CE2 – Towards a Large Scale Hybrid Search Engine with Integrated Ranking Support

Haofen Wang
Department of Computer Science & Engineering
Shanghai Jiao Tong University
Shanghai, 200240, China
whfcarter@apex.sjtu.edu.cn

Thanh Tran
Institute AIFB
Karlsruhe University
Karlsruhe, Germany
dtr@aifb.uni-karlsruhe.de

Chang Liu
Department of Computer Science & Engineering
Shanghai Jiao Tong University
Shanghai, 200240, China
liuchang@apex.sjtu.edu.cn

ABSTRACT

The Web contains a large amount of documents and increasingly, also semantic data in the form of RDF triples. Many of these triples are annotations that are associated with documents. While structured query is the principal mean to retrieve semantic data, keyword queries are typically used for document retrieval. Clearly, a form of hybrid search that seamlessly integrates these formalisms to query both documents and semantic data can address more complex information needs. In this paper, we present CE^2, an integrated solution that leverages mature database and information retrieval technologies to tackle challenges in hybrid search on the large scale. For scalable storage, CE^2 integrates database with inverted indices. Hybrid query processing is supported in CE^2 through novel algorithms and data structures, which allow for advanced ranking schemes to be integrated more tightly into the process. Experiments conducted on Dbpedia and Wikipedia show that CE^2 can provide good performance in terms of both effectiveness and efficiency.

Categories and Subject Descriptors

E.2 [**Data Storage Representations**]; H.3 [**Information Storage and Retrieval**]

General Terms

Algorithms, Design, Experimentation, Performance

Keywords

Hybrid Search, Annotations, Scalable Storage, Ranking

1. INTRODUCTION

Recently, we have seen a strong increase in the availability of semantic data, *RDF triples* in particular. DBPedia and DBLP are two example repositories that contain millions of triples. Often, this data is associated with documents in the form of annotations. Currently, keyword search is commonly supported for the retrieval of documents. Beyond document retrieval, semantic data available on the Web represents a source of knowledge, which can addresses other retrieval scenarios. Instead of documents, the engine might deliver concrete answers to the user as a result of a complex question that is represented as a structured query. More specialized systems such as Digital Library applications also employ *hybrid queries*, a special structured query that also contains keyword elements. This type of queries is ideal for doing *hybrid search* [1], a paradigm that allows for querying over textual data and semantic data in an integrated way.

Hybrid search on the large scale web environment however bring about more challenges. In particular, it requires the capability to *store*, *index* and integrate the large amount of documents and semantic data. Also, there is a need for *scalable query processing* such that results can be returned with satisfactory response time. Another aspect that is crucial in the Web environment is *ranking*. In this paper, we elaborate on infrastructure components that are necessary to support large scale hybrid search. In particular, we address the above challenges through the following contributions:

- We elaborate on a unified framework to represent and to query both RDF data and documents in an integrated way.
- We leverage mature database and information retrieval technologies for the development of an integrated repository. Furthermore, we propose novel algorithms and data structures for hybrid query processing that is tightly integrated with a flexible ranking mechanism.
- The repository and the hybrid query engine have been embedded into an integrated solution called CE^2 to store, index and to query large amount of RDF data and documents.

2. HYBRID SEARCH

2.1 Resources

Resources in hybrid search are represented through a graph based model G that contains entities, documents, their relations and attributes.

Definition 1. A *resource graph* G is a tuple (V, L, E) where

- V is a finite set of *vertices*, namely the disjoint union $V_E \uplus V_D \uplus V_C \uplus \{V_{Doc}\} \uplus V_L \uplus V_{Text}$ with V_E representing

CIKM'08, October 26–30, 2008, Napa Valley, California, USA.

entities, V_D representing documents, V_C representing entity classes, V_{Doc} representing the document class, V_L representing literals, and V_{Text} representing texts.

- L is a finite set of *edge labels*, namely the disjoint union $L_R \uplus L_A \uplus \{type, subclass, annotation, keyword\}$ standing for inter-entity edges L_R and entity-attribute edges L_A.
- E is a finite set of *edges* of the structure $e(v_1, v_2)$ with $v_1, v_2 \in V$ and $e \in L$. Moreover, the following restrictions apply:
  - $e \in L_R$ if and only if $v_1, v_2 \in V_E$,
  - $e \in L_A$ if and only if $v_1 \in (V_E \uplus V_D)$ and $v_2 \in V_L$,
  - $e = keyword$ if and only if $v_1 \in (V_E \uplus V_D)$ and $v_2 \in V_{Text}$,
  - $e = type$ if and only if $v_1 \in V_E$ and $v_2 \in V_C$, or $v_1 \in V_D$ and $v_2 = V_{Doc}$,
  - $e = subclass$ if and only if $v_1, v_2 \in V_C$,
  - $e = annotation$ if and only if $v_1 \in V_D$ and $v_2 \in (V_E \uplus V_C \cup G)$.

Vertices corresponding to entities and documents are identified by Uniform Resource Identifiers (URIs). As identifiers for other elements, we use class names, literals, texts and edge labels respectively. Note that *annotation* plays a particularly important role in hybrid search as it captures the associations between a document and other resources. There are entity annotations, class annotations and complex annotations such as triples that are represented in terms of a resource graph.

2.2 Queries and Answers

In order to query the resources defined above, we extend the notion of conjunctive queries defined in [2] as follows:

Definition 2. A hybrid query q is an expression of the form $(x_1, \ldots, x_k).\ \exists x_{k+1}, \ldots x_m.\ A_1 \wedge \ldots \wedge A_r$, where $x_1, \ldots, x_k$ are called distinguished variables, $x_{k+1}, \ldots, x_m$ are called undistinguished variables and $A_1, \ldots, A_r$ are query atoms. These atoms are of the form $p(v_1, v_2)$, where $p \in L \backslash \{subclass\}$ is called predicate, and v_1, v_2 are called variables or terms.

We further restrict the query pattern to be tree shaped with only one single distinguished variable that is the root. This leads to a more efficient search process while a large portion of user information needs can still be expressed.

A solution to q on a resource graph G is a mapping from the variables in the query to vertices e such that the substitution of variables in the graph pattern would yield a subgraph of G. The *substitutions of distinguished variables* constitute the query answers.

2.3 Ranking

Considering the specific nature of hybrid queries, two principles are proposed to guide the design of the ranking scheme.

- **Quality propagation.** Besides the quality of an entity or a document, the qualities of its connected entities or documents should also be taken into account due to the nature of hybrid search. An entity or a document should receive higher score when connecting to higher quality neighbors. When finding the successors of US presidents related to war, J.F Kennedy should be ranked first in the result because that his predecessor Eisenhower had the tightest connection to war.
- **Quantity aggregation.** the number of connected entities or documents should also be considered. Thus, when an answer has more neighbors, it tends to be ranked higher. In the query "Find institutions that Turing Award winners work in", CMU, UCB, and IBM are the top 3 institutions, because that they have the largest number of Turing Award winners, according to Wikipedia.

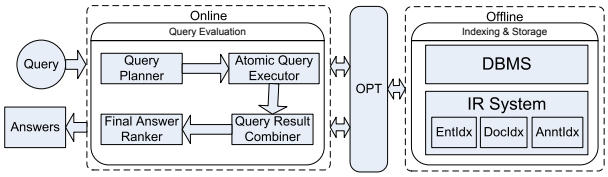

Figure 1: Architecture of CE^2.

3. CE^2 – AN INTEGRATED SOLUTION

CE^2 comprises of a repository and a query engine. Fig. 1 shows the decomposition of CE^2 into two main components. The first one is the repository. Text data associated with entities and documents as well as annotations are stored in separate inverted indices, and the remaining semantic data is kept in the database. The second component is a hybrid query engine that comprises of several submodules. The Query Planner decomposes a query into several parts, which are processed by Atomic Query Executor. These results are fed into the Query Result Combiner and finally, to the Answer Ranker.

4. PRELIMINARY RESULTS

CE^2 is implemented in Java on top of a DB2 database and Lucene. Experiments are conducted on RDF data contained in DBpedia and documents from Wikipedia. We compare the ranked results produced by CE^2 against keyword search in Lucene and query support in DB2 with Text Extender. It achieves 40 percent and 20 percent improvements for P@10 respectively while preserving affordable response time.

5. CONCLUSIONS AND OUTLOOK

In this paper, we have elaborated on a model for hybrid search. With respect to this model, we have leveraged database and IR technologies to scale over large amount of resources. In particular, we have presented CE^2 to support hybrid query processing against these resources with tightly integrated ranking. We further plan to exploit data uncertainty for ranking and adopt more advanced query optimization strategies into the evaluation of hybrid queries.

6. REFERENCES

[1] R. Bhagdev, S. Chapman, F. Ciravegna, V. Lanfranchi, and D. Petrelli. Hybrid search: Effectively combining keywords and semantic searches. In *ESWC*, 2008.

[2] I. Horrocks and S. Tessaris. Querying the semantic web: a formal approach. In *ISWC*, 2002.