Dataplorer: A Scalable Search Engine for the Data Web

Haofen Wang, Qiaoling Liu,
Gui-Rong Xue, Yong Yu
Shanghai Jiao Tong University
Shanghai, 200240, China
{whfcarter, lql, grxue,
yyu}@apex.sjtu.edu.cn

Lei Zhang, Yue Pan
IBM China Research Lab
Beijing, 100094, China
{lzhangl, panyue}@cn.ibm.com

ABSTRACT

More and more structured information in the form of semantic data is nowadays available. It offers a wide range of new possibilities especially for semantic search and Web data integration. However, their effective exploitation still brings about a number of challenges, e.g. usability, scalability and uncertainty. In this paper, we present Dataplorer, a solution designed to address these challenges. We consider the usability through the use of hybrid queries and faceted search, while still preserving the scalability thanks to an extension of inverted index to support this type of query. Moreover, Dataplorer deals with uncertainty by means of a powerful ranking scheme to find relevant results. Our experimental results show that our proposed approach is promising and it makes us believe that it is possible to extend the current IR infrastructure to query and search the Web of data.

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

General Terms: Algorithms, Performance, Experimentation

Keywords: hybrid query, inverted index, ranking, faceted search

1. INTRODUCTION

Structured information is nowadays available and grows rapidly to form a *Web of data*. It offers a wide range of new possibilities that were until now out of reach. To prepare the arrival of new applications and scenarios benefiting from this information, several initiatives have appeared, among which it is possible to notice the Linking Open Data project [1].

However, the effective exploitation of the Web of data still brings about a number of challenges: **Usability** – Structured queries cannot be imposed as such to the end-user, usually lacking knowledge about the language itself and the data schema. **Scalability** – As the already impressive amount of data available is ever growing, efficient search solutions able to scale to this level are becoming essential. **Uncertainty** – The diverse provenance of the data, as well as the different ways it can be interpreted or queried require the consideration of the uncertainty and imperfection.

In this paper, we present Dataplorer, a solution designed to address these challenges without sacrificing one for the sake of the others. The usability challenge is addressed by providing the user with *hybrid query* capabilities, leveraging the power of structured queries and the ease of use of keyword search. We also propose a *faceted search* functionality that allows users to progressively compose the structured part of their information need after having started with imprecise keywords. Scalability is one of the main challenges that hybrid queries are facing, due to the large amount of data. Inspired from the cross field of DB and IR integration, we make IR compatible with hybrid search through an *extension of the inverted index*, and thus able to scale as well as to handle structured information. To ensure that uncertainty does not remain as a problem to return relevant results, we provide a powerful *ranking scheme* that considers structures of both data and hybrid queries for score propagation and aggregation during results ranking. As an improvement of our previous work [3], we support faceted search with integrated ranking to tackle both usability and uncertainty issues while preserving efficiency.

WWW 2009, April 20–24, 2009, Madrid, Spain.
ACM 978-1-60558-487-4/09/04.

2. SYSTEM OVERVIEW

The architecture is presented by Figure 1. It is composed of three main components: a front-end interface, needed to address the usability challenges by making the functionalities accessible to the user, a search component, in charge of retrieving and ranking the results, and a component used to index the Web of data in an efficient way. The system demo can be found through `http://dataplorer.apexlab.org`.

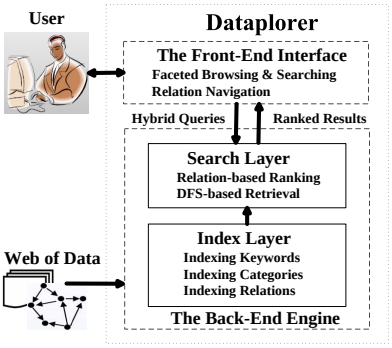

Figure 1: Dataplorer Architecture

2.1 Hybrid Query Capability

Let us assume that a user called Alice is willing to find information about martial art films directed by Hong Kong

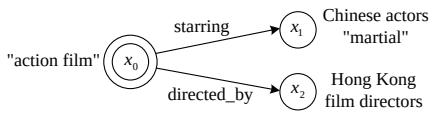

Figure 2: An Example Hybrid Query

Table 1: Translating data into "Documents" in IR

Document	Field	Term
concept C	text	tokens in textual properties
relation R	text	tokens in textual properties
individual i	type	concepts that i belongs to
	subjOf	all relations R that $(i, R, ?)$ is a triple in data
	objOf	all relations R that $(?, R, i)$ is a triple in data
	text	tokens in textual properties of i

directors and starring Chinese martial art actors. The ability to use a conjunction of keywords like "martial art" and precise semantic relations like "directed_by" gives the opportunity to improve the user experience when addressing complex information needs. Figure 2 depicts the example query. The definition of the hybrid queries used by Dataplorer is based on the work previously done by [2] focusing on conjunctive queries that we extended to combine keyword and structured data search. A *hybrid conjunctive query* q is a query expression of the form $q(x) \leftarrow \exists \overrightarrow{y}.conj(x, \overrightarrow{y})$, where x is called the target variable, $\overrightarrow{y}$ are existentially quantified variables called non-distinguished variables, and $conj(x, \overrightarrow{y})$ is a conjunction of terms of the form $C(z)$, $R(z_1, z_2)$, or $R^-(z_1, z_2)$. z, z_1, z_2 are individuals or variables in x or $\overrightarrow{y}$. R is a relation name and C represents a concept expression that is a boolean combination of concept names and keyword concept W. In this paper, we restrict to queries whose graph patterns are *trees*, as in [2]. We also restrict the query result to be *unary* (i.e., a single target variable x in the query). These restrictions lead to a much more simplified and hence efficient query evaluation procedure, while a large portion of information needs can still be expressed.

2.2 Data Indexing

Inverted index is based on the concepts of documents, fields, and terms. Each term is associated with a posting list of documents containing it. In addition, for each of these documents, there is a list of positions showing where the term appears in it. Current search engines have proved this technique to easily scale to the Web. By leveraging the index structure, we translate the Web of data into documents, fields and terms in a proper way, as shown by Table 1. In order to index a relation instance (s, R, o), We use the position list to store the objects of relation R under subject s, where s is treated as a document with a field named `subjOf` and a term R in the field. Similarly, the `objOf` field can be considered as indexing instances of the relation R^-.

2.3 Faceted Search with Integrated Ranking

Dataplorer efficiently answers a tree-shaped hybrid query using a DFS (i.e. Depth First Search) based retrieval algorithm according to a well-optimized query plan. The query evaluation is reduced to three basic operations over posting lists including (1) Basic-retrieval $b(f, t)$. Given a field f and a term t, it retrieves the corresponding posting list as an Ascending Integer Stream (AIS), which is accessed from the smallest integer to the largest one. (2) Merge-sort $m(S_1, \cap, S_2)$. Given two AISs S_1 and S_2, it computes $S_1 \cap S_2$ and returns a new AIS. (3) Mass-union $u(S, R)$. Given a relation R and a AIS S, it computes the set $\{o \mid \exists s : s \in S \wedge (s, R, o)\}$ and returns it as a AIS.

We propose a ranking scheme according to ranking principles *quality propagation* and *quantity aggregation*, which means that the higher the quality of a neighbor (or the more the number of its neighbors), the higher the rank of a result. First, we attach a scoring capability to each AIS and obtain a Scored AIS (SAIS), e.g. $S' = \{\langle d_1, 0.4\rangle, \langle d_3, 0.8\rangle, ...\}$. We then generalize the three basic operations to $b'(f, t)$, $m'(S'_1, \cap, S'_2)$, $u'(S', R)$, so that they are adapted to SAIS. For basic-retrieval $b'(f, t)$, we score an item in its result by $b'(f, t)[d] = RSV(t, d)$ where $RSV(t, d)$ computes the relevance of d with respect to t according to the tf-idf principle and returns a degree in [0,1]. For merge-sort $m'(S'_1, \cap, S'_2)$, we score an item in its result by $m'(S'_1, \cap, S'_2)[d] = S'_1[d] \cdot S'_2[d]$. For mass-union $u'(S', R)$, we score an item in its result by $u'(S', R)[o] = 1 - \prod_{s:s\in S'\wedge(s,R,o)}(1 - S'[s])$. Thus, the ranking scheme is tightly integrated in query processing.

Faceted search is recognized as an effective means to explore the Web of data. Starting from a keyword query, the user gets the ranked results as well as the corresponding available facets (i.e. concepts and relations) to construct hybrid queries iteratively. Dataplorer computes the data distribution of all associated facets. We reduce the computation to hybrid queries by modeling concepts and relations as "general documents" and introducing TYPE, SUBJECT_OF, OBJECT OF as "general relations".

3. EVALUATION

For the experiments, we used DBLP, DBpedia, Freebase, USCensus, GeoNames, SW.org, and SWRC. Totally, the indexed data adds up to 1.1 billion triples. We compared the query performance of Dataplorer with SOR[1] and RDF-3X[2] on 45 structured conjunctive queries with different shapes, lengths and variable numbers. Thanks to the benefit of spatial locality for fast access using the extension of the inverted index, Dataplorer outperforms RDF-3X in most cases although RDF-3X has employed well-designed compression and optimization technologies. Dataplorer's performance is comparable to that of SOR with cache, and is one or two orders of magnitude faster than SOR when performing cold-start. We further created 20 hybrid queries from the questions collected from 10 users. Dataplorer achieved more than 20% improvements on $P@10$ compared with that of traditional keyword search systems and database with full-text extensions. Moreover, all these queries can be answered within one second which indicates the good efficiency as well.

4. REFERENCES

[1] C. Bizer, T. Heath, K. Idehen, and T. Berners-Lee. Linked data on the web. In *Proc. of WWW*, 2008.
[2] I. Horrocks and S. Tessaris. Querying the semantic web: A formal approach. In *Proc. of ISWC*, 2002.
[3] L. Zhang, Q. Liu, J. Zhang, H. Wang, Y. Pan, and Y. Yu. Semplore: An IR Approach to Scalable Hybrid Query of Semantic Web Data. In *Proc. of ISWC*, 2007.

[1] http://www.alphaworks.ibm.com/tech/semanticstk
[2] http://www.mpi-inf.mpg.de/ neumann/rdf3x