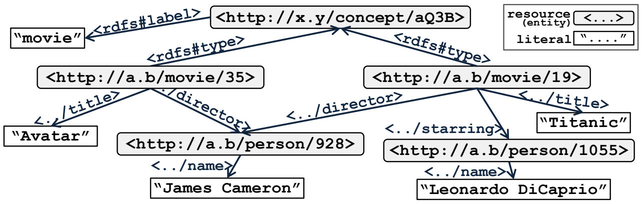

Figure 1: Graph representation of RDF data

In this paper, we propose a probabilistic entity retrieval model that can capture indirect relationships between nodes in the RDF graph. We design the model based on the assumption that the descriptions of an entity exist at any literal node that can be reached from the resource (entity) node by following the paths in the graph.

2. PROBLEM DEFINITION

The RDF data consists of a set of *triples*, where each *triple* has three components: `(subject, predicate, object)`. Both the *subject* and *predicate* are *resources*, and the *object* is either a *resource* or a *literal*. A *resource* represents any entity or concept (e.g. a movie), and is identified by URI (e.g. `<../movie/35>`). A *literal* contains textual information (e.g. `"Avatar"`). Our problem is to rank resources E according to their probability of being relevant given a keyword query Q, denoted by $P(E|Q)$.

3. RETRIEVAL MODEL

We propose a novel RDF resource retrieval model *PathLM* based on the language modeling. It simulates the generation of query Q by following paths from a resource node E to several related literal nodes L_j. Using Bayes' Rule, we have:

$$P(E|Q) \overset{rank}{=} P(Q|E)P(E), \tag{1}$$

[1] http://www.w3.org/standards/semanticweb/data

[2] A triple can be thought of as a directed edge from the subject node to object node in the RDF graph.

SIGIR'12, August 12–16, 2012, Portland, Oregon, USA.
ACM 978-1-4503-1472-5/12/08.

where $P(Q|E)$ is query likelihood, $P(E)$ is resource prior. Assuming independence between query terms, we have:

$$P(Q|E) = \prod_{i=1}^{|Q|} P(q_i|E). \qquad (2)$$

A resource can generate terms by passing through literal nodes since only literal nodes contain textual information used to match the query terms. We have:

$$P(q_i|E) = \sum_{j=1}^{m} P(q_i|L_j)P(L_j|E), \qquad (3)$$

where L_j is a j-th literal node that can be reached from the resource node E. There are m literal nodes related to E, and the sum of $P(L_j|E)$ is 1 ($\sum_{j=1}^{m} P(L_j|E) = 1$).

Finally, we obtain:

$$P(E|Q) = P(E) \prod_{i=1}^{|Q|} \sum_{j=1}^{m} P(L_j|E)P(q_i|L_j). \qquad (4)$$

The details of three probability terms in Eq. 4 are as follows.

$P(E)$*: Resource Prior.*

$P(E)$ is the prior probability of a resource. The value can be assumed to be uniform or determined using the number of literal nodes related to the resource node.

$P(L|E)$*: Select Paths to Literals given Resource.*

$P(L_j|E)$ is the probability of selecting a path to literal L_j given a resource E. It indicates the importance of the path from the resource node E to the literal node L_j in the RDF graph. The value can be either assigned manually based on the path (e.g. $E \xrightarrow{\texttt{<director>}} \circ \xrightarrow{\texttt{<name>}} L_j$), or learned.

$P(q|L)$*: Query Likelihood given Literal.*

We use the language modeling approach to simulate the generation of query terms given a literal node. $P(q_i|L_j)$ is estimated by dividing the term frequency of q_i in the literal L_j by the length of L_j. It is smoothed using a Dirichlet prior μ with the collection language model. We have:

$$P(q_i|L_j) = \frac{tf(q_i, L_j) + \mu \frac{c_{q_i}}{|C|}}{|L_j| + \mu}. \qquad (5)$$

4. EVALUATION

We evaluate our approach using the evaluation framework used in the *Semantic Search Challenge 2010*[3]. The dataset is the *Billion Triple Challenge 2009* collection. After deduplication, there are about 886 million triples, 175 million resources, and 296 million literals. We index literals using *Indri* with the Krovetz stemmer and no stopwords. The query set has 92 queries extracted from the query logs of search engines. The relevance judgments are made with three-point scale. We add judgments if they are not available. For each query, the model returns top-100 resources, and is evaluated with MAP, P@10, and NDCG. We conduct a series of two-tailed paired t-tests of 99% confidence.

We compare our *path*-based approach to three baseline approaches: 1) *Plain text*, 2) *Attribute with uniform weights*, and 3) *Attribute with different weights*. First, for the *Plain text* approach, a pseudo-document is built for each resource by concatenating literals that are directly linked to the resource node [4, 6]. Two standard retrieval models, *BM25* and *Language Model with Dirichlet smoothing (LM)*, are used. The parameters in the models, such as b and μ, are tuned empirically. Next, the *attribute*-based models are designed by defining attributes as literal nodes whose distances from the resource node are 1. We employ *BM25F* [2, 3] and *Mixture of Field LM (MFLM)* [7, 6, 4]. We note that our *PathLM* can be thought of as a type of the *MFLM*. The weights of attributes are assigned uniformly at first, then we assign them differently by defining some important predicates manually [2]. Lastly, for our *path*-based approach, we consider several selected paths from the resource node to literal nodes where distances between them are 1 or 2.

Table 2: Retrieval performance. $\ast$ indicates significant difference between the BM25- and LM-based models within the same approach. $\dagger$, $\ddagger$ indicate significance over the corresponding models of *Plain* and *Attr(uni)*, respectively. $\triangledown$ indicates significantly worse results compared to *PathLM*.

Approach	Ret.Model	MAP	P@10	NDCG
Plain text	BM25	0.2366 $\triangledown$	0.4293 $\triangledown$	0.4288 $\triangledown$
	LM	0.2426 $\triangledown$	0.4272 $\triangledown$	0.4438 $\triangledown$
Attr. (uniform w)	BM25F	0.2014 $\triangledown$	0.4380	0.3886 $\triangledown$
	MFLM	0.2711 $\ast\dagger\triangledown$	0.4783	0.4765 $\ast\dagger\triangledown$
Attr. (different w)	BM25f	0.2523 $\ddagger\triangledown$	0.4826 $\dagger\ddagger$	0.4484 $\ddagger\triangledown$
	MFLM	0.2889 $\ast\dagger\ddagger\triangledown$	0.5076 $\dagger$	0.4913 $\ast\dagger\ddagger\triangledown$
Path	**PathLM**	0.3268 $\dagger\ddagger$	0.5033 $\dagger$	0.5245 $\dagger\ddagger$

Table 2 shows experimental results. The LM-based models generally perform better than the BM25-based ones. We also observe that the *MFLM* outperforms the original *LM*. When more weights are given to some important attributes, both *BM25F* and *MFLM* perform better. The proposed path-based approach outperforms all baselines in terms of MAP and NDCG ($p < 0.001$). Moreover, it outperforms all previous work that used the same collection [6, 2, 3]. A possible explanation is that it can retrieve more relevant resources by exploiting some relations like `sameAs` [3].

As future work, we plan to conduct more detailed evaluation based on the proposed model, such as 1) determining resource prior in various ways, 2) learning which kinds of paths have greater effect on performance, and 3) applying different smoothing methods. After further research, we expect to better understand the properties of our path-based retrieval model which enables us to incorporate indirect relationships in the RDF graph [5].

Acknowledgments: This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MEST) (No. 20110030812).

5. REFERENCES

[1] K. Balog, P. Serdyukov, and A. P. de Vries. Overview of the trec 2010 entity track. In *TREC*, 2010.

[2] R. Blanco, P. Mika, and S. Vigna. Effective and efficient entity search in rdf data. In *ISWC*, 2011.

[3] S. Campinas, R. Delbru, N. Rakhmawati, D. Ceccarelli, and G. Tummarello. Sindice bm25mf at semsearch 2011. In *SemSearch*, 2011.

[4] J. Dalton and S. Huston. Semantic entity retrieval using web queries over structured rdf data. In *SemSearch*, 2010.

[5] M. Kahng, S. Lee, and S.-g. Lee. Ranking objects by following paths in entity-relationship graphs. In *PIKM (at CIKM)*, 2011.

[6] R. Neumayer, K. Balog, and K. Nørvåg. On the modeling of entities for ad-hoc entity search in the web of data. In *ECIR*, 2012.

[7] P. Ogilvie and J. Callan. Combining document representations for known-item search. In *SIGIR*, 2003.

[8] J. Pound, P. Mika, and H. Zaragoza. Ad-hoc object retrieval in the web of data. In *WWW*, 2010.

[3] `http://km.aifb.kit.edu/ws/semsearch10/`