Extracting Authoring Information Based on Keywords and Semantic Search

Faisal Alkhateeb, Amal Alzubi, Iyad Abu Doush
Computer Sciences Department
Yarmouk University, Irbid, Jordan
{alkhateebf,iyad.doush}@yu.edu.jo

Shadi Aljawarneh
Faculty of Science and Information Technology
Al-Isra University, Amman, Jordan
shadi.jawarneh@ipu.edu.jo

Eslam Al Maghayreh
Computer Sciences Department
Yarmouk University, Irbid, Jordan
eslam@yu.edu.jo

ABSTRACT

Many people, in particular researchers, are interested in searching and retrieving authoring information from online authoring databases to be cited in their research projects. In this paper, we propose a novel approach for retrieving authoring information that combines keyword and semantic-based approaches. In this approach, the user is interested only in retrieving authoring information considering some specified keywords and ignore how the internal semantic search is being processed. Additionally, this approach exploits the semantics and relationships between different resources for a better knowledge-based inference.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Search process

Keywords

Semantic web, RDF, SPARQL, Authoring Information, Keyword Search, Semantic Search

1. INTRODUCTION

The world wide web (or simply the web) has become the first source of knowledge for all life domains. It can be seen as an extensive information system that allows exchanging the resources as well as documents. The semantic web is an evolving extension of the web aiming at giving well defined forms and semantics to the web resources (e.g., content of an HTML web page) [4].

Due to the growth of the semantic web, semantic search became an attracting approach. The term refers to methods of searching web documents beyond the syntactic level of matching keywords. Exposing metadata is an essential point for a semantic search approach associated with the semantic web. The most important recent development is in the area of embedding metadata directly into web documents. RDF (Resource Description Framework) [15] is a knowledge representation language dedicated to the annotation of resources within the Semantic web. Currently, many documents are annotated via RDF due to its simple data model and its formal semantics. For example, it is embedded in (X)HTML web pages using the RDFa language [1], in SMIL documents [7] using RDF/XML [3], etc. SPARQL [17] is a W3C recommendation language developed in order to query RDF knowledge bases, e.g., retrieving nodes from RDF graphs.

Another approach that is found in search engines is based on using keywords. More precisely, both queries and documents are typically treated at a word or gram level (similar to Information Retrieval). The search engine is missing a semantic-level understanding of the query and can only understand the content of a document by picking out documents with the most commonly occurring keywords.

The objective of this paper is to provide a novel approach for retrieving authoring information that combines keyword-based and semantic-based approaches. In this approach, the user is interested only in retrieving authoring information considering some specified keywords and ignores how the internal semantic search is being processed. In particular, the user is interested in searching authoring information from online authoring information portals (such as DBLP[1], ACM[2], IEEE[3], etc). For instance, show me all documents of the author "faisal alkhateeb" or the author "jerome euzenat" with a title containing "SPARQL". In the proposed approach, keywords are used for collecting authoring information about the authors, which are then filtered with semantic search (using RDF and SPARQL) based on the semantic relations of the query.

The remainder of the paper is organized as follows: we introduce the research background in Section 2. The combined approach is presented in Section 3 as well as a testcase illustrating the proposed approach. A review of related work is discussed in Section 4. Discussion issues drawn from this study are presented in Section 5.

2. RESEARCH BACKGROUND

This section provides an overview of the elements that are necessary for presenting the proposed approach namely:

[1] `http://www.informatik.uni-trier.de/~ley/db/`
[2] `http://portal.acm.org/portal.cfm`
[3] `http://www.ieee.org/portal/site`

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
ISWSA'10, June 14–16, 2010, Amman, Jordan.
Copyright 2010 ACM 978-1-4503-0475-7/06/2010 ...$10.00.

BibTeX, RDF, and SPARQL.

2.1 BibTeX

BibTeX[4][16, 10] is a tool and a file format which are used to describe and process lists of references, mostly in conjunction with LaTeX documents. BibTeX makes it easy to cite sources in a consistent manner, by separating bibliographic information from the presentation of this information. BibTeX uses a style-independent text-based file format for lists of bibliography items, such as articles, books, thesis. Each bibliography entry contains some subset of standard data entries: `author`, `booktitle`, `number`, `organization`, `pages`, `title`, `type`, `volume`, `year`, `institution`, and others. Bibliography entries included in a .bib file are split by types. The following types are understood by virtually all BibTeX styles: `article`, `book`, `booklet`, `conference`, `inproceedings`, `phdthesis`, etc.

EXAMPLE 1. *The following is an instance of a BibTeX element:*

```
@article{DBLP:AlkhateebBE09,
 author    = {Faisal Alkhateeb and Jean-Francois
                Baget and Jerome Euzenat},
 title     = {Extending SPARQL with regular exp-
               ression patterns (for querying RDF)
             },
 journal   = {J. web Sem.},
 volume    = {7},
 number    = {2},
 year      = {2009},
 pages     = {57-73},
       }
```

2.2 RDF

RDF is a language for describing resources. In its abstract syntax, an RDF document is a set of triples of the form `<subject, predicate, object>`.

EXAMPLE 2. *The assertion of the following RDF triples:*

```
{ <ex:person1 foaf:name "Faisal Alkhateeb">,
  <ex:document1 BibTeX:author ex:person1>,
  <ex:document1 rdf:type BibTeX:inproceedings>,
  <ex:document1 BibTeX:title "PSPARQL">,
  <ex:person1 foaf:knows ex:person2>,
  <ex:person2 foaf:name "Jerome Euzenat">,
  <ex:document1 BibTeX:author ex:person2>,
}
```

means that there exists an inproceedings document, which is coauthored by two persons named "Faisal Alkhateeb" and "Jerome Euzenat", whose title is "PSPARQL".

An RDF document can be represented by a directed labeled graph, as shown in Figure 1, where the set of nodes is the set of terms appearing as a subject or object in a triple and the set of arcs is the set of predicates (i.e., if `<s, p, o>` is a triple, then $s \xrightarrow{p} o$).

2.3 SPARQL

SPARQL is the query language developed by the W3C for querying RDF graphs. A simple SPARQL query is expressed using a form resembling the SQL SELECT query:

[4] `http://www.bibtex.org/`

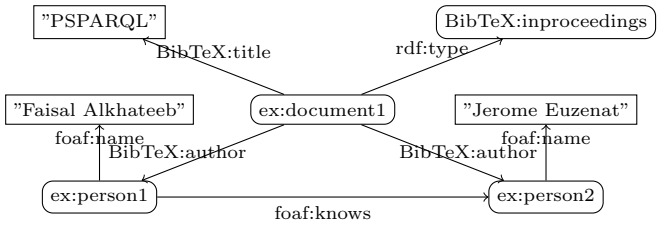

Figure 1: An RDF graph.

$$\texttt{SELECT}\ \vec{B}\ \texttt{FROM}\ u\ \texttt{WHERE}\ P$$

where u is the URL of an RDF graph G to be queried, P is a SPARQL graph pattern (i.e., a pattern constructed over RDF graphs with variables) and $\vec{B}$ is a tuple of variables appearing in P. Intuitively, an answer to a SPARQL query is an instantiation of the variables of $\vec{B}$ by the terms of the RDF graph G such that the substitution of the values to the variables of P yields to a subset of the graph G.[5]

EXAMPLE 3. *Consider the RDF graph of Figure 1 representing some possible authoring information. For instance the existence of the following triples* $\{\langle$*ex:document1, rdf:type, BibTeX:inproceedings*$\rangle$, $\langle$*ex:document1, BibTeX:title, "PSPARQL"*$\rangle\}$ *asserts that there exists an inproceedings document whose title is "PSPARQL".*

The following SPARQL query modeling this information:

```
SELECT  *
FROM  <Figure1>
WHERE  {
         ?document BibTeX:author ?author .
         ?document BibTeX:title "PSPARQL" .
         ?author foaf:name ?name .
       }
```

could be used, when evaluated against the RDF graph of Figure 1, to return the following answers:

#	?document	?author	?name
1	ex:document1	ex:person1	"FaisalAlkhateeb"
2	ex:document1	ex:person2	"JeromeEuzenat"

In RDF there exists a set of reserved words (called RDF Schema or simply RDFS [6]), designed to describe the relationships between resources and properties, e.g., `classA subClassOf classB`. It adds additional constraints to the resources associated to the RDFS terms, and thus permitting more consequences (reasoning).

EXAMPLE 4. *Using the RDF graph presented in Figure 1, we can deduce the following triple* `<ex:document1 rdf:type BibTeX:proceedings>` *from the following triples* `<ex:document1 rdf:type BibTeX:inproceedings>` *and* `<BibTeX:inproceedings rdfs:subClassOf BibTeX:publications>`*. Hence, the following SPARQL query :*

```
SELECT  *
FROM  <Figure1>
WHERE  {
         ?document rdf:type BibTeX:publications .
         ?document BibTeX:author ?author .
```

[5] When using RDFS semantics [6], this intuitive definition is irrelevant and one could apply RDFS reasoning rules to calculate answers over RDFS documents.

```
        ?document BibTeX:title "PSPARQL" .
        ?author foaf:name ?name .
      }
```

returns the same set of answers described in Example 1 because a inproceedings is a subclass of publications.

SPARQL provides several result forms other than SELECT that can be used for formating the query results. For example, a CONSTRUCT query can be used for building an RDF graph from the set of answers to the query. More precisely, an RDF graph pattern (i.e., an RDF involving variables) is specified in the CONSTRUCT clause that will be constructed. For each answer to the query, the variable values are substituted in the RDF graph pattern and the merge of the resulting RDF graphs is computed.[6] This feature can be viewed as rules over RDF permitting to build new relations from the linked data.

EXAMPLE 5. *The following CONSTRUCT query:*

```
CONSTRUCT {?author BibTeX:coauthorof ?document .}
FROM   <Figure1>
WHERE  {
        ?document BibTeX:author ?author .
        ?document BibTeX:title "PSPARQL" .
        ?author foaf:name ?name .
      }
```

constructs the RDF graph (containing the coauthor relation) by substituting for each located answer the values of the variables `?author` *and* `?document` *to have the following graph (as done for SPARQL, we encode the resulting graph in the Turtle language[7]):*

```
@prefix ex: <http://ex.org/> .

ex:person1 BibTeX:coauthorof ex:document1 .
ex:person2 BibTeX:coauthorof ex:document1 .
```

3. METHODOLOGY

The Extracting Authoring Information system which we have implemented is used to achieve the following:

Given: - A user query in the form of textual keywords. **Find**: - A set of BibTeX elements that are relevant to the query.

The proposed methodology consists of the following major phases: connecting to Google search engine, connecting to DBLP page and extracting BibTeX elements, converting BibTeX to RDF and keywords to SPARQL query, and then evaluate the SPARQL query against the RDF document. The first two phases deal with extracting author information based on keyword search, while the third and the fourth represent the semantic search. In the following, we present the basic work flow of the system as well as its main components.

3.1 System Work Flow

As shown in Figure 2, the system works as follows: the user firstly enters the keywords to be searched such as keywords from author name, title of the paper title, year of publication, etc. Then, uses goolge search engine to correct misspelled entered keywords (in particular, names of the authors) as well as finding the pages for the corrected entered keywords (for instance, DBLP pages of the author). After that, BibTeX elements will be extracted and these BibTeX elements will be converted to RDF document. The corrected keywords will be transformed to a SPARQL query to be used for querying the RDF document corresponding to the extracted BibTeX elements.

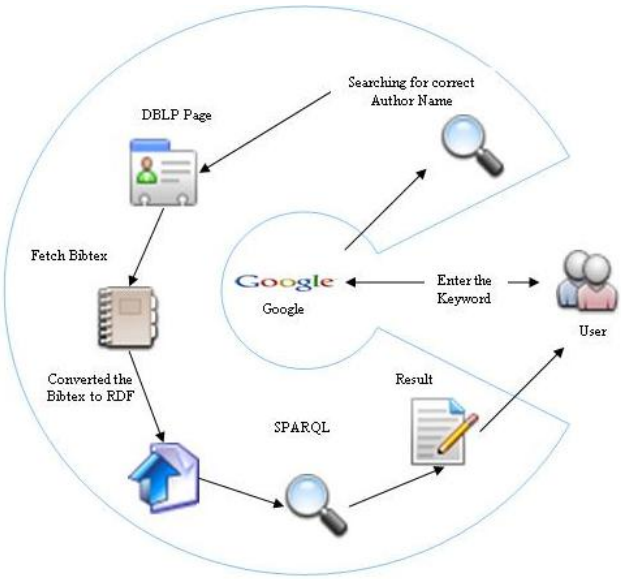

Figure 2: The Basic Flow of the System.

3.2 System Components

The following are the main components of the system:

- **Google Search**: after entering the keywords in the corresponding positions, they will be passed to a component that connects to Google engine. That is, the magic URL "`http://www.google.com/search?hl=ar&q=`" +"searchParameters" of Google search engine will be used to search for the specified keywords. To this end, there could be two cases returned from this search either:
  - correct author name; or
  - misspelled author name.

  In the second, the new search path *"did you mean structure"* will be used to reconnect to the Google search engine. This process is repeated until finding the corresponding author page in the specified authoring database (DBLP, ACM, IEEE, etc).

- **BibTeX extractor**: this component is responsible for extracting the BibTeX elements and save them in a file for later usage. it should be noticed that this component contains several methods, each of them is specific to a bibliography database. This is due to the fact that each bibliography database has its own style to include BibTeX elements in the authoring web pages. Therefore, we suggest to include BibTeX elements in web pages as an RDFa annotations[8].

[6] A definition of RDF merge operation can be found at `http://www.w3.org/TR/2001/WD-rdf-mt-20010925/#merging`.

[7] `http://www.dajobe.org/2004/01/turtle/`.

[8] `http://www.w3.org/TR/xhtml-rdfa-primer/`

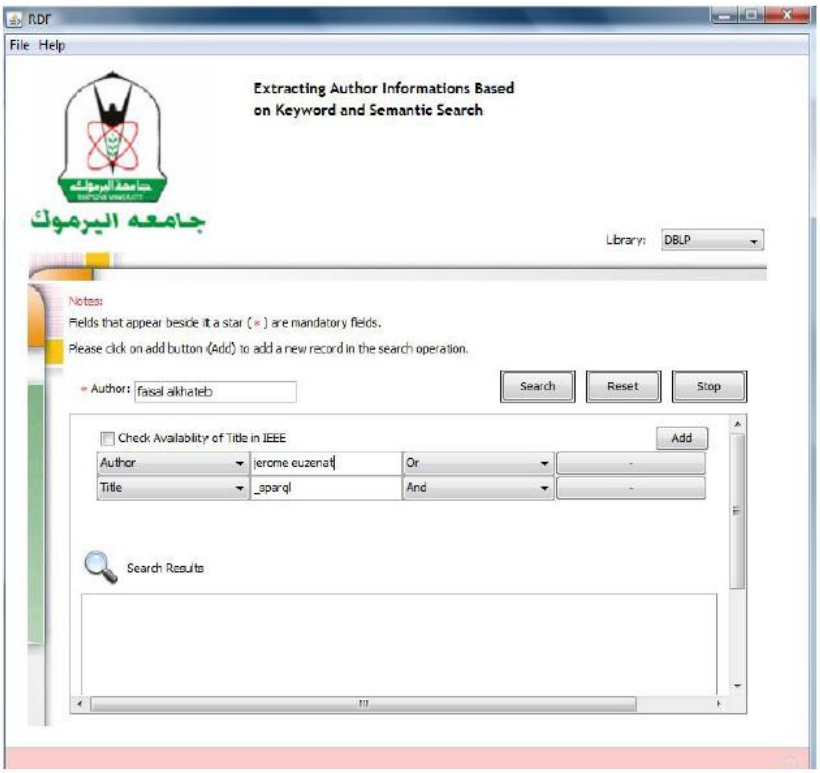

Figure 3: The user interface of the system as well as the found results.

- **BibTeX parser**: BibTeX elements are then converted to RDF documents using results from the BibTeX parser that we have implemented in the system. Note that if the RDFa is used to annotate BibTeX elements, then there is no need for this parser. In this case, the online RDF distiller[9] could be used to extract RDF documents corresponding to the annotated BibTeX elements from web pages. In addition to the RDF triples that correspond to the BibTeX entries, RDF triples corresponding to RDFS relationships (such as `<BibTeX:inproceedings rdfs:subClassOf BibTeX:proceedings>` and `<BibTeX:booklet rdfs:subClassOf BibTeX:book>`) are added to the RDF document to allow reasoning more results.

- **Keywords to SPARQL query**: the entered keywords are also used to build a SPARQL query automatically. The query will be then used to filter the results obtained in search based on keywords. More precisely, when entering keywords, the user selects the type of the data entry to be entered such as "Title", "Author", "Publication", "Pages", and so on. Note that, the user can enter multiple authors. If the keyword begins with underscore "_", this means that the entered keyword is part of the BibTeX data entry. In this case, the "regex" function can be used in the Filter constraint to build the SPARQL query. Otherwise, it is considered to be an exact search for such keyword. Moreover, the user can specify the relationship between the entered keywords (i.e., "or" or "and"). When building the SPARQL query, these relationships correspond to the "UNION" and "AND" SPARQL query graph patterns.

- **Query evaluator**: this component is used to evaluate the SPARQL query (i.e., the query obtained from the entered keywords) against the RDF document (i.e., the RDF document obtained from the file containing the BibTeX elements) to find and construct the precise results. Any query evaluator could be used at this stage[10], but we have used jena[11].

It should be noticed that DBLP provides the capability of searching by allowing users to pose keyword-based queries over only its bibliography dataset. For instance, one can pose the query "alkhateeb|jerome euzenat" that searches for documents matching the keyword "alkhateeb" or "jerome euzenat". The search process in DBLP offers good features such as a search is triggered after each keystroke with instant times if the network connections is not lame and case-insensitive search [2]. However, a misspelled keyword such as "alkhateb" has no hits while "alkhateeb" returns five documents. Additionally, the semantic relations are neither fully preserved nor well defined. In particular, one can pose the query "alkhateeb|euzenat" providing 79 documents while putting a space after the pipe "alkhateeb| euzenat" provides only 2 documents. The semantic reasoning is not provided (see Example 4). We avoided these limitations in the proposed methodology.

3.3 Test Case

[9] `http://www.w3.org/2007/08/pyRDFa/`

[10] `http://esw.w3.org/topic/SparqlImplementations`

[11] `http://jena.sourceforge.net/`

Suppose that the user had entered "faisal alkhateb" as an author, "jerome euzenat" as an another author, and "_sparql" as a title in the interface shown in Figure 3 and selected DBLP as a search database as well "or" and "and" connections between the authors and the title keywords. Then the query equation will be as: ((Author or Author)) and Title) = ((faisal alkhateeb or jerome euzenat) and _sparql).

The search will be done in Google to check if the author name exists in DBLP or not. In this testcase, the Google engine corrects the misspelled author name "faisal alkhateb" and uses "faisal alkhateeb" instead to connect to the DBLP with the correct name. Then the BibTeX elements corresponding to the keywords "faisal alkhateb", "jerome euzenat", and "sparql" are extracted from DBLP:

```
@article{DBLP:AlkhateebBE09,
 author    = {Faisal Alkhateeb and Jean-Francois
                Baget and Jerome Euzenat},
 title     = {Extending SPARQL with regular expre-
              ssion patterns (for querying RDF)},
 journal   = {J. web Sem.},
 volume    = {7},
 number    = {2},
 year      = {2009},
 pages     = {57-73},}

 ...
```

The BibTeX elements will be then converted to an RDF document such as the one in Example 2. Also, the corrected entered keywords will be used to build the following SPARQL query used to filter the results:

```
CONSTRUCT{?doc BibTeX:author "Faisal Alkhateeb"
?doc BibTeX:author "Jerome Euzenat"
...
}
FROM<RDF document corresponds to BibTeX>
WHERE{{{?doc BibTeX:author "Faisal Alkhateeb".
        ?doc BibTeX:title ?title.
        ?doc BibTeX:year ?year.
        ?doc BibTeX:pages ?pages. }
Union { ?doc BibTeX:author "Jerome Euzenat".
        ?doc BibTeX:title ?title.
        ?doc BibTeX:year ?year.
        ?doc BibTeX:pages ?pages. }}
      { ?doc BibTeX:title ?title }
        FILTER(regex(?title, "^sparql"))}
    }
```

Note that the keyword "_sparql" begins with underscore "_" and so it is considered to be part of the title while other keywords such "'faisal alkhateeb" do not and considered to be the full author names. Note that the user can specify a range for the publishing years. For instance, show me the authoring information between "2004" and "2008". In this case, she/he must can enter "2004-2008" in the year field, which in tern converted to the following part of a SPARQL query:

```
?document BibTeX:hasyear ?year .
FILTER ((?year >=2004) && (?year <= 2008))
```

4. RELATED WORK

The literature on combining the keyword search with the semantic search is rich; in this section we provide a brief overview of some relevant proposals.

Semantic web languages (i.e., RDF and OWL) can be used for knowledge encoding and can be used by services, tools, and applications [11]. The semantic web will not enable only human to process web contents, but also machines will be able to process web contents. This can help in creating intelligent services, customized web, and have more powerful search engines [9].

Traditional search engines use keywords as their search basis. Semantic search applies semantic processing on keywords for a better retrieval search. Hybrid search utilizes the keyword search from regular search along with the ability to use semantic search to query and reason using Meta data. Using ontologies the search engines can find pages that have different syntax, but similar semantics [9].

The hybrid search provided users with more capabilities for searching and reasoning to get better results. According to Bhagdev et al. [5] there are three types of queries that are possible using hybrid search:

- Semantic search using the defined Meta data and the relations between instances.
- Regular search using keywords.
- Search for keywords within specific contents.

Kiryakov et al. [14] proposed a system in which the user can select between keyword based search or ontology based search, but s/he cannot merge them to obtain search results using the two approaches together.

Another work by Bhagdev et al. [5] introduced a search method that combines ontology and keyword search based methods. The research results shows that the use of hybrid search gives a better performance over keyword search or semantic search in real world cases.

Rocha et al. [18] combined ontology based information retrieval with regular search in a semantic search technique. They used spread activation algorithm to get activation value of the relevance of search results with keywords. The links in the ontology are given weights according to certain properties. The proposed method do not identify promptly the unique concepts and relations.

In another work Gilardoni et al. [12] provided integration of keyword based search with ontology search, but with no capability for Boolean queries.

Hybrid search is implemented by some large companies in the industry. Google Product Search[12]is a semantic search service from Google which searches for products by linking between different attributes in the knowledge base to retrieve a product. Sheth et al. [19] use keyword query to apply multi-domain search by automatically classifying and extracting information along with ontology and meta data information.

Guha et al. [13] used a semantic search that uses an approach which combines traditional search and other data from distributed sources to answer the user query in more details. In the work of Davies et al. [8] QuizRDF is introduced. A system that combines the traditional search method with the ability to query and navigate RDF. The system shortcoming when there is a chaining in the query.

5. DISCUSSION

We have presented in this paper, an approach for searching and extracting authoring information. The approach is based on keyword and semantic search approaches. In the keyword search part, the entered keywords are used to collect authoring information. In this part, the Google search

[12] http://www.google.com/products

engine is used to correct the misspelled keywords, in particular the author's name, which allows to give more results. additionally, ad-hoc routines are used to extract bibliography elements from online databases. So, we suggest to include BibTeX elements in web pages as an RDFa annotations so that standard methods can be exploited. In the semantic part, the SPARQL query obtained from entered keywords is queried against the metadata corresponding to the authoring information, which allows to give more precise results.

6. REFERENCES

[1] Adida, B., and Birbeck, M. RDFa primer - bridging the human and data webs. Working draft, W3C, 2008. http://www.w3.org/TR/xhtml-rdfa-primer/.

[2] Bast, H., Mortensen, C. W., and Weber, I. Output-sensitive autocompletion search. *Inf. Retr. 11*, 4 (2008), 269–286.

[3] Beckett, D., and McBride, B. RDF/XML syntax specification (revised). Recommendation, W3C, 2004. http://www.w3.org/TR/rdf-syntax-grammar/.

[4] Berners-Lee, T., Hendler, J., and Lassila, O. The semantic web, 2001. http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21.

[5] Bhagdev, R., Chapman, S., Ciravegna, F., Lanfranchi, V., and Petrelli, D. Hybrid search: Effectively combining keywords and semantic searches. In *ESWC* (2008), pp. 554–568.

[6] Brickley, D., and Guha, R. RDF vocabulary description language 1.0: RDF schema. Recommendation, W3C, 2004. http://www.w3.org/TR/rdf-schema/.

[7] Bulterman, D., Grassel, G., Jansen, J., Koivisto, A., Layaïda, N., Michel, T., Mullender, S., and Zucker, D. Synchronized Multimedia Integration Language (SMIL 2.1). Recommendation, W3C, 2005. http://www.w3.org/TR/SMIL/.

[8] Davies, J., and Weeks, R. Quizrdf: Search technology for the semantic web. In *HICSS '04: Proceedings of the Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 4* (Washington, DC, USA, 2004), IEEE Computer Society, p. 40112.

[9] Decker, S., Melnik, S., van Harmelen, F., Fensel, D., Klein, M., Broekstra, J., Erdmann, M., and Horrocks, I. The semantic web: the roles of XML and RDF. 63–73.

[10] Fenn, J. Managing citations and your bibliography with BibTeX. *The Prac\TeX Journal 4* (2006). http://www.tug.org/pracjourn/2006-4/fenn/.

[11] Finin, T., and Ding, L. Search Engines for Semantic Web Knowledge. In *Proceedings of XTech 2006: Building Web 2.0* (May 2006).

[12] Gilardoni, L., Biasuzzi, C., Ferraro, M., Fonti, R., and Slavazza, P. Lkms - a legal knowledge management system exploiting semantic web technologies. In *International Semantic Web Conference* (2005), Y. Gil, E. Motta, V. R. Benjamins, and M. A. Musen, Eds., vol. 3729 of *Lecture Notes in Computer Science*, Springer, pp. 872–886.

[13] Guha, R., McCool, R., and Miller, E. Semantic search. In *WWW '03: Proceedings of the 12th international conference on World Wide Web* (New York, NY, USA, 2003), ACM, pp. 700–709.

[14] Kiryakov, A., Popov, B., Terziev, I., Manov, D., and Ognyanoff, D. Semantic annotation, indexing, and retrieval. *Web Semantics: Science, Services and Agents on the World Wide Web 2*, 1 (2004), 49 – 79.

[15] Manola, F., and Miller, E. RDF primer. Recommendation, W3C, 2004. http://www.w3.org/TR/rdf-primer/.

[16] Patashnik, O. Bibtexing, 1988. http://ftp.ntua.gr/mirror/ctan/biblio/bibtex/contrib/doc/btxdoc.pdf.

[17] Prud'hommeaux, E., and Seaborne, A. SPARQL query language for RDF. Recommendation, W3C, January 2008. http://www.w3.org/TR/rdf-sparql-query/.

[18] Rocha, C., Schwabe, D., and Aragao, M. P. A hybrid approach for searching in the semantic web. In *WWW '04: Proceedings of the 13th international conference on World Wide Web* (New York, NY, USA, 2004), ACM, pp. 374–383.

[19] Sheth, A., Bertram, C., Avant, D., Hammond, B., Kochut, K., and Warke, Y. Managing semantic content for the web. *IEEE Internet Computing 6*, 4 (2002), 80–87.