Combined Structured and Keyword-Based Search in Textually Enriched Entity-Relationship Graphs

Keywords: Information Retrieval, Ranking, Keyword, Entity-Relationship Graph, SPARQL, Activation Spreading

Davide Magatti
MAGATTI@DISCO.UNIMIB.IT
Department of Informatics, Systems and Communications
Universita' degli Studi di Milano-Bicocca,
Viale Sarca 336, 20126 Milan, Italy

Florian Steinke, Markus Bundschus, Volker Tresp
FLORIAN.STEINKE/MARKUS.BUNDSCHUS.EXT/VOLKER.TRESP@SIEMENS.COM
Siemens AG, Corporate Research, Otto-Hahn-Ring 6, 81541 Muenchen, Germany

Abstract

We present a novel method to combine simple, flexible, keyword-based search with expressive structured queries. We assume that an entity-relationship graph is given where some of the nodes are linked to unstructured text documents. The aim of our approach is to efficiently search for relevant entities or facts about entities. Using several examples, we demonstrate the new types of querying that can be realized by our approach.

1. Introduction

The interest in semantic web techniques has steadily been growing in recent years. By comparing the standard web with the semantic web one might realize that there are two common but largely distinct ways to represent, store and retrieve information. On the one hand, there are large collections of unstructured text documents. Most web documents are of this form, and also many popular services, such as Wikipedia, are fundamentally document or text based. Information retrieval in this unstructured domain is commonly done with keyword-based search and the results of a query are typically ranked lists of documents.

Keyword-based search is very powerful, since it is flexible, can be implemented efficiently, and is highly intuitive for most users. Yet, keyword-based search also has its well-known problems. High recall is hampered by the fact that there might be many expressions with the same semantic meaning (polysemy). Specificity is negatively influenced by context-dependent semantics of many words (polymorphism). Due to low specificity, a meaningful ranking of the search results is indispensable in keyword-based search.

The extraction of knowledge from the retrieved documents is typically up to the user himself. Thus, when searching directly for specific entities, list of entities or facts, traditional search engines are only of limited use.

On the other hand there are structured data sources, from which information can be extracted with formal queries. As structured information sources we consider semantic networks such as DBpedia (Auer et al., 2007), YAGO (Suchanek et al., 2007) or Linked Life Data (LLD) (Ontotext, 2009). These sources directly encode entity-relationship (ER) graphs, which consist of entities like persons, countries, etc. and of relations or facts concerning these entities, e.g. statements like `Albert Einstein bornIn Germany`. Moreover, we also consider traditional relational databases, which can be mapped into ER graphs via tools like D2R Server (Bizer & Cyganiak, 2006) or Openlink Virtuoso (Erling & Mikhailov, 2009).

For structured information repositories information retrieval is typically performed with structured queries

in languages like SPARQL (W3C, 2007) for semantic web domains or with SQL for traditional relational databases. These languages allow for very precise query formulation, for efficient filtering and aggregation, such that the search results produce a well-defined set. For structured queries, a ranking of the query results is of less importance.

In this paper we address two critical problems with structured information storage and retrieval. First, much knowledge is still and will continue to be in text form. While entity extraction and relation extraction have recently made great progress (Sarawagi, 2008; Suchanek et al., 2009; Bundschus et al., 2008), it seems highly likely that, also in the near future, important information will remain in textual, unstructured form. A second problem is the high complexity of structured queries, see e.g. the example queries in (Ontotext, 2009). So even if all relevant knowledge could be transformed into structured form, many users would still have great difficulties to retrieve this information. For standardized queries an intelligent user interface may automate the formulation of a certain type of query. However, for any non-standard search task the user has to write specific structured search queries which requires deep formal thinking and good knowledge of the structure of the data store.

To motivate our approach we observe that in practice many information repositories are in fact a combination of the two data-storage regimes described above: they both contain structured and unstructured information. For example consider the textual repository Wikipedia where many important facts are available in structured form, e.g. via DBpedia (Auer et al., 2008) or YAGO. However the structured information is only available for the info-boxes in each wikipage and not for the content of the main article and important content remains hidden in the textual description of each Wikipedia node. In our work, we will work on both the structured part of Wikipedia and the textual information. We consider the joint information sources as a textual enriched ER graph. As a second example, consider the computing environment of a large company. There are often huge collections of unstructured textual documents available like emails, project reports, or product handbooks. At the same time, there are typically many well-curated databases available listing and linking entities like employees, hierarchies of departments, customers and products. Available documents can often be linked to one or more entities in the structured representation. In total, one can thus consider the whole information repository again as a textually enriched ER graph.

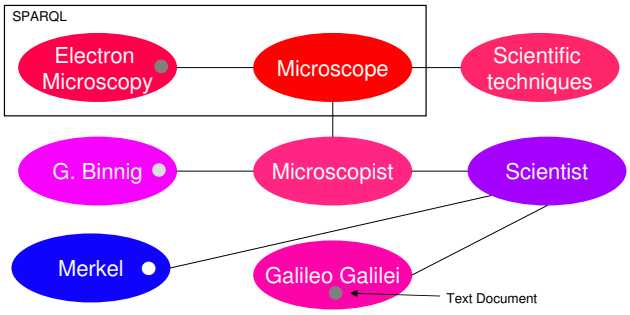

Figure 1. Stylized subgraph of the YAGO ER graph. Some of the nodes are linked to text documents (depicted via circles within the nodes). Given a set of keywords (here "microscope"), the documents obtain some relevance (dark circles mean high relevance). This translates directly into a score for the nodes connected to the document. These scores are generalized to all nodes with and without text with help of the proposed relevance propagation algorithm (the resulting relevances are color-coded, red means high score, blue low score). A SPARQL query finally cuts out pieces of the ER graph. These are ranked according to their aggregated relevance scores.

Given a textually enriched ER graph we show how to formulate hybrid queries consisting of user provided keywords and simple structured queries, which might be encoded in a user interface. Dependent on the query, the results will be ranked lists of entities or lists of facts, that hold between the entities. Technically, we contribute a novel, sound and efficient method to propagate text-based relevance scores on ER graphs to use these for ranking the results of a SPARQL query. While in one setting of our approach, the keywords can be used to rank the results of a classical faceted search, our method is much more flexible and powerful. It realizes approximate keyword matches in the ER graph and also allows for more complex structured queries. The advantages of the proposed approach will be demonstrated in the experimental section.

2. Proposed Hybrid Search Engine

In information retrieval it is often useful to first expand a query in order to alleviate some of the problems with polysemy. To the reduce polymorphism, one afterwards narrows down the search results according to the semantics of the query. We implement this basic idea as follows: First, we use a keyword based query on the text-documents in the ER graph and then propagate the keyword relevance score via a propagation algorithm to the whole ER graph. We then perform a SPARQL query as an effective semantic filter and rank the SPARQL results according to the relevance computed from the keywords. This process is shown

schematically in Figure 1.

2.1. Keyword Query and Relevance Propagation

Given the ER graph with links to textual documents, we index a node with the words in all the texts associated with that node using Lucene[1]. If a node is connected to more than one document, we join the documents and index the result.

At query time, we can then retrieve all the documents that contain the given keywords or match a given regular expression efficiently. Since the documents are linked to nodes in the ER graph, a query implicitly specifies a subgraph of the ER graph.

With $G = (V, E)$ being the ER graph with n vertices $v_i \in V$ and m edges $(i, j) \in E$, the Lucene query formally yields a Lucene relevance score l_i for the text associated with a node and thus for the node v_i itself. For nodes that are not returned by Lucene we define $l_i = 0$.

The keyword-based node relevances l_i are then generalized via exploiting the known, meaningful structure of the ER graph. Following a page rank like principle (Brin & Page, 1998) or similarly activation spreading ideas (Crestani, 1997), we compute novel relevance scores r_i for each node, by iterating

$$r_i = l_i + \lambda \sum_{(j,i)\in E} \frac{r_j}{d_j}.$$

Here, d_j is the out-degree of node j and $0 \leq \lambda \leq 1$ is a weighting factor. Thus, the novel relevance r_i of each node is the Lucene relevance l_i plus contributions of the novel relevances of nodes that are connected via incoming edges.

The weighting factor λ discounts relevance propagation over long distances. λ close to 1 means that propagation distance is not restricted, whereas small λ means that only close neighbors in the ER graph get significant activation. The optimal choice of λ is task dependent. An example is discussed in the experimental section.

We compute the solution of the resulting sparse linear equation efficiently with an iterative sparse equation solver, namely GMRES (Saad & Schultz, 1986) in Java[2]. Alternatively, this equation could be solved with locally optimal updates which are performed until convergence; this would correspond the Gauss-Seidel method for solving the sparse linear system.

[1] `http://lucene.apache.org`

[2] `http://code.google.com/p/matrix-toolkits-java/`

2.2. Structured Search and Final Result Ranking

In a next step, we filter out relevant entities and facts using a SPARQL-select query. We store the ER graph in a RDF store [3] where we execute the SPARQL query. The result is table where the columns encode the different variable bindings and the rows are instances found in the database, generally not sorted. Each row thus consists of one or more references to entities or literals, and potentially expresses one or more facts about these objects.

Having performed the keyword-based scoring of all the nodes in the ER graph, we now rank the rows returned by the SPARQL query such that more relevant entities or facts are placed on top of the result list. At the moment, we simply sum-up the relevances of all entities in each row. This roughly expresses an OR semantics for ranking. In the future, we will also investigate other combination rules.

3. Experiments

In order to evaluate the proposed approach we selected the YAGO knowledge base[4] (Suchanek et al., 2007) as a structured information source and Wikipedia as a textual resource. YAGO consists of approx. 2 million entities and more than 20 million facts describing these entities (Suchanek et al., 2007). Facts are automatically extracted from Wikipedia and combined with concepts from Wordnet (Fellbaum, 1998). YAGO's accuracy is estimated to be about 95%. In addition, many YAGO entities are linked to Wikipedia pages via the relation `describes` and we thus obtain an interlinked, textually enriched ER graph.

The full text of all Wikipedia pages is indexed with Lucene. We adapted the standard Lucene score by adding a normalization factor that takes into account the length of the document. This normalization factor is necessary due to the nature of encyclopaedic-style document collections, where important or famous entities tend to have longer than average textual descriptions.

In the following we present three show cases. We focus on examples, where it is either difficult or impossible to retrieve particular information with a standard SPARQL query or a keyword search alone. Qualitatively, the examples can be grouped as follows:

- *Context-Aware Fact Search:* Search for entities

[3] Ontotext OWLIM `http://www.ontotext.com/owlim/`

[4] version from February 1st 2010

and facts by specifying the interesting aspects with keywords (see **Example 1+2**).

- *Context-Aware Category Search:* Search with keywords for abstract categories which are not linked to a special textual description (see **Example 3**).

3.1. Context-Aware Entity Search

In this setting we would like to retrieve entities or facts about entities where the specific context is specified via keywords.

Example 1 *Give me companies with number of employees and annual revenue which have sth. to do with ultrasound.*
In this example, we query our hybrid search engine with the keyword "ultrasound" and a SPARQL-select with the following WHERE-clause,

```
?company rdf:type wordnet_company
?company yago:hasEmployeees  ?employees
?company yago:hasRevenue ?revenue
```

The results of our approach are presented in Table 3.1. The 10 top-ranked results are all companies that produce ultrasound devices. An exception here is Turtle Beach Syst., which produces a PC sound card named "Ultrasound".

Note that there is no category for companies that produce ultrasound devices in YAGO. Thus, this question could not have been answered with a structured query on YAGO alone, which could have only retrieved general companies. At the same time, a keyword-only query on Wikipedia with the keywords "company ultrasound" produces only two companies while the other returned items are related to technology pages. This example thus demonstrates the need for hybrid search techniques.

Moreover, note that our approach directly returns not only relevant company names, but also their number of employees and the revenue. Our approach thus goes beyond keyword-aware faceted search.

Example 2 *Give me physicists and their advisors that worked in the area of quantum mechanics and have sth. to do with Los Alamos.*
The results of our hybrid approach are shown in Table 2. Again the result is very reasonable. The first result, Robert Oppenheimer, was the director of the famous Los Alamos Scientific Laboratory, and both Oppenheimer and his advisor Born worked in the area of quantum mechanics. But also the other persons obtained with our approach were famous quantum physicists with connections to Los Alamos laboratories.

r	Company	Employees	Revenue	Score
1	General Electric	327000	$ 172.738 M.	9083.89
2	GE Healthcare			8419.13
3	Philips	125500	26.976 M.	8404.8
4	Siemens AG	430000	$ 110,820 M.	8092.45
5	Neusoft Group	12000	$ 355 M.	4640.18
6	SRI International			4299.93
7	Agfa-Gevaert	13565	3.300 M.	3759.03
8	Foster-Miller			3055.97
9	Ellex Medical Las.			3011.97
10	Turtle Beach Syst.			2958.37

Table 1. Ranked results of our approach for example 1 (keyword "ultrasound").

3.2. Context-Aware Category Search

Relevance propagation allows us to generalize keyword relevance of a subset of nodes to all nodes in the ER graph, even if some of them are not linked to texts themselves. Such nodes are for example the category nodes in YAGO, which group a number of entities with respect to a specific topic or property, but which do not have a description other than the title of the category. With the help of relevance propagation, our system is thus nevertheless able to search for categories by specifying context related keywords.

Example 3 *Give me categories related to microscopes.*
Here, we query our search engine with the keyword "microscope" and a SPARQL-select with WHERE-clause,

```
?category rdf:type ?concept
```

To give a notion about the influence of the weighting factor λ, we assess the distance of the shortest path D between the nodes that obtained a Lucene score and the nodes from the top ten result set. Table 3 shows the results and corresponding distances for $\lambda = 1$ (respectively Table 4 for $\lambda = 0.1$). It can be seen that for $\lambda = 1$, the top ten results are enriched with nodes that have a longer shortest path distance than the top ten results for $\lambda = 0.1$.

In both cases the results are highly plausible. They can be grouped roughly into two types: Categories that get investigated with the help of microscope techniques, e. g. *plants* or *pathogens*, and categories that are technically related to the concept microscope, e. g. *X-rays*.

r	Physicist	Advisor	Score r_i
0	Robert Oppenheimer	Max Born	86579.94
1	David Bohm	Robert Oppenheimer	77108.46
2	Willis Lamb	Robert Oppenheimer	62835.95
3	Philip Morrison	Robert Oppenheimer	61497.02
4	Richard Feynman	John Archibald Wheeler	54672.82
5	George Zweig	Richard Feynman	52808.75
6	Chen Ning Yang	Edward Teller	47098.77
7	Edward Teller	Werner Heisenberg	46347.56
8	Lincoln Wolfenstein	Edward Teller	45903.60
9	John von Neumann	Leopold Fejr	34172.28
10	Emilio G. Segre'	Enrico Fermi	31561.64
10	Enrico Fermi	Luigi Puccianti	31561.64

Table 2. Ranked results of our approach for example 2 (keywords "Los Alamos" and "Quantum")

4. Related Work

Unlike our proposed approach, most semantic search engines aim at retrieving relevant documents supported by semantic annotations, see (Mangold, 2007) for a recent survey. Our goal, however, is to directly retrieve entities and facts from the ER graph, such that the user does not have to search through the documents himself.

This paradigm is similar to structured search engines such as NAGA (Kasneci et al., 2008), where a flexible subgraph pattern is given to retrieve pieces of an ER graph. While they also rank the results of a structured query, they do not take into account keywords.

Searching and ranking entities given a keyword query is done for the scientific domain by (Nie et al., 2005). For instance, the Libra system[5] returns lists of conferences, persons and research papers. However, this system does not return facts and no generalization by relevance propagation is utilized.

Another attempt to bridge the gap between textual input and structured search is the translation of natural language or keyword queries into a structured formalism. However, this involves advanced understanding of language, a highly ambitious effort. To alleviate this problem (Tran et al., 2007) introduce a method that maps keyword queries to entities in a knowledge base.

We also refer to work in the traditional database community that provides mechanisms to make Relational Database Management Systems (RDBMS) searchable with keywords, see e. g. (Bhalotia et al., 2002; Agrawal et al., 2002). In contrast, our aim is to use the effectiveness and richness of textual features to *rerank* the formal query and thus to narrow down the user intention.

Most similar to our approach is the work proposed by (Rocha et al., 2004). In their work the authors describe a hybrid approach for searching ER-graph like data repositories. As we do in our work, they also use the idea relevance propagation or activation spreading. Our work can be seen as an extension of this work by including a reranking component based on the user's intention expressed via keywords. More precisely, activation spreading (Crestani, 1997) does not include IR-style ranking scores when retrieving relevant results.

5. Conclusions

We have presented a framework for hybrid search on textually enriched ER graphs. It integrates flexible and intuitive keyword search with the specificity of structured query languages. This is advantageous in several respects: First, keywords are very flexible and allow the incorporation of unstructured information that is not made explicit in the semantic structure of the ER graph. At the same time the graph structure is exploited both for a suitable generalization of keyword relevance via relevance propagation, as well as for a precise filtering in terms a given SPARQL query. Our method can not only retrieve lists of entities based on a hybrid query, but gives direct access to lists of facts that otherwise would have to be collected manually starting from each relevant entity.

There are several directions to extend and improve our work. While preliminary results have shown the effectiveness of our proposed approach, a quantitative evaluation in terms of retrieval performance will be a matter of ongoing research. Second, we have so far not used formal reasoning in the ER graph. Such additional, logic-based generalization would have to be

[5] `http://academic.research.microsoft.com/`

compared with the more probabilistic generalization step that we have included via the relevance propagation formalism. We imagine that these two approaches could well complement each other.

In total, we feel that, while information retrieval in the structured and in the unstructured domain separately are well-established, searching and retrieving information on mixed data sources such as textually enriched entity relationship graphs will be a growing field of interest. There are many more interesting problems that can be looked at such as, for example, clustering or hierarchical question answering. With the advent of linked data stores like Linked Life Data or YAGO/Wikipedia there are many large and interesting data sources available for experiments with such tasks. While we have not demonstrated our method on corporate intranet data, we believe that this area could also be a major application field for the proposed method.

References

Agrawal, S., Chaudhuri, S., & Das, G. (2002). DBXplorer: A system for keyword-based search over relational databases. *ICDE '02: Proceedings of the 18th International Conference on Data Engineering* (p. 5).

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., & Ives, Z. (2007). Dbpedia: A nucleus for a web of open data. *Lecture Notes in Computer Science*, *4825*, 722.

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., & Ives, Z. (2008). Dbpedia: A nucleus for a web of open data. *Proceedings of 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference (ISWC+ASWC 2007)* (pp. 722–735). Busan, Korea.

Bhalotia, G., Nakhe, C., Hulgeri, A., Chakrabarti, S., & Sudarshan, S. (2002). Keyword searching and browsing in databases using banks. *ICDE*.

Bizer, C., & Cyganiak, R. (2006). D2R Server–publishing relational databases on the semantic web. *5th International Semantic Web Conference*.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. *Computer networks and ISDN systems*, *30*, 107–117.

Bundschus, M., Dejori, M., Stetter, M., Tresp, V., & Kriegel, H. P. (2008). Extraction of semantic biomedical relations from text using conditional random fields. *BMC Bioinformatics*, *9*, 207+.

Crestani, F. (1997). Application of spreading activation techniques in information retrieval. *Artificial Intelligence Review*, *11*, 453–482.

Erling, O., & Mikhailov, I. (2009). RDF Support in the Virtuoso DBMS. *Proceedings of the 1st Conference on Social Semantic Web* (pp. 1617–5468).

Fellbaum, C. (1998). *Wordnet: An electronical lexical database*. Cambridge, MA: The MIT Press.

Kasneci, G., Suchanek, F., Ifrim, G., Ramanath, M., & Weikum, G. (2008). Naga: Searching and ranking knowledge. *Proc. of ICDE* (pp. 1285–1288).

Mangold, C. (2007). A survey and classification of semantic search approaches. *International Journal of Metadata, Semantics and Ontologies*, *2*, 23–34.

Nie, Z., Zhang, Y., Wen, J., & Ma, W. (2005). Object-level ranking: Bringing order to web objects. *Proceedings of the 14th international conference on World Wide Web* (p. 574).

r	Category	score	D
0	wikicategory wordnet microscope	1234643.5	3
1	wikicategory Scientific techniques	1112785.29	3
2	wordnet disease	962138.78	1
3	wordnet person	632517.98	1
4	wikicategory wordnet optics	521473.51	3
5	wikicategory Plant pathogens and diseases	486216.12	3
6	wikicategory wordnet measuring instruments	322737.59	3
7	wikicategory X-rays	281627.11	3
8	wordnet anatomy	280101.66	1
9	wikicategory wordnet igneous rock	230442.75	3

Table 3. Results of our system for example 3 (keyword "microscope", $\lambda = 1$)

r	Category	score	D
0	wikicategory wordnet microscope	22079.62	3
1	wikicategory Scientific techniques	11873.37	3
2	wordnet disease	7419.3	1
3	wikicategory wordnet optics	6621.56	3
4	wordnet person	4254.36	1
5	wordnet scientist	4056.44	1
6	wikicategory wordnet lens	3426.64	3
7	wikicategory Types of cancer	2964.22	1
8	wikicategory Technology timelines	2849.33	1
9	wikicategory wordnet genetics	2477.66	1

Table 4. Results of our system for example 3 (keyword "microscope", $\lambda = 0.1$)

Ontotext (2009). Linked life data. `http://www.linkedlifedata.com/`.

Rocha, C., Schwabe, D., & Aragao, M. (2004). A hybrid approach for searching in the semantic web. *Proceedings of the 13th international conference on World Wide Web* (pp. 374–383).

Saad, Y., & Schultz, M. (1986). GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, *7*, 856–869.

Sarawagi, S. (2008). Information extraction. *Found. Trends databases*, *1*, 261–377.

Suchanek, F. M., Kasneci, G., & Weikum, G. (2007). Yago: A Core of Semantic Knowledge. *16th international World Wide Web conference (WWW 2007)*. New York, NY, USA: ACM Press.

Suchanek, F. M., Sozio, M., & Weikum, G. (2009). Sofie: a self-organizing framework for information extraction. *WWW '09: Proceedings of the 18th international conference on World wide web* (pp. 631–640). New York, NY, USA: ACM.

Tran, T., Cimiano, P., Rudolph, S., & Studer, R. (2007). Ontology-based interpretation of keywords for semantic search. *Lecture Notes in Computer Science*, *4825*, 523.

W3C (2007). SPARQL Query Language for RDF. `http://www.w3.org/TR/rdf-sparql-query/`.