Information Retrieval and Knowledge Discovery Utilizing a BioMedical Patent Semantic Web

Sougata Mukherjea, Bhuvan Bamba, and Pankaj Kankar

Abstract—Before undertaking new biomedical research, identifying concepts that have already been patented is essential. A traditional keyword-based search on patent databases may not be sufficient to retrieve all the relevant information, especially for the biomedical domain. This paper presents BioPatentMiner, a system that facilitates information retrieval and knowledge discovery from biomedical patents. The system first identifies biological terms and relations from the patents and then integrates the 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 Web 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.

Index Terms—Biomedical information retrieval, Semantic Web, information extraction.

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. First, 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 or an interaction between biological entities. 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 first identifies biological terms and relations in the patents and then integrates information from these 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. The system can also summarize a collection of patents based on the biological terms associated with them. 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. BioPatentMiner also enables the user to visualize the relationships in the Semantic Web in various ways.

Fig. 1 shows the overall architecture of the system. The system uses a crawler for downloading patents and a parser for parsing these patents. The *Annotators* identify the biological terms and relations in the parsed patents and a Semantic Web is built from the annotated documents. The process of building the Semantic Web is discussed in detail in Section 3. To facilitate information retrieval, we have developed a method for determining the importance of the Semantic Web resources. This is explained in Section 4.

Semantic Web Retriever incorporates several techniques of information retrieval from the Semantic Web. The *Visualizer* is a client side Swing-based Java WebStart application for visualizing the relationship between patents in various ways. We will describe these components of BioPatentMiner and present some retrieval scenarios to show how BioPatentMiner can be used for information retrieval and knowledge discovery on a collection of biomedical patents in Sections 5 and 6. Finally, Section 7 will conclude the paper.

• *The authors are with the IBM India Research Lab, Block I, IIT, New Delhi 110016, India. E-mail: {smukherj, bhuvanbh, kpankaj}@in.ibm.com.*

Manuscript received 21 July 2004; revised 7 Oct. 2004; accepted 10 Feb. 2005; published online 17 June 2005.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number TKDESI-0253-0704.

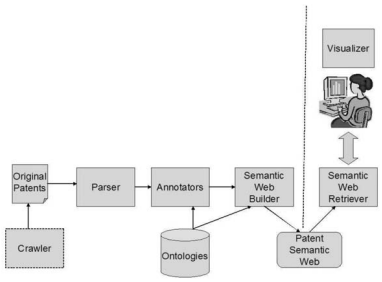

Fig. 1. Architecture of BioPatentMiner.

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) [23]). Research systems that utilize different techniques for retrieving information from Patent databases have also been developed. For example, [13] 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, [10] uses a probabilistic information retrieval system for searching and classifying US patents. Another related system is described in [12] which tries to use techniques like correspondence and cluster analysis for mining patents. A report on a SIGIR Workshop on Patent Retrieval [6] highlights some of the challenges in the domain of patent retrieval.

In this paper, we focus on biomedical patents whose retrieval involves some unique challenges. An interesting system for querying protein patents is Kleisli [4]. 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 Determining Relationships between Biological Entities from Scientific Documents

To facilitate knowledge discovery, BioPatentMiner identifies biological terms and relations in the patents. Automatic extraction of useful information from online biomedical literature is a challenging problem because these documents are expressed in a natural language form. The first task is to recognize and classify the biological entities in the scientific text. We have developed the BioAnnotator system [19] for this purpose. It uses rules and dictionary lookup for identifying and classifying biological terms.

After the biological entities are recognized, the next task is to identify the relations between these entities. To determine these relations (for example, protein-protein interactions), one approach is to use templates that match specific linguistic structures [24]. Natural language processing techniques that use parsers of increasing sophistication have also been utilized [14]. While no extensive validation results are available on these systems, their specificity was estimated at 60-80 percent. Recently, research has gone beyond treatment of single sentences to look at relations that span multiple sentences through the use of coreference [15].

We have used a template filling approach based on a natural language parser to identify relations between the biological entities discovered by the BioAnnotator system. Our system is customizable allowing the user to identify various types of interactions.

2.3 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*. 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 also allows inference of new triples based on several simple rules.

2.4 Building and Querying the Semantic Web

In recent times, tools like Jena [5] have been developed to facilitate the development of Semantic Web applications. 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 [8] 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.

2.5 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* [2] which has been

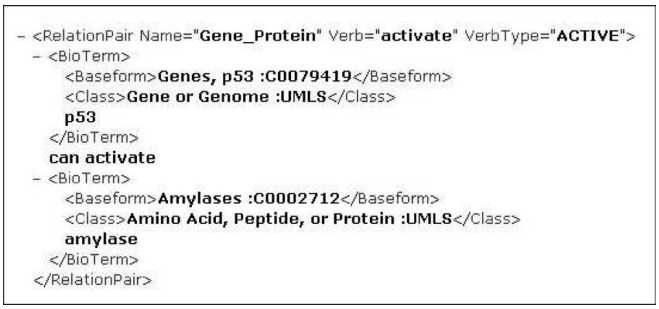

Fig. 2. Annotations for the sentence "*p53 is activated by amylase*."

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 [9] 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.

3 Building the Biomedical Patent Semantic Web

BioPatentMiner is a system to facilitate knowledge discovery from patents related to Biomedicine. In this section, we will explain how we annotate the patents and build a Semantic Web. The system uses a crawler to download patents from the USPTO site [23] 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, Abstract, 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.

3.1 Annotators

The biological terms in the parsed files are first annotated by the **BioAnnotator** system [19]. BioAnnotator identifies and classifies biological terms in scientific text using publicly available biomedical dictionaries and a Rule Engine. At present, Unified Medical Language System (UMLS) [22] 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. The biological concepts can be known by various synonyms one of which is identified by UMLS as the primary name or baseform. BioAnnotator annotates the patents with the baseform and the class of the identified biological terms.

TABLE 1
Precision and Recall of the Relation Annotator

	Precision	Recall
Approx	0.9194	0.8028
Exact	0.613	0.5352

Determining relations between entities from natural language text is a very difficult problem because the relations may be expressed in various forms. In order to simplify the problem, we assume that we would be only trying to identify certain types of relations between certain classes of biological entities. The user specifies these in a XML configuration file.

The **RelationAnnotator** examines documents that have already been annotated by a natural language parser and BioAnnotator. It identifies biological entities specified in the configuration file by matching the classes of the identified entities with the specified regular expressions. It then tries to identify relations between these entities based on templates for patterns which specify relationships in sentences. For example, some common patterns are:

1. *Subject Verb_Group Object* (for example, *p53 activates amylase*).
2. *Object Passive_Verb_Group Subject* (for example, *amylase is activated by p53*).
3. *Noun (Nominal form of verb) Object Subject* (for example, *activation of amylase by p53*).

If a template is matched it is assumed that a relation of the matching verb group (or nominal form) has been identified. Note that only biological entities and verbs in the sentences are considered during template matching. The annotated patents are represented in XML. For example, the annotation for the sentence "*p53 can activate amylase*" is shown in Fig. 2.

3.1.1 Evaluation

Evaluation of the BioAnnotator is presented in [19]. Formal evaluation of the Relation Annotator is difficult because there is no standard corpus. Therefore, for evaluation, we manually examined 88 patents retrieved by a query "*protein protein interactions*" from the USPTO. One hundred twenty-four relations between proteins were identified in the abstracts of these patents. The results of evaluation of Relation Annotator using this corpus are shown in Table 1. When an automatically identified relation is matched with a manually annotated relation, one can look for exact or approximate match. For exact match, the annotations should match exactly. For approximate match, one of the annotations should be a substring of the other. The table shows that for approximate match, the precision and recall are 92 percent and 80 percent, respectively, while for exact match they are 61 percent and 54 percent, respectively. The precision and recall are affected by a variety of reasons. On closer analysis of the results, we found that:

1. Usually, it is very hard to get an agreement even between two experts about the extent of a biological

term. For example, for the phrase "*human cancer tissue*," one expert may consider the whole phrase to be a biological term while another may only mark *cancer tissue* to be the biological term. Because of the differences in the extent of biological terms between the automatic and manual, our exact match precision and recall are much lower than the approximate match scores.

2. More templates are required to identify all types of relations. For example, at present we could not identify the relation "*transcriptional regulatory proteins activated in response to various cytokines*" because the sentence does not match any of our current templates for relations.
3. We also need to refine our current templates to prevent false positives like "*L-Selectin and also binds to P-Selectin*" where RelationAnnotator incorrectly identified *L-Selectin* as the subject of the relation.
4. We cannot handle complex sentences that require Anaphora Resolution. Quite often, anaphors like *it, they* or anaphoric noun phrases like *the protein* are the actual arguments to a relation.
5. Errors of the underlying shallow parser and BioAnnotator also affect the precision and recall. For example, for the sentence "*p45 associates with p19*," BioAnnotator failed to recognize *p45* and *p19* as proteins.

3.2 Semantic Web Builder

To facilitate knowledge discovery we want to integrate information from 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, Semantic Web Builder is utilized to build a Semantic Web based on the annotated patents and biomedical dictionaries using RDF and RDFS. The user can either create one Semantic Web containing all the patents of interest or several Semantic Webs each containing a collection of related patents. 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:

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

One problem is that the patents can refer to the same assignee by different names. For example, in one patent, a company can be named *A inc.* while in another it may be called *A corporation*. To ensure that we create only one Semantic Web resource for each assignee, we replace punctuations from the Assignee names with spaces, tokenize based on spaces and remove tokens like *inc* and *corporation*. The final assignee name is determined from the remaining tokens.

We also integrate UMLS into the Semantic Web. 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 synonyms associated with the concept are stored as RDFS labels. Each biological concept is linked to the various Semantic Network classes that it belongs to by the *RDF:type* property. The property *has_term* links the patents to the UMLS concepts they refer to.

For each type of relation (for example, *activate, inhibit*, etc.) that is identified by the Relation Annotator, we create a RDF property (unless the property corresponds to a Semantic Network property). Now, suppose the Relation Annotator has identified a relation of type *activate* in Patent P_i. We create a new property, for example, *activate_j_i*, and add the following triples to the Semantic Web:

1. $<activate\_j\_i\ subPropertyOf\ activate>$.
2. $<C\_1\ activate\_j\_i\ C\_2>$, where C_1 and C_2 are the Biological entities which are the subject and object of the identified relation. Note that for some relation (for example, "*interaction between p53 and p45*"), the subject and object cannot be determined. In these cases, in the RDF triple, we choose the first concept as the subject.
3. $<P\_i\ supports\ activate\_j\_i>$, where *supports* is a property that indicates that a patent has evidence about a relation.

3.3 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.

3.3.1 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.

3.3.2 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 holds:

1. 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.
2. $(p_r, RDFS : domain, r_1)$ and $(p_r, RDFS : range, r_2)$ 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

In this section, we will discuss how we determine the importance of Semantic Web resources. Two different methods of calculating the importance are presented; *Subjectivity and Objectivity scores* based on hub and authority scores [9] and *Resource Rank* based on Page Rank [2].

4.1 Subjectivity Score (SS) and Objectivity Score (OS)

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 score (SS)** and **Objectivity score (OS)** 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 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. The subjectivity and objectivity weights will vary from one Semantic Web to another and needs to be determined by experimentation.

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 $SS[n]$ be its subjectivity score and $OS[n]$ be its objectivity score
3. Initialize $SS[n]$ and $OS[n]$ to 1 for all r in R.
4. While the vectors SS and OS have not converged:
   a. For all n in N, $OS[n] = \sum_{(n1,n)\in E} SS[n1] * objWt$, where $objWt$ is the objectivity weight of the property representing the link.
   b. For all n in N, $SS[n] = \sum_{(n,n1)\in E} OS[n1] * subWt$, where $subWt$ is the subjectivity weight of the property representing the link.
   c. Normalize the SS and OS 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 Resource Rank (RR)

Instead of calculating the Subjectivity and Objectivity scores we can calculate a single score called the **Resource Rank (RR)**. The Resource Rank $RR(n)$ of a resource can be calculated as:

TABLE 2
Importance Scores for Some Resources in the UMLS Semantic Web

Resource	RR	SS	OS
Injury or Poisoning	0.694	0.391	1.0
Pharmacologic Substance	0.41	1.0	0.081
Body Space or Junction	1.0	0.854	0.31

$$RR[n] = \frac{(1-d)}{N} + d\Bigg[\sum_{(n1,n)\in E} RR[n1] * objWt(e1) + \sum_{(n,n2)\in E} RR[n2] * subWt(e2)\Bigg],$$

where $e1$ is the link from $n1$ to n and $e2$ is the link from n to $n2$. N is the total number of resources in the Semantic Web.

The Resource Rank is similar to PageRank [2]. The parameter d is a damping factor which can be set between 0 and 1. PageRank is based on modeling Web surfing as a Random Walk in the Web graph where at each step the user goes to one of the neighboring pages with a probability d and jumps to a random page with a probability $1-d$. The page rank of a Web page corresponds to the probability of visiting the page during the random walk.

Resource Rank can be calculated using a simple iterative algorithm with the ranks for all resources initialized to 1. The damping factor ensures that the algorithm will converge even if the graph is not strongly connected. (Obviously, in many cases, the propertyGraph may not be strongly connected). Another criterion for convergence is that the graph should be *aperiodic*. We can assume that like the WWW graph, propertyGraphs for most Semantic Webs will be aperiodic.

We have made three changes for calculating the Resource Rank:

1. Unlike the WWW, a resource's rank should be calculated based on both the incoming and outgoing links.
2. Unlike the WWW, each link will not be equally important; the links will have a subjectivity and objectivity weight, as described before, which will be used during calculation of the rank.
3. For PageRank calculation, the importance of an adjacent vertex that has a link to the current vertex is divided by the outdegree of the adjacent vertex. We believe that this is not appropriate for the Semantic Web resources since their importance will be dependent on both their indegree and outdegree. For example, consider a triple $< r1, p, r2 >$. If $r1$ is important, we believe that $r2$ should also be important. The importance of $r1$ should increase if it has many outlinks; therefore, dividing the importance of $r1$ by its outdegree while calculating the importance of $r2$ does not seem to be correct.

4.3 Evaluation

To validate our technique of determining the importance of Semantic Web resources, we have developed several example Semantic Webs. In this section, we will discuss our experiments based on these Webs.

TABLE 3
Importance Scores for Some Resources in the *Cephalosporin* Patent Semantic Web

Resource	RR	SS	OS
C0007732	1.0	0.0	1.0
P_4278793	0.487	0.001	0.479
Glaxo	0.506	0.952	0.0
Merck	0.480	0.592	0.0

4.3.1 UMLS Semantic Network

Table 2 shows the importance scores for several UMLS Semantic Network classes. For the calculation of importance, all UMLS properties had the default weight of 1.0. The resource representing the Semantic Network class *Injury or Poisoning* has the maximum objectivity score while *Pharmacologic Substance* has the maximum subjectivity score. The class *Body Space or Junction* has the maximum Resource Rank. Feedback from a domain expert very familiar with UMLS indicates that we were able to identify some of the important UMLS classes by our technique.

4.3.2 Patent Semantic Web

Let us consider a Biomedical Patent Semantic Web created by querying the USPTO site to retrieve patents that have *Cephalosporin* (an antibiotic) in the title or abstract. In total, there were 5,833 resources in the Semantic Web consisting of 1,071 patents that were retrieved by the query, the patents they referred to as well as the inventors, assignees and biological terms of these patents. Table 3 shows the importance scores for several resources in the Semantic Web. For calculation of the importance, we reduced the subjectivity weight of properties like *refers_to* and *has_term* and the objectivity weights of the properties like *invented* and *assigned*. The UMLS concept *C0007732* has the highest Resource Rank and Objectivity score. This is not surprising since this concept represents Cephalosporin. The Patent 4278793 titled "*Cephem derivative*" has the highest Objectivity score and Resource Rank. This patent is referred to by 120 patents and, thus, its high rank is justified. Well-known pharmaceutical companies like *Glaxo* and *Merck* have high Subjectivity score as well as Resource Rank. These companies have been assigned several important patents. It seems that for this Semantic Web also we could identify important resources using our techniques.

4.3.3 TAP Organization Semantic Web

The TAP project [20] has developed a Semantic Web covering various domains like Athletes, Musicians, Organizations, etc., by utilizing HTML scrapers on some popular Web sites. We also applied our technique to the TAP Organization Semantic Web. Table 4 shows the importance scores for several resources in this Semantic Web. All properties were assigned the default scores of 1.0. The resource *CMU CSDepartment* has the maximum objectivity score as well as Resource Rank while a graduate student of CMU has the maximum subjectivity score. On the other

TABLE 4
Importance Scores for Some Resources in the TAP Organization Semantic Web

Resource	RR	SS	OS
CMU CSDepartment	1.0	0.0	1.0
CMUGrad H AnCheng	0.074	1.0	0.0
Apple Computer	0.0	0.0	0.0

TABLE 5
Properties of the propertyGraph for the Example Semantic Webs

	Vertex	Edge	Edge/Vertex
UMLS	92	351	3.82
Patents	5833	27450	4.71
TAP	6225	6509	1.05

hand, the resource for *Apple Computers* has zero values for all scores! This does not seem to be correct since the importance of a well-known company should be quite high.

4.4 Discussion

While our technique was able to identify important resources of Semantic Webs for UMLS Semantic Network and US Patents related to *Cephalosporin*, it did not give good results for the Organization Semantic Web developed by the TAP project. We believe that our technique for determining the importance will only be useful for Semantic Webs that represent most of the knowledge related to a domain. For the TAP Semantic Web while some organizations have most of their knowledge encoded, for others important information is missing. For example, all faculty and graduate students of CMU are represented. On the other hand, for *Apple Computers*, practically no information is encoded. It is true that one of the main assumptions of the Semantic Web (in comparison to earlier AI Knowledge Representation systems) is that it assumes an open world; all knowledge may not be represented in the Web. However, unless the Semantic Web is reasonably complete, our algorithms will not be really appropriate.

Another important factor for the success of link analysis algorithms is that the underlying graph should not be sparse. Table 5 shows the number of vertices and edges in the three example Semantic Webs. (For the UMLS Semantic Web, although the Semantic Network had 135 classes, only 92 were connected to others by links. The nonconnected classes were ignored). While UMLS and Patent Semantic Webs are dense graphs, the TAP Semantic Web is very sparse. One problem with the link analysis algorithm is the *Tightly-Knit Community (TKC) Effect* [11]. A tightly-knit community is a small but highly interconnected set of sites. Roughly speaking, the TKC effect occurs when such a community scores high in link-analyzing algorithms, even though there may be more important sites. In a sparse graph, this problem will be more pronounced and a small but highly interconnected set of nodes may dominate the scoring since many others may not be well connected. Thus, in the TAP Organization Semantic Web, pages related to CMU Computer Science Department were well connected and therefore had very high scores.

Another observation is that a resource may have high scores not because it is important but because it is very common. For example, if in the Patent Semantic Web the countries of the inventors and assignees were included, a resource for the United States of America may become the most important resource in the Web. This is similar to common Web sites like Yahoo getting high importance scores for WWW link analysis. To prevent this, the user may specify the type of resources for which the importance scores are to be calculated. Another option is to have a Stop Resource list that includes the common resources in the Semantic Web.

5 Retrieving Information from the Biomedical Semantic Web

SemanticWebRetriever is the runtime component of the BioPatentMiner running inside a Web Application Server. It facilitates various types of information retrieval from the Semantic Web. We will discuss these techniques in the next two sections.

5.1 Keyword Search

We support keyword search on the annotated patents using the Juru XML search engine [3]. The patents can be retrieved based on various criteria similar to USPTO. 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 (at the time of the experiment). On the other hand, our system retrieved 196 patents for the query "*nucleic acid*" on a collection of patents with the keyword *glycolysis* in title or abstract. 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).

For several queries, hundreds of patents may be retrieved and effective ways of ranking the results are required. USPTO ranks the retrieved patents by the issue date of the patents. 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.

5.2 RDQL

We develop Semantic Webs using Jena [5] which utilizes RDQL, a query language for RDF [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 (based on a RDFS rule). Fig. 3 shows the results of an example RDQL query on a collection of patents on "*protein protein interactions*." The query retrieves UMLS concepts that are the subject and

1	patent- http://9.182.217.101/semWeb/Data/PPIPatents/P_5972625.xml subject- leukocyte adhesion molecule, lam 1 object- cluster of diff antigen 62
2	patent- http://9.182.217.101/semWeb/Data/PPIPatents/P_5972625.xml subject- l selectin object- p-selectin
3	patent- http://9.182.217.101/semWeb/Data/PPIPatents/P_5922842.xml subject- kinase, tyrosine object- peptide
4	patent- http://9.182.217.101/semWeb/Data/PPIPatents/P_6133419.xml subject- transphosphorylases object- protein, nos
5	patent- http://9.182.217.101/semWeb/Data/PPIPatents/P_5972625.xml subject- antigens, cd621 object- padgem-pl act gr ext mem prot

Fig. 3. Results of a RDQL query showing patents having relations of type *interact* between biological entities.

object of all relations of type *interact*. The patents supporting the relations are also retrieved. This type of information cannot be easily retrieved from the USPTO.

5.3 Summarizing a Cluster of Patents

Sometimes a Patent Analyst may want to identify the main theme in a collection of biomedical patents. The collection may be the result of a text search or any other collection of interest to the user. BioPatentMiner allows the summarization of a collection of patents based on the important biological terms contained in them.

Given a collection of patents, our objective is to identify a set of important biological terms from the annotated patents which gives an "optimal" summary of the biological properties and functions of the entire collection of patents. Just determining the terms with the highest frequency would not give good results. We have developed a technique which determines the important biological terms based on a number of factors including:

1. The number of patents a term occurs in.
2. The frequency of occurrence of the term in the patents.
3. The particularity (if any) of a term to a small set of patents. (This determines terms which are not present in all the patents but are very important to a subset).

Our method is derived from a technique we have developed to summarize a cluster of genes [7].

We have evaluated our technique on patents obtained by a Juru search on various collections of patents. For example, we queried a collection of US patents related to *diabetes* with the keyword *insulin*. The list of important biological terms for the 412 patents retrieved by the query is shown in Fig. 4.

Diabetes is a *Metabolic disorder/disease* whose biological name is *Diabetes Mellitus*. Its common manifestation is an abnormally high level of *Glucose* in *Blood* (*hyperglycemia*). Diabetes is of two types, *Insulin-dependent* which is characterized by insufficient *insulin* in body and *Noninsulin-Dependent* characterized by diminished ability of body cells to use insulin (*Insulin Resistance*). Non-Insulin-Dependent diabetes patients often exhibit *Obesity*. Fig. 4 shows that most of the important terms related to diabetes occur in the list. Moreover, the term "*Diabetes Mellitus, Non-Insulin-Dependent,*" which is the more common form of diabetes,

Term Listings
1) Insulin
2) Diabetes
3) Insulin Resistance
4) Salts
5) Iodine
6) Diabetes Mellitus, Non-Insulin-Dependent
7) Glucose
8) Hyperglycemia
9) Obesity
10) Blood
11) Tissues
12) Sulfur
13) Pharmaceutical Preparations
14) Hormones
15) Peptides
16) Atherosclerosis
17) Nitrogen
18) Metabolic Diseases
19) Antibodies
20) Diabetes Mellitus, Insulin-Dependent

Fig. 4. The biological terms summarizing the patents retrieved by searching a collection of *diabetes* patents with *insulin*.

occurs much higher in the ranks than "*Diabetes Mellitus, Insulin-Dependent*."

6 Semantic Association

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.

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 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 Fig. 5 represent a propertyGraph. Several resources are shown with the dashed arrows representing paths between the resources and solid arrows representing edges between the resources. We will also discuss an efficient implementation of Semantic Associations and present some retrieval scenarios.

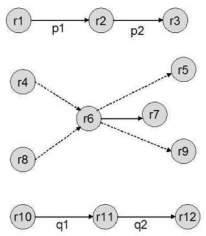

Fig. 5. An Example propertyGraph.

6.1 ρ-path-associated

Two resources r_1 and r_2 are ρ-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 Fig. 5, resources $(r4, r9)$ and $(r5, r8)$ are ρ-path-associated.

6.1.1 Implementation

To determine whether two resources are ρ-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 [21], 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:

1. 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.
2. While $(n < N)$ && $(th >= 0)$:
   a. Filter the property graph to include only $r1$ and $r2$ and resources whose importance is greater than th.
   b. Determine the directed paths from $r1$ to $r2$ as well as $r2$ to $r1$ in the filtered graph.
   c. 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.

TABLE 6
Number of Paths of Different Lengths for Different Values of Threshold between *Biologically_Active_Substance* and *Biologic_Function*

Path Length	Threshold				
	0.0	0.05	0.1	0.2	0.3
1	2	2	2	2	2
2	3	3	3	3	1
3	6	6	6	4	1
4	20	20	18	8	1
5	93	93	82	37	0
Total paths of length ≤ 5	124	124	111	54	5

6.1.2 Example

Table 6 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 0.3 there are only five 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, Fig. 6a is a visualization that shows the ρ-path-associated directed paths of length $\leq$ 5 between *Biologically_Active_Substance* and *Biologic_Function* for a threshold of 0.3. 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. For example, Fig. 6b shows the path association at a threshold of 0.1. Now, there are many more paths.

6.2 ρ-join-associated

Two directed paths in the propertyGraph are said to be *joined* if they have one vertex common. The common vertex is the *join node*. For example, in Fig. 5 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 ρ-join-associated if there are joined paths p_1 and p_2 and either of these two conditions is 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, $(r4, r8)$ and $(r5, r9)$ are sets of ρ-join-associated resources in the example graph.

6.2.1 Implementation

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:

1. 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.

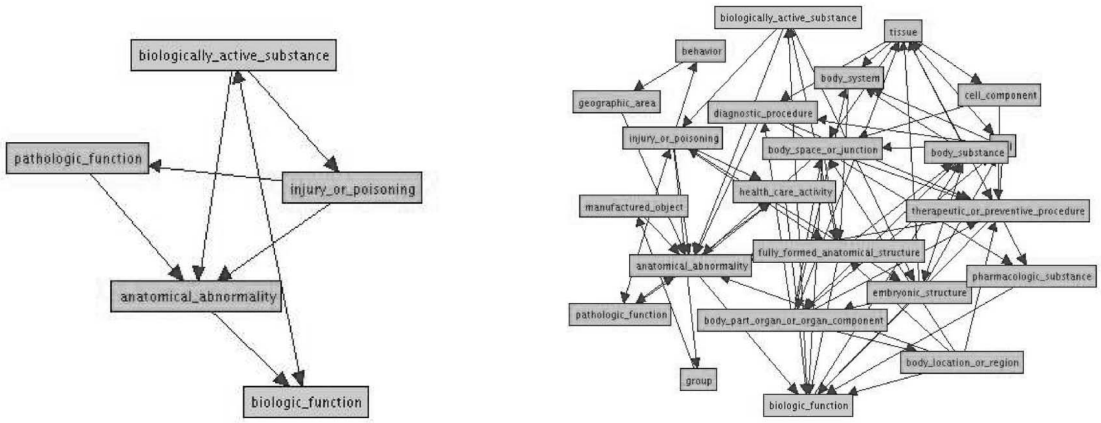

Fig. 6. Visualizing the path associations between two UMLS Classes at thresholds of 0.3 and 0.1, respectively.

2. While $(n < N)\&\&(th >= 0)$.
   a. Filter the property graph to include only $r1$ and $r2$ and resources whose importance is greater than th.
   b. 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.
   c. 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 Fig. 5 while determining the join association between $r4$ and $r8$).
   d. 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.
   e. 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.

6.2.2 Example

Fig. 7 shows the Join Associations between two assignees *Pfizer* and *Ranbaxy* in a collection of patents about these two companies. The paths from the assignees to the join nodes in the Semantic Web graph are displayed. It shows that the assignees are related based on several patents and UMLS concepts which are the join nodes. For example, *Ranbaxy* is assigned a patent 6673369 which refers to the patent 6068859 of *Pfizer*. This kind of information may be useful for the companies for discovering potential patent infringements. Note that this technique of determining semantic associations is useful for all classes of patents and not restricted to the biomedical domain.

6.3 ρ-cp-associated

Two resources r_1 and r_2 are ρ-cp-associated if r_1 is of type c_1, r_2 is of type c_2, and either of these two conditions is 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 ρ-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* ρ-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*.

6.3.1 Implementation

The procedure $\rho\_cp\_associated(r1, r2, L, Ceiling)$ determines the $cp\_associations$ between the resources $r1$ and $r2$. L and

Join Node	Path from pfizer	Path from ranbaxy
P 6068859	pfizer→P 6068859	ranbaxy→P 6673369→P 6068859
P 5697922	pfizer→P 5697922	ranbaxy→P 6261601→P 5697922
hydrogen, nos	pfizer→P 5728711→hydrogen_nos	ranbaxy→P 5410044→hydrogen_nos
salts	pfizer→P 6436987→salts	ranbaxy→P 5948440→salts
high blood cholesterol level	pfizer→P 5703052→high_blood_cholesterol_level	ranbaxy→P 6541511→high_blood_cholesterol_level
capsules	pfizer→P 5358502→capsules	ranbaxy→P 6296871→capsules
P 5705190	pfizer→P 6068856→P 5705190	ranbaxy→P 6673369→P 5705190

Fig. 7. Join Associations between two assignees. Paths from the assignees to the join nodes are shown.

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:

1. Determine the set of classes $C1$ and $C2$ that the resources belong to. (If the resources are themselves classes, this step is not necessary).
2. The ancestors of $C1$ and $C2$ can be determined from the 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.
3. 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 $\rho\_cp\_associated$.
4. 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.

6.4 ρ-iso-associated

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

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

Two resources are ρ-iso-associated if they are the origins of isomorphic paths. For example, in Fig. 5 if $p1 \subset q1 \wedge p2 \subset q2$, $r1$ and $r10$ are ρ-iso-associated.

6.4.1 Implementation

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 ρ-iso-associated (with an isomorphic path of length one). Therefore, determining whether two resources are ρ-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.

7 Conclusion

This paper introduced BioPatentMiner, a system that incorporates various techniques to retrieve information about biomedical patents. The system identifies and classifies the biologically significant terms in the patents and determines relations between these terms. Then, the information from the patents is integrated with concepts in biomedical dictionaries to create a Semantic Web. The system incorporates a method to calculate the importance of Semantic Web resources that can be used to rank the results of a query. We have also developed algorithms 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:

1. 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.
2. We plan to improve the recall of our Relation Annotator by adding new templates for discovering other relation patterns in sentences. To improve the precision, we need to fine-tune the existing templates. Another challenge is that currently we are only determining the relations from the Patent abstracts. We need to examine the full-text of the patents also. Our initial observation is that since the patents are written by patent attorneys, information extraction from them is more difficult than journal publications. Since the patents are written using more complex sentences, we need to break these sentences into simpler chunks and also incorporate Anaphora resolution.
3. One of our major concerns is scalability. Systems like Jena cannot handle really large Semantic Webs and answer queries in real-time. One option is to have a distributed Web and modify our algorithms to retrieve information from multiple distributed sources.
4. There are various sources of biomedical knowledge like patents, research publications, and ontologies. Since it is difficult for researchers to easily gain an 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," *Proc. 12th Int'l World-Wide Web Conf.*, May 2003.
[2] 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 Int'l World-Wide Web Conf.*, vol. 30, nos. 1-7, pp. 107-117, Apr. 1998.
[3] D. Carmel, E. Amitay, M. Hersovici, Y. Maarek, Y. Petruschka, and A. Soffer, "Juru at TREC-10: Experiments with Index Pruning," *Proc. 10th Text Retrieval Conf.*, pp. 228-237, 2001.
[4] J. Chen, L. Wong, and L. Zhang, "A Protein Patent Query System Powered by Klesili," *Proc. ACM SIGMOD Conf.*, 1998.
[5] JENA, http://www.hpl.hp.com/semweb/jena2.htm, 2005.
[6] N. Kando and M. Leong, "Workshop Patent Retrieval: SIGIR 2000 Workshop Report," *ACM SIGIR Forum*, vol. 34, no. 1, pp. 28-30, Apr. 2000.
[7] P. Kankar, S. Adak, A. Sarkar, K. Murari, and G. Sharma, "MedMeSH Summarizer: Text Mining for Gene Clusters," *Proc. Second SIAM Int'l Conf. Data Mining*, 2002.

[8] S. Karvounarakis, S. Alexaki, V. Christophides, D. Plexousakis, and M. Scholl, "RQL: A Declarative Query Language for RDF," *Proc. 11th Int'l World-Wide Web Conf.*, May 2002.
[9] J. Kleinberg, "Authorative Sources in a Hyperlinked Environment," *Proc. Ninth ACM-SIAM Symp. Discrete Algorithms*, May 1998.
[10] L. Larkey, "A Patent Search and Classification System," *Proc. ACM Digital Library Conf.*, 1999.
[11] R. Lempel and S. Moran, "The Stochastic Approach for Link-Structure Analysis (SALSA) and the TKC Effect," *Proc. Ninth Int'l World-Wide Web Conf.*, pp. 387-401, May 2000.
[12] M. Marinescu, M. Markellou, G. Mayritsakis, K. Perdikuri, S. Sirmakessis, and A. Tsakalidis, "Knowledge Discovery in Patent Databases," *Proc. ACM Conf. Information and Knowledge Management*, 2002.
[13] M. Osborn, T. Strzalknowski, and M. Marinescu, "Evaluating Document Retrieval in Patent Database: A Preliminary Report," *Proc. ACM Conf. Information and Knowledge Management*, 1997.
[14] J. Park, H. Kim, and J. Kim, "Bidirectional Incremental Parsing for Automatic Pathway Identification with Combinatory Categorial Grammar," *Proc. Pacific Symp. Biocomputing*, pp. 396-407, 2001.
[15] J. Pustejovski, J. Castano, J. Zhang, M. Kotecki, and B. Cochran, "Robust Relational Parsing over Biomedical Literature: Extracting Inhibit Relations," *Proc. Pacific Symp. Biocomputing*, 2002.
[16] Resource Description Format, http://www.w3.org/1999/02/22-rdf-syntax-ns, 2005.
[17] Resource Description Format Schema, http://www.w3.org/2000/01/rdf-schema, 2005.
[18] A. Seaborne, "RDQL: A Data Oriented Query Language for RDF Models," http://www.hpl.hp.com/semweb/rdql-grammar.html, 2005.
[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," *Proc. ACM Conf. Information and Knowledge Management*, 2003.
[20] TAP, http://tap.stanford.edu, 2005.
[21] R. Tarjan, "Fast Algorithms for Solving Path Problems," *J. ACM*, vol. 28, no. 3, July 1991.
[22] UMLS, http://umlsks.nlm.nih.gov, 2005.
[23] United States Patent and Trademark Office, http://www.uspto gov/patft/, 2005.
[24] L. Wong, "PIES: A Protein Interaction Extraction System," *Proc. Pacific Symp. Biocomputing*, pp. 520-531, 2001.

Sougata Mukherjea received the bachelor's degree from Jadavpur University, Calcutta, the MS degree from Northeastern University, Boston, and the PhD degree from the Georgia Institute of Technology, Atlanta (all in computer science). He is a research staff member at IBM India Research Lab. Before joining IBM, he held research and software architect positions in companies in Silicon Valley, California, including NEC USA, BEA Systems, and Verity. His research interests include information visualization and retrieval and applications of text mining in areas like Web search and bioinformatics.

Bhuvan Bamba received the BTech degree in 2003 from the Indian Institute of Technology, Madras. He is currently working at IBM India Research Lab with the Knowledge Management Group. His areas of interest are data mining, information retrieval and visualization, information integration, natural language processing, databases, Semantic Web, bioinformatics, and computer vision. He has been with IBM since July 2003.

Pankaj Kankar received the BE degree in electronics and communication engineering from NIT Surathkal, India, and the ME degree in electronics from BITS Pilani, India. Since June 2000, he has been working as a research staff member at IBM India Research Lab. Prior to that, he worked for three years at Hughes in the areas of telecommunication and networking. His research interests include text mining, clustering and programming models, and tools for voice user interfaces.