BioPatentMiner: An Information Retrieval System for BioMedical Patents

Sougata Mukherjea, Bhuvan Bamba

IBM India Research Lab
New Delhi, India
E-mail: {smukherj,bhuvanbh}@in.ibm.com

Abstract

Before undertaking new biomedical research, identifying concepts that have already been patented is essential. Traditional keyword based search on patent databases may not be sufficient to retrieve all the relevant information, especially for the biomedical domain. More sophisticated retrieval techniques are required. This paper presents BioPatentMiner, a system that facilitates information retrieval from biomedical patents. It integrates information from the patents with knowledge from biomedical ontologies to create a Semantic Web. Besides keyword search and queries linking the properties specified by one or more RDF triples, the system can discover Semantic Associations between the resources. The system also determines the importance of the resources to rank the results of a search and prevent information overload while determining the Semantic Associations.

1 Introduction

Before undertaking expensive and time consuming research for Drug Discovery, it is essential to determine what related biomedical concepts have already been patented. Online Patent databases exist for most countries that generally allow traditional keyword based search on various fields of a Patent (like Inventor, Assignee, Abstract, etc.) However, sometimes more complex retrieval techniques need to be supported. For example, a company may need to identify relationships with a competitor based on their assigned patents. For the biomedical domain there are additional complexities. Firstly, many biomedical concepts are known by a variety of names; therefore keyword based search on just a few of the synonyms may not retrieve all the relevant patents. Moreover, sometimes researchers may want to query on a class of biological terms; for example one may wish to retrieve all patents related to genes that have been issued to a competitor. Another complication is that sometimes Pharmaceutical companies patent a group of related molecules or an amino acid sequence. Therefore, discovering semantic relationships between biological concepts and patents, companies and inventors will be very useful. Because of these complexities most Pharmaceutical companies employ several Patent Analysts to manually examine hundreds of patents retrieved by querying the Patent databases.

In this paper we present **BioPatentMiner**, a system that facilitates information retrieval from biomedical patents. The system integrates information from the patents with biomedical ontologies and creates a *Biomedical Semantic Web*. Since the user information requirement will be varied, different views of the underlying information space are utilized. While for keyword based search, the traditional information retrieval model is useful, to answer queries linking the properties specified by one or more *RDF* triples, SQL-type declarative query languages are the most effective. On the other hand, to determine the semantic associations between Semantic Web resources, Graph algorithms are utilized. Since a real-world Biomedical Semantic Web will consist of thousands of resources we have also developed a technique to determine the importance of a resource in a Semantic Web. The importance is used to rank the results of a search and to filter the information space while determining the Semantic Associations between two resources.

The paper is organized as follows. Section 2

**Proceedings of the 30th VLDB Conference,
Toronto, Canada, 2004**

cites related work. Section 3 gives an overview of the system. Section 4 explains our method for determining the importance of the Semantic Web resources. Section 5 describes how we utilize the importance values to determine the Semantic associations between two resources. Section 6 presents some scenarios to show how BioPatentMiner can be used for information retrieval from a collection of biomedical patents. Finally, section 7 concludes the paper.

2 Related Work

2.1 Patent Retrieval Systems

Many countries provide Web interfaces for searching their patent databases (for example, the United States Patent and Trademark Office (USPTO)[22]). Research systems that utilize different techniques for retrieving information from Patent databases have also been developed. [15] introduces a system that integrates a series of shallow natural language processing techniques into a vector based document information retrieval system for searching a subset of US patents. On the other hand [13] uses a probabilistic information retrieval system for searching and classifying US patents. Another related system is described in [14] which tries to use techniques like Correspondence and Cluster analysis for mining patents. A report on a SIGIR Workshop on Patent Retrieval [9] highlights some of the challenges in the domain of Patent Retrieval.

In this paper we are focusing on Biomedical patents whose retrieval involves some unique challenges. An interesting system for querying Protein Patents is Kleisli [7]. Given a protein sequence, it uses Patent and Protein databases as well as Bioinformatics tools to identify whether similar protein sequences have already been patented. Some of these Bioinformatics tools can be utilized to augment our system as well.

2.2 Semantic Web Languages

BioPatentMiner creates a Semantic Web integrating the knowledge from patents and biomedical dictionaries. RDF [16] has become the standard language for representing any Semantic Web. It describes a Semantic Web using *Statements* which are *triples* of the form *(Subject, Property, Object)*. Subjects are *resources* which are uniquely identified by a *Uniform Resource Identifier (URI)*. Objects can be resources or literals. Properties are first class objects in the model that define binary relations between two resources or between a resource and a literal.

RDF Schema (RDFS) [17] makes the model more powerful by allowing new resources to be specializations of already defined resources. RDFS Classes are resources denoting a set of resources, by means of the property *RDF:type* (instances have property RDF:type valued by the class). All resources have by definition the property RDF:type valued by *RDF:Resource*. Moreover, all properties have RDF:type valued by *RDF:Property* and classes are of the type *RDFS:Class*.

Two important properties defined in RDFS are *subClassOf* and *subPropertyOf*. Two other important concepts are *domain* and *range* which apply to properties and must be valued by classes. They restrict the set of resources that may have a given property (the property's *domain*) and the set of valid values for a property (its *range*). A property may have as many values for *domain* as needed, but no more than one value for *range*. For a triple to be valid, the type of the object must be the range class and the type of the subject must be one of the domain classes.

RDFS allows inference of new triples based on several simple rules. Some of the important rules are:

1. $\forall s, p_1, o, p_2$ $(s, p_1, o) \wedge (p_1,$ RDFS:subPropertyOf, $p_2) => (s, p_2, o)$

2. $\forall r, c_1, c_2$ $(r,$RDF:type$,c_1)$ $\wedge$ $(c_1,$ RDFS:subClassOf, $c_2) => (r, RDF : type, c_2)$

3. $\forall c_1, c_2, c_3$ $(c_1,$RDFS:subClassOf$,c_2)$ $\wedge$ $(c_2,$ RDFS:subClassOf, $c_3)$ $=>$ $(c_1, RDFS : subClassOf, c_3)$

2.3 Building and Querying the Semantic Web

In recent times tools like Jena [8] have been developed to facilitate the development of Semantic Web applications. Researchers have also endeavored to represent existing knowledge bases in the Semantic Web languages. For example, [11] describes an effort to represent Unified Medical Language System (UMLS) [21] using Semantic Web languages.

The development of effective information retrieval techniques for the Semantic Web has become an important research problem. There are a number of proposed techniques for querying RDF data including RQL [10] and RDQL [18]. Most of these query languages use a SQL-like declarative syntax to query a Semantic Web as a set of RDF triples. They also incorporate inference as part of query answering. However, these languages are not able to determine complex relationships between two resources. For this purpose, [1] introduced the concept of **Semantic Associations** between Semantic Web resources. However no effective implementation of Semantic Associations was presented. We discuss our implementation of semantic associations in Section 5.

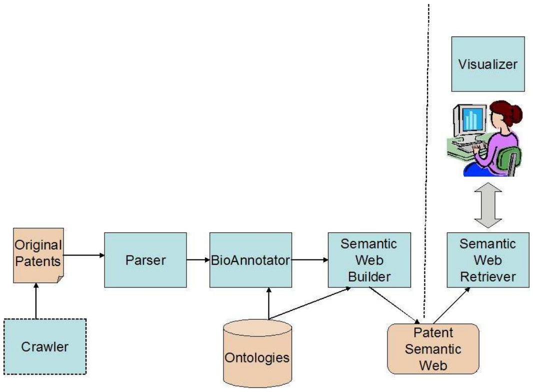


Figure 1: Architecture of BioPatentMiner

2.4 Determining WWW Page Importance

In this paper we introduce a technique to determine the importance of resources in a Semantic Web. This has been influenced by the extensive research in recent years to determine the importance of World-wide Web pages. The most well-known technique is *Page Rank* [4] which has been used very effectively to rank the results in Google Web search engine.

Another technique of finding the important pages in a WWW collection has been developed by Kleinberg [12] who defined two types of scores for Web pages which pertain to a certain topic: **authority** and **hub** scores. Documents with high Authority scores are authorities on a topic and therefore have many links pointing to them. On the other hand, documents with high hub scores are resource lists - they do not directly contain information about the topic, but rather point to many authoritative sites. Transitively, a document that points to many good authorities is an even better hub, and similarly a document pointed to by many good hubs is an even better authority. Kleinberg's algorithm has been refined in CLEVER [6] and Topic Distillation [2]. Both of these algorithms augment Kleinberg's link analysis with textual analysis. A good overview of various link analysis techniques to find hubs and authorities and suggestions for improvements are presented in [3].

3 BioPatentMiner: System Overview

BioPatentMiner is a system to facilitate knowledge discovery from patents related to Biomedicine. Figure 1 shows the overall architecture of the system. The system uses a crawler to download patents from the USPTO site [22] based on a query. The system can be also used on a collection of biomedical patents obtained by other techniques. The Parser parses these patents to extract information like Inventors, Assignees, Title, Abstracts etc. At present the parser assumes that the patents are in the HTML format of the USPTO site. It can be tuned for other formats.

The biological terms in the parsed files are then annotated by the **BioAnnotator** system [19]. BioAnnotator identifies and classifies biological terms in scientific text. It uses publicly available biomedical dictionaries like UMLS for this purpose. BioAnnotator also uses a Rule Engine to identify unknown and new biological terms. The annotated patents are represented in XML.

To facilitate knowledge discovery we want to integrate the information of the patents and biomedical ontologies. We believe that Semantic Web languages enable information from heterogeneous sources to be seamlessly integrated. Moreover one can utilize inference during querying. Therefore, **SemWebBuilder** is utilized to build a Semantic Web based on the annotated patents and biomedical dictionaries using RDF and RDFS. The patents, assignees and inventors of the patents as well as the Biomedical concepts identified by the BioAnnotator are represented as resources in the Semantic Web. Four properties link the resources:

- <*patentA* **refers_to** *patentB*> (patentA refers to patentB)
- <*inventorC* **invented** *patentD*> (inventorC has invented patentD)
- <*assigneeE* **assigned** *patentF*> (patentF is assigned to assigneeE)
- <*patentG* **has_term** *bioTermH*> (patentG has the biological concept bioTermH, as determined by BioAnnotator)

At present Unified Medical Language System (UMLS) [21] is used as the biomedical knowledge source. UMLS is a consolidated repository of medical terms and their relationships, spread across multiple languages and disciplines (chemistry, biology, etc). An essential section of UMLS is a **Semantic Network** which has 135 biomedical semantic classes like *Gene* *or Genome* and *Amino Acid, Peptide, or Protein.* The semantic classes are linked by a set of 54 semantic relationships (like *prevents*, *causes*). In addition there are biological concepts each of which are associated with one or more semantic classes. For example, the concept *blood cancer* has the semantic class *Neoplastic Process.* We created RDFS classes for all the Semantic Network classes and RDF Properties for all Semantic Network relationships except *isa*. A RDF statement is created to represent each relationship among the classes. The *isa* relationship is represented by *RDFS:subClassOf* relationship if it is between classes and *RDFS:subPropertyOf* relationship if it is between properties. The biological concepts are represented as RDF resources. They are named by their UMLS concept ids and the various names associated with the concept are stored as RDFS labels. The property *has_term* links the patents to the UMLS concepts they refer to.

SemWebRetriver is the run-time component of the system running inside a Web Application Server. It facilitates various types of information retrieval from the Semantic Web:

- SemWebRetriever supports keyword search on the annotated patents using the Juru XML search engine [5]. It facilitates retrieval of patents based on various criteria similar to USPTO.
- We develop Semantic Webs using Jena [8] which utilizes RDQL, a query language for RDF in Jena [18]. RDQL uses a declarative SQL-like syntax for querying information contained in one or more RDF triples. Although RDQL is data-oriented and does not support inference, Jena can create certain triples on-demand using inference. For example, if the triples *(c1 RDFS:subClassOf c2)* and *(r1 RDF:type c1)* are present, Jena can automatically infer that *(r1 RDF:type c2)* also exists.
- SemWebRetriever also identifies Semantic Associations between Web resources. This is discussed further in Section 5.

Visualizer is a client side Swing-based Java WebStart application. It allows the visualization of the Semantic Associations.

3.1 Graphical Representation of the Information Space

To fully capture the richness of a Semantic Web, a graphical representation of the information space is required. Let us define a Semantic Web as (C, P, NC) where C are the classes, P are the properties and NC are the normal resources (neither classes nor properties) that are defined for the Semantic Web. For creating the graphs we ignore classes and properties that are not defined in the local namespace (for example *RDF:Resource, RDFS:subClassOf*, etc.) We represent the information space using two graphs: isaGraph and propertyGraph.

3.1.1 isaGraph

The isaGraph is a directed graph whose vertices represent C, the classes of the Semantic Web. For all triples *(c1 RDFS:subClassOf c2)* defined in the Semantic Web, an edge $(c2, c1)$ is created in the isaGraph. Thus, the isaGraph represents the class hierarchy (*subClassOf* relation) of the Semantic Web. We ignore triples formed by inference while creating this graph. Note that the *subClassOf* relation cannot be represented as a tree, since a class can have more than one parent.

3.1.2 propertyGraph

Let P_r be a subset of P, containing only properties whose objects are resources. Let R be a subset of $(C \cup NC)$ satisfying the condition:
$\forall(r \in R)\exists(p_r \in P_r)$ such that r is a subject or object of a triple whose predicate is p_r or r is the domain or range of p_r.
The propertyGraph is a directed graph representing the properties defined in the local namespace. Its vertex set is R, the resources that are related to other resources by local properties. An edge from r_1 to r_2 exists in the propertyGraph if any one of the conditions hold:

- A triple (r_1, p_r, r_2) exists in the Semantic Web for any $(p_r \in P_r)$. In other words, an edge is created between two resources in the propertyGraph if they are the subject and object of a triple.
- $(p_r, \mathit{RDFS{:}domain}, r_1)$ and $(p_r, \mathit{RDFS{:}range}, r_2)$ exist in the Semantic Web for any $(p_r \in P_r)$. In other words, an edge is created between two resources (classes) in the property graph if they are the domain and range of a local property (and are thus related).

Note that we ignore triples formed by inference while creating this graph.

4 Semantic Web Resource Importance

SemWebRetriever queries can retrieve patents, assignees, inventors or biological concepts. In many cases many results will be retrieved and effective ways of ranking the results are required. Just ranking using information retrieval techniques like term frequency may not always provide the most intuitive results for the user. As Web search engines have shown, ranking based on the importance of the retrieved Web pages is very useful. Similarly, we can determine the importance of a resource in the Semantic Web to facilitate ranking. In this section we will discuss how we determine the importance of Semantic Web resources.

4.1 Subjectivity and Objectivity scores

A resource that has relationships with many other resources in the Semantic Web can be considered to be important since it is an important aspect of the overall semantics; the meaning of many other resources of the Semantic Web have to be defined with respect to that resource. In the context of the propertyGraph, vertices that have a high in-degree or out-degree should be considered important.

Kleinberg's hub and authority scores give a good indication about the connectivity of nodes in the WWW graph. It not only considers the number of links to and from a node but also the importance of the linked nodes. If a node is pointed to by a node with high hub score, its authority score is increased. Similarly, if a node points to a node with high authority score, its hub score is increased. Therefore, we calculate scores similar to the hub and authority scores of the propertyGraph to get an estimate of the importance of the resources in the Semantic Web. These scores are called **Subjectivity** and **Objectivity** scores corresponding to hub and authority scores. A node with high subjectivity/objectivity score is the subject/object of many RDF triples.

In the WWW all links are similar and can be considered to be equally important while calculating the hub and authority scores. On the other hand in a Semantic Web links in the propertyGraph represent properties; all the properties may not be equally important. For example, consider the property *has_term* in the Patent Semantic Web which links a Patent to the biological term it contains. The importance of the patent should not be dependent on the number of biological terms it contains. However, a biological term's importance should increase if it is referred to in many patents. On the other hand, consider the property *invented* in our Semantic Web which links an Inventor to a patent. The importance of a patent should not increase if it has many inventors. However, the importance of an inventor is obviously dependent on her patents. Therefore for each property we have a predefined subjectivity and objectivity weights which determine the importance of the subject/object of the property. By default these scores are 1.0. Properties like *has_term* will have a lower subjectivity weight while properties like *invented* will have a lower objectivity weight.

Kleinberg's algorithm has been modified to calculate the subjectivity and objectivity scores of Semantic Web resources as follows:

1. Let N be the set of nodes and E be the set of edges in the propertyGraph.
2. For every resource n in N, let $S[n]$ be its subjectivity score and $O[n]$ be its objectivity score
3. Initialize $S[n]$ and $O[n]$ to 1 for all r in R.
4. While the vectors S and O have not converged:
   (a) For all n in N, $O[n] = \sum_{(n1,n)\in E} S[n1] * objWt$ where $objWt$ is the objectivity weight of the property representing the link
   (b) For all n in N, $S[n] = \sum_{(n,n1)\in E} O[n1] * subWt$ where $subWt$ is the subjectivity weight of the property representing the link
   (c) Normalize the S and O vectors

Our modification is that while determining the subjectivity and objectivity scores of a vertex we multiply the scores of the adjacent vertex by the subjectivity/objectivity weights of the corresponding link. This will ensure that the scores of the resources are not influenced by unimportant properties. For example, a low objectivity weight for the *invented* property will ensure that the objectivity scores of patents are not increased by the number of inventors for that patent.

An important observation is that there is no "preferred direction" for a property. For example instead of the *invented* property we can have the *invented_by* property for which a patent is the subject and the inventor is the object. Thus, depending on the schema, a resource could equally well be a subject or an object. That is, the Subjectivity and Objectivity scores will be affected by the schema. However, the combined Subjectivity and Objectivity scores will be independent of the schema.

4.2 Determining Class Importance

The importance of a Semantic Web class is determined by how well it is connected to other classes. Obviously, this will be dependent on its subjectivity and objectivity scores. If c_1 is a subclass of c_2, all the properties of c_2 should be inherited by c_1. Therefore, the importance of a class should also be influenced by its parents. Because of the transitive property of the *subClassOf* relation, the importance of a class should actually be dependent on all its ancestors. However, we believe that a class should only marginally influence a distant descendent much lower in the *isa* hierarchy. Based on these beliefs, we calculate the importance of a class as:

1. Let $parentWt$, $subWt$, $objWt$ be predefined constants that determine the importance attached to the parents, subjectivity and objectivity scores while calculating the importance.
   $parentWt + subWt + objWt = 1.0$
2. If there are no links between class and non-class resources, filter the propertyGraph to include only

the classes and the links between them. (In other words, we remove all data resources and their related properties from the propertyGraph). If there are links between the schema and data resources the filtering is not necessary.

3. Calculate the Subjectivity and Objectivity scores of the classes from this graph.

4. Let C be the set of nodes and E be the set of edges in the isaGraph. (Obviously C contains the classes of the Semantic Web).

5. For every class c in C, let $S[c]$, $O[c]$, $PI[c]$ and $I[c]$ be its subjectivity, objectivity, parent importance and importance scores respectively.

6. $PI[c] = \frac{\sum_{(c1,c) \in E} I[c1]}{indegree(c)}$

7. $I[c] = PI[c] * parentWt + S[c] * subjWt + O[c] * objWt$

Thus, the importance of a class is determined by its subjectivity and objectivity scores and the importance of its parents. If (c_1, $subClassOf$, c_2) and (c_2, $subClassOf$, c_3), then $I(c_2)$ will be influenced by $I(c_3)$. Since $I(c_1)$ is influenced by $I(c_2)$, it is also influenced by $I(c_3)$. However, the influence of an ancestor on a node is inversely proportional to its distance from the node. It should be noted that we ignore RDF and RDFS vocabulary elements like RDF:Resource while calculating the Class Importance because we are only interested in the classes defined in the local namespace.

In many Semantic Webs, there will be no links connecting the schema (Class) and non-class (Data) resources. Thus there will be two separate subgraphs. If one of these subgraphs is more densely connected compared to the other subgraph, the importance scores of the vertices in the sparsely connected subgraph will be insignificant. To prevent this scenario, if there are no links between class and non-class resources, we filter non-class resources from the propertyGraph while calculating the Subjectivity and Objectivity scores of classes.

4.3 Determining Resource Importance

We believe that the importance of a Semantic Web non-class resource should be determined by how well it is connected to other resources. We also believe that it should be influenced by the importance of the classes it belongs to. Therefore we calculate the importance of a non-class resource as follows:

1. Let *classWt*, *subWt*, *objWt* be predefined constants that determine the importance attached to the classes, subjectivity and objectivity scores while calculating the importance.
   $classWt + subWt + objWt = 1.0$

2. If there are no links between class and non-class resources, filter the propertyGraph to only include the non-class resources in the Semantic Web and the links between them. (In other words, we remove all schema resources and their related properties from the propertyGraph).

3. Calculate the Subjectivity and Objectivity scores from this graph.

4. Let NC be the non-class resources in the Semantic Web. For every resource n in NC, let $S[n]$, $O[n]$, $CI[n]$ and $I[n]$ be its subjectivity, objectivity, class importance and importance scores respectively.

5. Let $noClass[n]$ be the number of triples in the Semantic Web where n is the subject and $RDF{:}type$ is the predicate.

6. $CI[n] = \frac{\sum_{(n,RDF:type,c) \in SemanticWeb} I[c]}{noClass[n]}$

7. $I[n] = CI[n] * classWt + S[n] * subWt + O[n] * objWt$

Thus the importance of a resource r is determined by its subjectivity and objectivity scores as well as the importance of all classes for which the triple (r, $RDF:type$, c) is defined explicitly in the Semantic Web. Note that the $subWt$ and $objWt$ constants for calculating the Class and Resource importance are different.

5 Semantic Associations

The RDF query languages like RDQL allow the discovery of all resources that are linked to a particular resource by an ordered set of specific relationships. For example, one can query a Semantic Web to find all resources that are linked to resource r_1 by the properties p_1 followed by p_2. Another option is to determine all the paths between resources r_1 and r_2 that are of length n. However, none of the query languages allow queries like "How are resources r_1 and r_2 related?" without any specification of the type of the properties or the length of the path. It is also not possible to determine relationships specified by undirected paths between two resources. In order to determine any arbitrary relationships among resources, Anyanwu and Sheth introduced the notion of **Semantic associations** based on ρ-queries [1]. In this section we will discuss an efficient implementation of Semantic Associations.

5.1 Definitions

Let us first give some definitions related to Semantic Associations based on the propertyGraph and the isaGraph. For the original definitions one should refer to [1]. For our definitions let Figure 2 represent a propertyGraph. Several resources are shown with the dashed

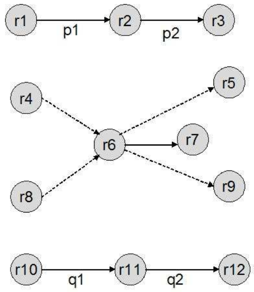


Figure 2: An Example propertyGraph

arrows representing paths between the resources and solid arrows representing edges between the resources.

- Two resources r_1 and r_2 are $\rho - path - associated$ if there is a direct path from r_1 to r_2 or r_2 to r_1 in propertyGraph. For example, in the example graph shown in Figure 2, resources $(r4, r9)$ and $(r5, r8)$ are $\rho - path - associated$.

- Two directed paths in the propertyGraph are said to be *joined* if they have at least one vertex common. The common vertex is the *join node*. For example, the directed paths from $r4$ to $r9$ and $r8$ to $r5$ are joined with the common vertex $r6$. Two resources r_1 and r_2 are $\rho - join - associated$ if there are joined paths p_1 and p_2 and either of these two conditions are satisfied:

  1. r_1 is the origin of p_1 and r_2 is the origin of p_2
  2. r_1 is the terminus of p_1 and r_2 is the terminus of p_2

  Thus in Figure 2 $(r4, r8)$ and $(r5, r9)$ are sets of $\rho - join - associated$ resources.

- Two resources r_1 and r_2 are $\rho - cp - associated$ if r_1 is of type c_1, r_2 is of type c_2 and either of these two conditions are satisfied:

  1. $c_1 = c_2$
  2. In the isaGraph there exists a class c_3 from which directed paths to both c_1 and c_2 exists.

  Thus resources are $\rho - cp - associated$ if they belong to the same class or classes which have a common ancestor. To prevent meaningless associations (like all resources belong to *RDF:Resource*), one can specify a *strong* $\rho - cp - associated$ relation which is true if either of these two conditions are also satisfied:

  1. The maximum path length from c_1 and c_2 to c_3 is below a threshold
  2. c_3 is a subclass of a set of user-specified general classes called the *ceiling*.

- Two directed paths of length n in the propertyGraph P and Q are isomorphic if:

  - They represent the properties $p_1, p_2, \ldots p_n$ and $q_1, q_2, \ldots q_n$ respectively; and
  - $\forall i, 1 \leq i \leq n, (p_i = q_i) \vee (p_i \subset q_i) \vee (q_i \subset p_i)$. Here $\subset$ represents the *subPropertyOf* relation.

  Two resources are $\rho - iso - associated$ if they are the origins of isomorphic paths. For example, in Figure 2 if $p1 \subset q1 \wedge p2 \subset q2$, $r1$ and $r10$ are $\rho - iso - associated$.

Two resources are said to be **semantically associated** if they are either $\rho - path - associated$ or $\rho - join - associated$ or $\rho - cp - associated$ or $\rho - iso - associated$.

5.2 Implementation

5.2.1 $\rho - path - associated$

To determine whether two resources are $\rho - path - associated$, a linear time algorithm can be used to determine whether there is a direct path between the two vertices in the propertyGraph. However, to be really useful, the user also needs to know how the two resources are related, that is, all the paths between the resources need to be determined. Just showing the shortest path may not be enough. Although fast algorithms exist for finding all paths between two vertices [20], for any real-world Semantic Web there will be a large number of paths between most resources. One solution suggested in [1] is to show paths whose length is less than some arbitrary number n. However, for a well connected propertyGraph, there may be a large number of such paths unless n is very small. While very small paths may not be very important, showing all sufficiently large paths may lead to information overload.

We have developed an algorithm that selectively shows the paths between the resources of interest based on the importance of the vertices in the path. The procedure $\rho\_path\_associated(r1, r2, N)$ determines at least the N most important paths between the resources $r1$ and $r2$ in the propertyGraph as follows:

- Let th be the current threshold and n be the number of paths found so far. Initialize th to a fairly large value less than one (≈ 0.5) and n to 0.
- while $(n < N)$ && $(th >= 0)$
  - Filter the property graph to include only $r1$ and $r2$ and resources whose importance is greater than th.
  - Determine the directed paths from $r1$ to $r2$ as well as $r2$ to $r1$ in the filtered graph.

    - Increment n by the number of paths found and decrement th by a small value (≈ 0.005)

The procedure can be initially called with a small value of N to identify the most important paths. If more paths are desired it can be subsequently called with a larger value of N. The procedure takes an optional fourth parameter, the initial threshold value; if a large number of paths are desired a smaller initial value of threshold can be specified. Thus the algorithm allows the user to see the important paths between two resources and still avoid information overload.

5.2.2 $\rho - join - associated$

The procedure $\rho\_join\_associated(r1, r2, N)$ determines the N most important join nodes forming join associations between the resources $r1$ and $r2$ in the propertyGraph as follows:

- Let th be the current threshold and n be the number of paths found so far. Initialize th to a fairly large value less than one (≈ 0.5) and n to 0.
- while ($n < N$) && ($th >= 0$)
    - Filter the property graph to include only $r1$ and $r2$ and resources whose importance is greater than th.
    - Let S_{end} be a set of all pairs of paths from $r1$ and $r2$ which have a common end vertex. Let vector C_{end} contain the common end vertices of these paths.
    - For every pair of paths in S_{end} check the paths from $r1$ to the end node and $r2$ to end node. If both the paths contain a vertex which already belongs to the vector C_{end} then this pair of paths does not lead to a join association and is eliminated from the set S_{end}. (This step will, for example, remove vertices $r5$, $r7$ and $r9$ in Figure 2 while determining the join association between $r4$ and $r8$).
    - Similarly, determine the set S_{start} that contains all pairs of paths to $r1$ and $r2$ from a common start vertex and the vector C_{start} containing the common start vertices of these paths.
    - Increment n by the join nodes found in C_{end} and C_{start}. Decrement th by a small value (≈ 0.005)

The procedure finds paths from/to $r1$ and $r2$ that end/start in a common (join) node. These paths represent the join associations.

5.2.3 $\rho - cp - associated$

The procedure $\rho\_cp\_associated(r1, r2, L, Ceiling)$ determines the $\rho\_cp\_associations$ between the resources $r1$ and $r2$. L and Ceiling are optional parameters to specify strong $\rho\_cp\_associations$. While L is the maximum permissible path length between the classes corresponding to the resources and the common ancestor, Ceiling specifies the most general set of classes that are to be considered. The procedure can be described as follows:

- Determine the set of classes $C1$ and $C2$ that the resources belong to. (If the resources are themselves classes, this step is not necessary).
- The ancestors of $C1$ and $C2$ can be determined from the Jena inference engine. We only consider ancestors that are subclasses of the set of classes specified by the Ceiling. Let the sets $C1_a$ and $C2_a$ contain the classes in $C1$ and $C2$ as well as their ancestors.
- Now a set of classes C_c that belong to both $C1_a$ and $C2_a$ is identified. We remove from C_c those classes whose children also belong to the set. If C_c is empty then $r1$ and $r2$ are not $\rho\_cp\_associated$.
- We check the paths from the common classes in C_c to the classes in $C1$ and $C2$ in the isaGraph. All paths of length less than L indicate the $\rho\_cp\_associations$ between $r1$ and $r2$. Note that since the number of edges in the isaGraph is quite small, there will not be many such paths.

5.2.4 $\rho - iso - associated$

Let us assume that two resources r_1 and r_2 have outgoing edges representing properties p_1 and p_2 respectively. If p_1 is the same as p_2 or is a $subPropertyOf\, p_2$ or vice versa, r_1 and r_2 are $\rho - iso - associated$ (with an isomorphic path of length one). Therefore, determining whether two resources are $\rho - iso - associated$ is trivial. However, determining the longest isomorphic path will require an exponential algorithm. Performance can be improved by applying it to a graph filtered by the importance scores.

5.2.5 Determining Path/Join Associations between a class and a non-class resource

The propertyGraph will generally not have many paths between a class and a non-class resource. This is because in most cases RDF triples are not created between schema and data resources except for triples of the form ($r1\ RDF : type\ c1$) specifying that a resource is of a particular class. Therefore for determining path or join associations between a class and a non-class resource the propertyGraph is not sufficient. There are two alternatives:

- Create a combined graph from propertyGraph and isaGraph containing all the vertices and edges of the graph as well as links from $r1$ to $c1$ for all triples of the form ($r1$ RDF:type $c1$).
- To determine an association between a class $c1$ and a non-class resource $r1$, besides finding paths between them (if any) in the propertyGraph, determine all resources of type $c1$ and find associations between these resources and $r1$. Inference should be utilized to find resources which are of a type which is a subclass of $c1$.

6 Experiments

A formal evaluation of the various techniques of BioPatentMiner is difficult since there is no standard corpus of biomedical patents available for testing. In this section we will present some scenarios where BioPatentMiner can be effectively used for information retrieval and knowledge discovery.

6.1 Experimental Collections

For our experiments we queried the USPTO site with the keyword *glycolysis*. We downloaded the 1346 patents retrieved by the query (in January 2004) and extracted relevant information about them. The title and abstracts of the patents were annotated by BioAnnotator. Then a Semantic Web was created from the patents (both the original 1346 patents and the patents they referred to), the assignees, the inventors and the UMLS biological terms in the patents. In total there were 7299 patents, 2852 inventors or assignees (some inventors are also assignees). The patents refer to 1291 UMLS concepts. The UMLS Semantic Network was also included in the Semantic Web.

6.2 Searching Annotated Documents

BioAnnotator annotates the patents with the baseform and the class of the identified biological terms. Baseform refers to the canonical form of the concept. For example, *caspase-3* has the baseform *CPP32 protein*. A biological concept can be referred to by various synonyms. For example, *caspase 3* is variously referred as *apopain*, *Yama protein*, *CPP32 protein*, etc. A consistent baseform tag allows the recognition of every reference to the biological concept even if it is called by different names. The class feature assigns each biological concept to its correct semantic class. For example, caspase-3 has the class *Amino Acid, Peptide or Protein*.

The annotated patents allow the retrieval of documents that would be missed by traditional keyword search. For example, a query on USPTO with *glycolysis* and *nucleic acid* in title or abstract only retrieved 29 patents. On the other hand, our system retrieved 196 patents for the query *nucleic acid* using the Juru search engine. This is because BioAnnotator identified several biological concepts that belong to the class *Nucleic Acid*. For example, unlike USPTO, BioPatentMiner retrieved the patent 6461611 since it contained *mRNA* which is a Nucleic Acid (UMLS concept C0035696).

6.3 Ranking Search Results

By default, the patents retrieved by a Juru search are ranked based on the date a patent was issued. However, sometimes ranking the patents by the importance of the patents is more useful. For example, if a company wants to determine the impact of its patents ranking by the importance is more appropriate. Similarly for RDQL queries to retrieve assignees, inventors or biological terms based on some criteria, ranking the results by the importance scores will be useful.

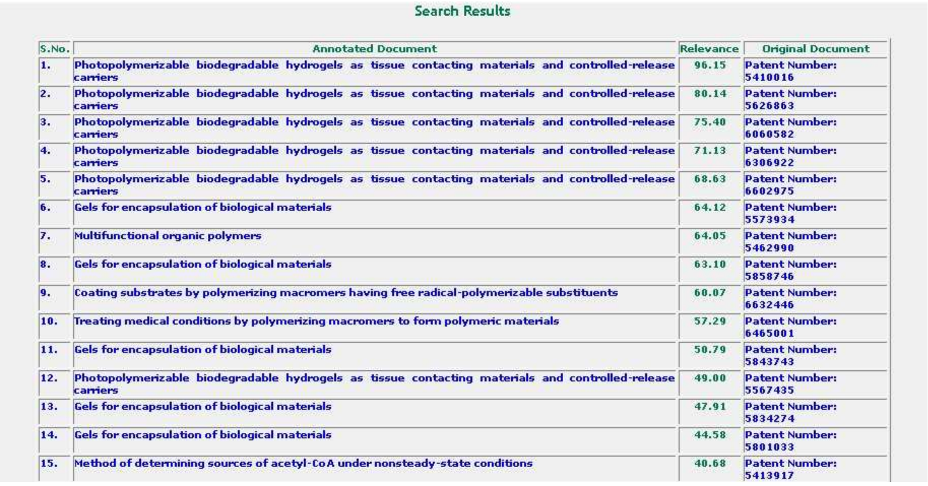
Figure 3 shows a search which retrieves all the patents issued to *University of Texas* ranked by the importance of the patents. The patent *5410016* is ranked the highest. This patent seems to have a high impact since it is referred to by 142 other patents. Similarly the second ranked patent is referred to by 36 other patents.

6.4 Semantic Associations

Sometimes a patent analyst will like to discover knowledge that is distributed across multiple patents. For example, a company or an inventor may like to find out all relationships with a competing company or all relationships with a class of biological concepts. Traditional retrieval techniques may not be adequate for the task. Semantic Associations may be useful for this purpose.

Figure 4 shows the Path Associations between inventor *Jeffrey A. Hubbell* and the UMLS class *Chemical*. Note that the Jena inference engine is utilized while determining all concepts of type *Chemical*. (For example, UMLS concept *C0017423* is of type *Biomedical or Dental Material* which is a subclass of *Chemical*). Determining this type of association from traditional retrieval techniques is very difficult.

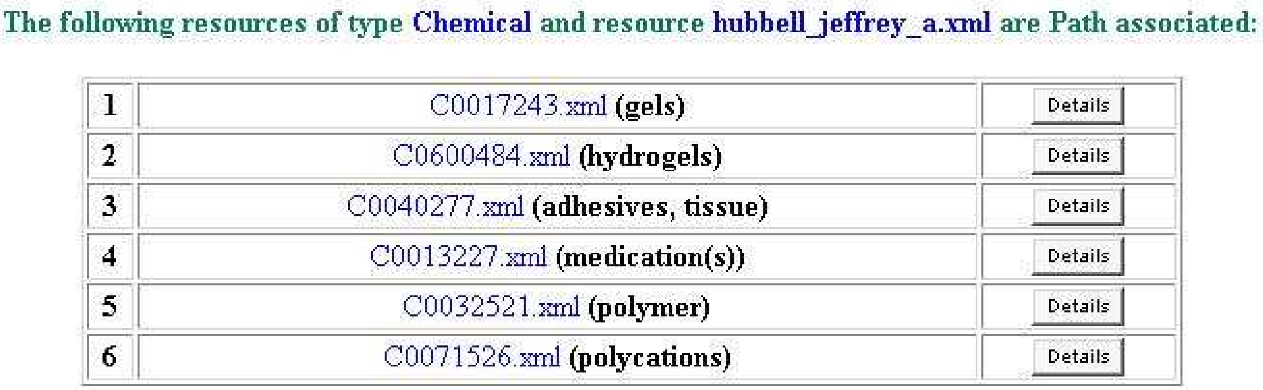
Figure 5 shows the Join Associations between two assignees *DSM Biotech GmbH* and *Purdue Research Foundation*. It shows that the Assignees are related based on several patents which are the join nodes. For example, *DSM Biotech GmbH* is assigned a patent 6316232 which refers to the patent 5168056 of *Purdue Research Foundation*. This kind of information may be useful for the companies for discovering potential patent infringements. Note that this technique of determining Semantic associations is



Search Results

S.No.	Annotated Document	Relevance	Original Document
1.	Photopolymerizable biodegradable hydrogels as tissue contacting materials and controlled-release carriers	96.15	Patent Number: 5410016
2.	Photopolymerizable biodegradable hydrogels as tissue contacting materials and controlled-release carriers	80.14	Patent Number: 5626863
3.	Photopolymerizable biodegradable hydrogels as tissue contacting materials and controlled-release carriers	75.40	Patent Number: 6060582
4.	Photopolymerizable biodegradable hydrogels as tissue contacting materials and controlled-release carriers	71.13	Patent Number: 6306922
5.	Photopolymerizable biodegradable hydrogels as tissue contacting materials and controlled-release carriers	68.63	Patent Number: 6602975
6.	Gels for encapsulation of biological materials	64.12	Patent Number: 5573934
7.	Multifunctional organic polymers	64.05	Patent Number: 5462990
8.	Gels for encapsulation of biological materials	63.10	Patent Number: 5858746
9.	Coating substrates by polymerizing macromers having free radical-polymerizable substituents	60.07	Patent Number: 6632446
10.	Treating medical conditions by polymerizing macromers to form polymeric materials	57.29	Patent Number: 6465001
11.	Gels for encapsulation of biological materials	50.79	Patent Number: 5843743
12.	Photopolymerizable biodegradable hydrogels as tissue contacting materials and controlled-release carriers	49.00	Patent Number: 5567435
13.	Gels for encapsulation of biological materials	47.91	Patent Number: 5834274
14.	Gels for encapsulation of biological materials	44.58	Patent Number: 5801033
15.	Method of determining sources of acetyl-CoA under nonsteady-state conditions	40.68	Patent Number: 5413917

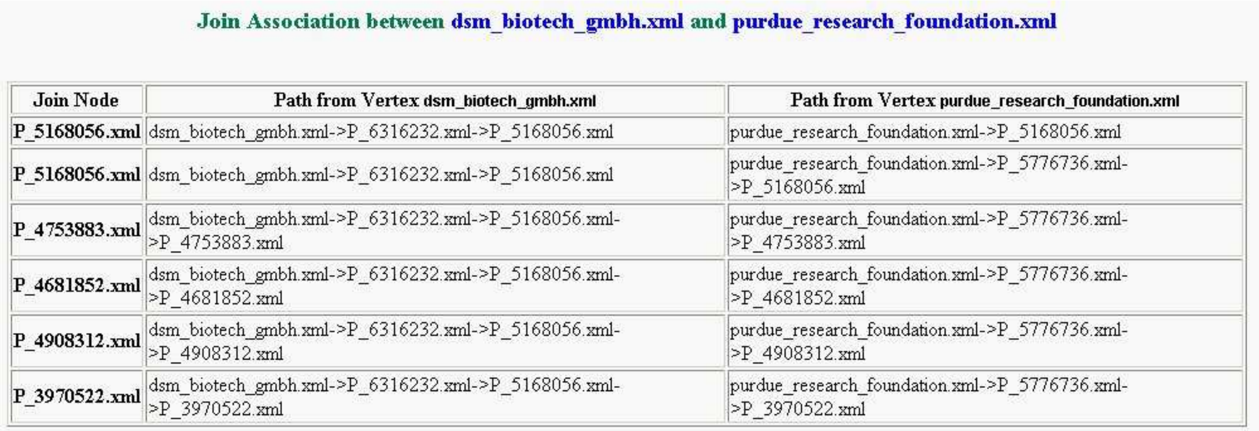
Figure 3: Results of a search with *University of Texas* as the *Assignee* ranked by the importance score



The following resources of type Chemical and resource hubbell_jeffrey_a.xml are Path associated:

1	C0017243.xml **(gels)**	Details
2	C0600484.xml **(hydrogels)**	Details
3	C0040277.xml **(adhesives, tissue)**	Details
4	C0013227.xml **(medication(s))**	Details
5	C0032521.xml **(polymer)**	Details
6	C0071526.xml **(polycations)**	Details

Figure 4: Path Associations between an inventor and a UMLS class



Join Association between dsm_biotech_gmbh.xml and purdue_research_foundation.xml

Join Node	Path from Vertex dsm_biotech_gmbh.xml	Path from Vertex purdue_research_foundation.xml
P_5168056.xml	dsm_biotech_gmbh.xml->P_6316232.xml->P_5168056.xml	purdue_research_foundation.xml->P_5168056.xml
P_5168056.xml	dsm_biotech_gmbh.xml->P_6316232.xml->P_5168056.xml	purdue_research_foundation.xml->P_5776736.xml->P_5168056.xml
P_4753883.xml	dsm_biotech_gmbh.xml->P_6316232.xml->P_5168056.xml->P_4753883.xml	purdue_research_foundation.xml->P_5776736.xml->P_4753883.xml
P_4681852.xml	dsm_biotech_gmbh.xml->P_6316232.xml->P_5168056.xml->P_4681852.xml	purdue_research_foundation.xml->P_5776736.xml->P_4681852.xml
P_4908312.xml	dsm_biotech_gmbh.xml->P_6316232.xml->P_5168056.xml->P_4908312.xml	purdue_research_foundation.xml->P_5776736.xml->P_4908312.xml
P_3970522.xml	dsm_biotech_gmbh.xml->P_6316232.xml->P_5168056.xml->P_3970522.xml	purdue_research_foundation.xml->P_5776736.xml->P_3970522.xml

Figure 5: Join Associations between two assignees

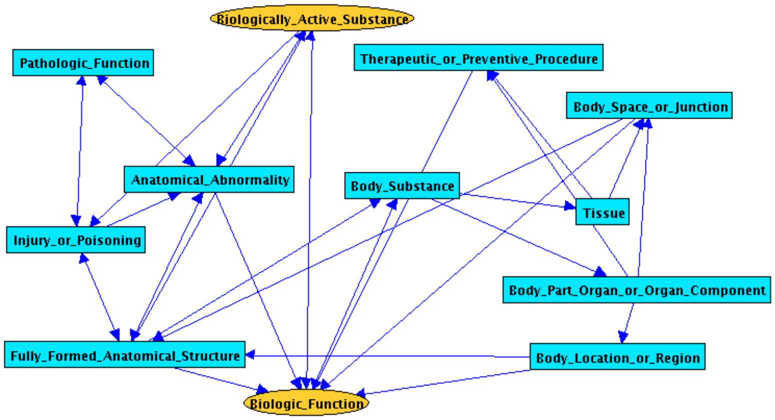


Figure 6: Visualizing the Path Associations between two UMLS Classes

useful for all classes of patents and not restricted to the Biomedical domain.

Besides determining associations between patents, inventors, assignees and UMLS concepts and classes, one can also identify associations within UMLS Semantic Network classes. For example, Table 1 shows the number of paths of different length identified between the resources representing UMLS classes *Biologically_Active_Substance* and *Biologic_Function* in the Semantic Web for different values of threshold. There are 124 paths of length $\leq$ 5 between the resources *Biologically_Active_Substance* and *Biologic_Function*. Showing all these paths will result in an information overload. Filtering the graph to show the most important paths will be more useful. For example at a threshold of 0.05 there are only 20 paths.

Showing the Semantic Associations textually may not be very intuitive for the users if many paths are retrieved. Therefore one can visualize the different types of associations between Semantic Web resources. For example, Figure 6 is a visualization that shows the $\rho-path-associated$ directed paths of length $\leq$ 5 between *Biologically_Active_Substance* and *Biologic_Function* for a threshold of 0.05. Note that to prevent clutter, the labels of the edges are only shown by clicking on them. The interface allows the user to change the value of threshold to see a different number of paths.

7 Conclusion

This paper introduced BioPatentMiner, a system that facilitates information retrieval for biomedical patents. The system identifies and classifies the biologically significant terms in the patents and integrates them with concepts in biomedical dictionaries to create a Semantic Web. The system incorporates a technique to calculate the importance of Semantic Web resources that can be used to rank the results of a query. We have also presented a method to determine the Semantic Associations between resources based on the importance of the resources. Some scenarios have been presented to show the usefulness of the system. Future work is planned along various directions:

Path Length	**Threshold**				
	0.0	0.005	0.01	0.03	0.05
1	2	2	2	2	2
2	3	3	3	3	2
3	6	6	6	4	3
4	20	20	20	12	3
5	93	91	87	68	10

Table 1: Number of paths of different lengths for different values of threshold between *Biologically_Active_Substance* and *Biologic_Function*

- We plan to conduct user studies with domain experts to validate the effectiveness of our techniques to facilitate information retrieval for biomedical patents. We are collaborating with a Pharmaceutical company for this purpose.

- In addition to refining our procedures for determining resource importance and Semantic Associations, we are trying to discover whether other techniques of information retrieval are useful for the Semantic Web. Scalability of the techniques should also be evaluated and improved.

- There are various sources of biomedical knowledge like patents, dictionaries and ontologies. Since it is difficult for researchers to easily gain understanding of a biomedical concept from these different knowledge sources, we believe that a Biomedical Semantic Web is essential. Our vision is that distributed Web servers would store the "meaning" of biological concepts and sets of inference rules will be stored in biomedical ontologies to enable automated reasoning on the concepts. This will enable researchers to perform a single semantic search to retrieve all the relevant information about a biological concept.

References

[1] K. Anyanwu and A. Sheth. ρ-Queries: Enabling Querying for Semantic Associations on the Semantic Web. In *Proceedings of the Twelfth International World-Wide Web Conference*, Budapest, Hungary, May 2003.

[2] K. Bharat and M. Henzinger. Improved Algorithms for Topic Distillation in a Hyperlinked Environment. In *Proceedings of the ACM SIGIR '98 Conference on Research and Development in Information Retrieval*, pages 104–111, Melbourne, Australia, August 1998.

[3] A. Borodin, G. Roberts, J. Rosenthal, and P. Tsaparas. Finding Authorities and Hubs from Link Structures on the World Wide Web. In *Proceedings of the Tenth International World-Wide Web Conference*, pages 415–429, Hong Kong, May 2001.

[4] S. Brin and L. Page. The Anatomy of a Large-scale Hypertextual Web Search Engine. *Computer Networks and ISDN Systems. Special Issue on the Seventh International World-Wide Web Conference, Brisbane, Australia*, 30(1-7):107–117, April 1998.

[5] D. Carmel, E. Amitay, M. Hersovici, Y. Maarek, Y. Petruschka, and A. Soffer. Juru at TREC-10: Experiments with Index Pruning. In *the Proceedings of the 10th Text Retrieval Conference*, pages 228–237, 2001.

[6] S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, P. Raghavan, and S. Rajagopalan. Automatic Resource Compilation by Analyzing Hyperlink Structure and Associated Text. *Computer Networks and ISDN Systems. Special Issue on the Seventh International World-Wide Web Conference, Brisbane, Australia*, 30(1-7):65–74, April 1998.

[7] J. Chen, L. Wong, and L. Zhang. A Protein Patent Query System Powered by Klesili. In *the Proceedings of the ACM SIGMOD Conference*, Seattle, WA, 1998.

[8] JENA. http://www.hpl.hp.com/semweb/jena2.htm.

[9] N. Kando and M. Leong. Workshop on Patent Retrieval: SIGIR 2000 Workshop Report. *ACM SIGIR Forum*, 34(1):28–30, April 2000.

[10] S. Karvounarakis, S. Alexaki, V. Christophides, D. Plexousakis, and M. Scholl. RQL: A Declarative Query Language for RDF. In *Proceedings of the Eleventh International World-Wide Web Conference*, Honolulu, Hawaii, May 2002.

[11] V. Kashyap and A. Borgida. Representing the UMLS Semantic Network using OWL (Or "Whats in a Semantic Web Link?"). In *the Proceedings of the Second International Semantic Web Conference*, Sanibel Island, Florida, 2003.

[12] J. Kleinberg. Authorative Sources in a Hyperlinked Environment. In *Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms*, May 1998.

[13] L. Larkey. A Patent Search and Classification System. In *the Proceedings of the ACM Digital Library Conference*, Berkeley, CA, 1999.

[14] M. Marinescu, M. Markellou, G. Mayritsakis, K. Perdikuri, S. Sirmakessis, and A. Tsakalidis. Knowledge Discovery in Patent Databases. In *the Proceedings of the ACM Conference on Information and Knowledge Management*, McLean, Virginia, 2002.

[15] M. Osborn, T. Strzalknowski, and M. Marinescu. Evaluating Document Retrieval in Patent Database: a Preliminary Report. In *the Proceedings of the ACM Conference on Information and Knowledge Management*, Las Vegas, Nevada, 1997.

[16] Resource Description Format. http://www.w3.org/1999/02/22-rdf-syntax-ns.

[17] Resource Description Format Schema. http://www.w3.org/2000/01/rdf-schema.

[18] A. Seaborne. RDQL: A Data Oriented Query Language for RDF Models. http://www.hpl.hp.com/semweb/rdql-grammar.html.

[19] L. Subramaniam, S. Mukherjea, P. Kankar, B. Srivastava, V. Batra, P. Kamesam, and R. Kothari. Information Extraction from Biomedical Literature: Methodology, Evaluation and an Application. In *the Proceedings of the ACM Conference on Information and Knowledge Management*, New Orleans, Lousiana, 2003.

[20] R. Tarjan. Fast Algorithms for Solving Path Problems. *Journal of ACM*, 28(3), July 1991.

[21] UMLS. http://umlsks.nlm.nih.gov.

[22] United States Patent and Trademark Office. http://www.uspto.gov/patft/.