Discovering Semantic Biomedical Relations Utilizing the Web

SAURAV SAHAY
Georgia Institute of Technology
SOUGATA MUKHERJEA
IBM India Research Lab
EUGENE AGICHTEIN, ERNEST V. GARCIA
Emory University
and
SHAMKANT B. NAVATHE and ASHWIN RAM
Georgia Institute of Technology

To realize the vision of a Semantic Web for Life Sciences, discovering relations between resources is essential. It is very difficult to automatically extract relations from Web pages expressed in natural language formats. On the other hand, because of the explosive growth of information, it is difficult to manually extract the relations. In this paper we present techniques to automatically discover relations between biomedical resources from the Web. For this purpose we retrieve relevant information from Web Search engines and Pubmed database using various lexico-syntactic patterns as queries over SOAP web services. The patterns are initially handcrafted but can be progressively learnt. The extracted relations can be used to construct and augment ontologies and knowledge bases. Experiments are presented for general biomedical relation discovery and domain specific search to show the usefulness of our technique.

Categories and Subject Descriptors: J.3 [**Computer Applications**]: Life and Medical Sciences—*Medical information systems*; H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Linguistic processing*; I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*language parsing and understanding*

General Terms: Algorithms, Experimentation, Languages

Additional Key Words and Phrases: Ontology construction, relation identification

ACM Reference Format:

Sahay, S., Mukherjea, S., Agichtein, E., Garcia, E. V., Navathe, S. B., and Ram, A. 2008. Discovering semantic biomedical relations utilizing the Web. ACM Trans. Knowl. Discov. Data. 2, 1, Article 3 (March 2008), 15 pages. DOI = 10.1145/1342320.1342323 http://doi.acm.org/10.1145/1342320.1342323

Author's address: S. Sahay, College of Computing, Georgia Institute of Technology, 801 Atlantic Drive, Atlanta, GA 30332.

1. INTRODUCTION

Semantic Web [Lee et al. 2001] is a vision of the next generation World Wide Web in which data from multiple sources described with rich semantics is integrated to enable processing by humans as well as software agents. Semantic Webs are described using the Resource Description Format (RDF) language, which provides a simple data model for describing relationships between resources in terms of named properties and their values. Resources can represent diseases, countries, companies, movies or any other entity or concept whose properties need to be represented semantically. RDF describes a Semantic Web using RDF *Statements*, which are *triples* of the form *<Subject, Property, Object>*. Subjects are *resources*. 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.

It is obvious that identifying relations between resources and describing them as RDF triples are essential initial steps to realize the vision of Semantic Web. However, the current situation of the Semantic Web is one of a vicious cycle wherein a true Semantic Web is nonexistent due to the lack of a semantic markup of data, which in turn arises due to the difficulty of discovering and establishing relationships among concepts and resources existing on the Web.

One of the goals of Semantic Web research is to incorporate most of the knowledge of a domain in an ontology that can be shared by many applications. Various ontologies and knowledge bases have been developed for several domains For example, *Unified Medical Language System (UMLS)* is a consolidated repository of medical terms and their relationships, spread across multiple languages and disciplines (chemistry, biology, etc). These ontologies organize information of various resources, each with their attributes, and describe simple relationships like *is-a* and *part-of* between concepts. However, they generally do not incorporate complex relationships between resources. For example, although UMLS contains details about many diseases, viruses and bacteria, it does not incorporate relations between diseases and the causes of the diseases. Therefore, representing these ontologies and knowledge sources in Semantic Web ontology languages will not be sufficient to create a Semantic Web.

Within a short time the World Wide Web has become the most comprehensive repository of information. From the Web one can easily determine relations between resources for most domains. Thus one can determine the cause of typhoid, the capital of Fiji, or the CEO of IBM. However, it is very difficult to utilize automated techniques to extract knowledge from unstructured Web pages. Moreover, because of the very large amounts of information, it is impossible to extract all these information manually and augment Semantic Web ontologies and knowledge bases.

This paper presents a novel technique to automatically discover relations from the Web resources utilizing their Web search services. We first query the search engines with lexico-syntactic patterns to retrieve relevant information. These patterns are initially handcrafted but can be progressively learnt. Instead of downloading Web pages, we extract relations from snippets, the small section of the result pages that contain relevant text from the search results containing the query string and titles from Pubmed abstracts. Since there are

millions of Pubmed abstracts, processing titles of relevant abstracts for fast online information extraction and precise relation discovery is reasonable. Thus the process is very efficient. The knowledge discovered by this technique can be used to augment the ontologies and knowledge bases and create a Semantic Web of a specific conceptual domain. We have utilized the technique to discover relations from general biomedical field to a specialized biomedical subdomain for domain specific ontology construction. Our experiments show the promise of our technique.

The article is organized as follows. The next section cites related work. Section 3 describes our system in detail. We introduce the technique to automatically discover any arbitrary relation between resources as well as how the discovered relations can augment ontologies. We also explain a bootstrapping based technique to learn the patterns for querying the search engines. Section 4 presents experiments to evaluate our techniques. Finally, Section 5 is the conclusion.

2. RELATED WORK

2.1 Information Extraction

Information extraction has long been the focus of active research. One of the most challenging tasks is to extract all the relations between entities that are specified in a document. To determine these relations, one approach is to use templates that match specific linguistic structures. For example, Wong [2001] utilizes templates to determine protein-protein interactions from biomedical literature. Machine Learning based approaches have also been utilized for extracting relations from unstructured text. For example, DIPRE [Brin 1999] and Snowball [Agichtein and Gravano 2000] use bootstrapping, a general class of semi-supervised learning algorithms for extracting relations. On the other hand, Zelenko et al. [2003] utilize fully supervised learning methods for extracting relations. McDonald [2005] gives a good overview of the Machine Learning based approaches for relation extraction.

Another challenge of information extraction systems is performance. These systems generally take more than 7 seconds to process an average-size document. Therefore it is not feasible to utilize these systems on a really large text corpus. To overcome this problem Agichtein and Gravano [2003] presented a technique to query text databases to retrieve "promising" documents; the Information Extraction system processes only these documents.

In this paper we introduce a technique of identifying relations between resources. We utilize a Web search engine to first determine Web pages that have those relations. Our system is very efficient because instead of downloading these documents we only process the result snippets. Since these snippets are only one or two sentences, information extraction is much simpler.

2.2 Knowledge Extraction from Large Text Collections using Search Engines

Marti Hearst had suggested that hyponyms could be acquired from Large Text Corpora [Hearst 1992]. For example, consider the sentence "The bow lute, such

as the Bambara ndang, is plucked." Even if we have not encountered the terms *bow lute* and *Bambara ndang*, we can infer from the sentence that *Bambara ndang* is a kind of *bow lute*. Thus lexico-syntactic patterns can be utilized to discover information from a large Text corpus.

This technique has been successfully utilized to discover knowledge from the World Wide Web, the largest Text corpus available for machine processing today. Instead of gathering information from the Web directly, these systems utilize Web search engines which have already crawled and indexed the information. For example, Know-it-all [Etzioni et al. 2004] was able to extract thousands of facts automatically using Web search engines. Similarly, PANKOW [Cimiano et al. 2004] could automatically discover names of resources like countries, cities and rivers. Several limitations of the PANKOW system have been alleviated by C-PANKOW [Cimiano et al. 2005].

Classification of terms is the determination of IS-A relation between the term and a class. Marti Hearst's idea has also been utilized to determine other type of relations including *part-of* [Berland and Charniak 1999] and causal [Girju and Moldovan 2002]. In this article we attempt to identify any arbitrary relation between two entities, which is a much more challenging problem.

Techniques have also been developed for learning the patterns with which to query search engines. For example, Etzioni et al. [2004] present extensions to the Know-it-all system to improve its recall. In order to ensure that more terms can be correctly classified by querying WWW search engines, techniques like Rule Learning, Subclass Extraction, and List Extraction were introduced. We have developed a technique for learning patterns for querying WWW search engines which is similar to the Rule Learning method; however we have generalized it for any type of relations. Our method is a bootstrapping based learning technique similar to DIPRE [Brin 1999] and Snowball [Agichtein and Gravano 2000]. The main difference from the earlier systems is that we do not need to examine the full text to learn patterns for extracting relations; we just examine the snippets returned by the search engines.

Biomedical Entity Annotation is a challenging research area that is precursor for efficient discovery of relations. Biological term extraction systems can be broadly divided into two types: those with a rule base and those with a learning method. In Fukuda et al. [1998], protein names are identified in biological papers using hand-coded rules. On the other hand, in Collier et al. [2000], supervised learning methods based on Hidden Markov Models are used. Subramaniam et al. [2003] have developed the BioAnnotator system, which is part of the current Relation Extraction system, and uses rules and dictionary lookup for identifying and classifying biological terms.

3. SYSTEM DESCRIPTION

In this section we will explain our technique utilizing the search engine results to discover relationships between resources. We also describe how the discovered relations can be used to augment Semantic Web ontologies and knowledge bases. Our method of learning the patterns to query search engines is also presented.

```
relationIdentifier(resource,property) {
   patterns = List of patterns in the Pattern Database for property
   synonyms = List of synonyms in the Ontology for resource
   initialize a Hash Map resultResources

   for each s in synonyms {
      for each p in patterns {
         queryString = p with ‘‘RESOURCE’’ replaced by s
         results = SearchResultSnippets(‘‘queryString’’)
         for each result in results {
            parsedResult = PoSTag(result)
            entityAnnotatedResult = EntityAnnotate(parsedResult)
            relAnnotatedResult = RelationAnnotate(entityAnnotatedResult)
            resultResource = relationEntity(relAnnotatedResult,s)
            resultResources{resultResource}++
         }
      }
  }
  return resultResources
}
```

Fig. 1. Pseudocode to determine the entity that has the relations specified by *property* with *resource*.

3.1 Relation Identification

Let us assume that our objective is to discover causal relationship between a disease and a biological entity. Given a disease d and a biomedical entity e, we can query search engines with phrases like “e causes d” or “d is caused by e” and count the number of results that are retrieved. However, there are thousands of entities (viruses, bacteria, parasites, etc.) that can cause a disease. Querying for each of them is not efficient. It would be more useful if given a disease we can discover the likely causes of the disease.

We have implemented a generic framework for discovering relations between resources. Figure 1 shows the pseudocode to determine the entity that takes part in relations specified by *property* with *resource*. For each property patterns that indicate each of these relations are manually determined and entered in a *Pattern Database*. Example pattern for the property “cause” is as follows:

—**causes**: causes RESOURCE, RESOURCE is caused by

More common patterns that occur on Web databases between resources can be learnt by our Pattern Learner module to augment the Pattern Database. This is discussed in Section 3.3.

We also determine synonyms for the given resource using an ontology. For each synonym and each pattern we issue phrase queries to a Search Engine. Presently we utilize Google WWW search engine and Pubmed database search engine. Thus if we want to determine p53 gene effectors, we would issue queries like “p53 is affected by,” “affects p53,” “bears on p53,” “impacts p53,” etc. We are using WordNet and UMLS ontologies for this purpose.

Previous systems like Know-it-all [Etzioni et al. 2004] and PANKOW [Cimiano et al. 2004] classify entities by counting the number of results retrieved by Google. Unfortunately, just the number of results is not sufficient for our purpose. However, downloading the result pages will make the process very slow. Therefore, we utilize the *result snippets*, the small section of the result pages that contain the query string that is returned with a Google search, and *abstract titles*, which are returned by the Pubmed web services search calls.

We determine the resource that is related to the given *resource* from these result snippets using 3 components:

—We first parse the snippet using a *Part-of-Speech Tagger*. This identifies entities like Noun Phrases, Verb Groups, etc.

—Then an *Entity Annotator* determines the resources (or entities) in the strings using ontologies as well as a Rule Engine. If all the synonyms of a resource are specified in an ontology, the Entity Annotator can identify a resource in a snippet in spite of variations in its naming. In many cases the ontology may not be comprehensive and may not contain all possible resources. In that case our Entity Annotator can recognize names of entities like variations of ontological terms not present in the ontology, Chemicals, etc. using a Rule Engine.

—Finally a *Relation Annotator* discovers the relations between the resources. At present we are using a simple template-based technique for relation identification. For example, some common templates which specify relationships in sentences are:
  —*Subject Verb_Group Object* (For example, "HIV causes AIDS")
  —*Object Passive_Verb_Group Subject* (For example, "AIDS is caused by HIV")
  —*Noun (Nominal form of verb) Object Subject* (For example, "causing of AIDS by HIV")

  If a template is matched it is assumed that a relation of the matching verb group (or nominal form) has been identified. Note that if there are noun phrases or adjectives between the entities and the verb groups in the sentences they are considered as qualifiers for the result resource. We have avoided using a deep parser as it considerably slows down the relation identification process for relation triples. Identification of complicated relations in longer sentences would deeply benefit from using a dependency parser.

The combination of Entity Annotator and Relation Annotator creates an annotated string from which the entity taking part in the relation with the *resource* can be easily identified. For example given the result snippet "AIDS is caused by HIV," the Part-of-speech tagger will recognize "is caused by" as the Verb Group, Entity Annotator will recognize *AIDS* and *HIV*, and the Relation Annotator recognizes *HIV* as the resource that is in causal relationship with *AIDS*. On the other hand for the more complex result snippet "Metabolic bone disease is caused by the lack of Vitamin D3", the Relation Annotator recognizes "Vitamin D3" as the resource that is in causal relationship with "Metabolic bone disease" with the qualifier *"the lack of."*

Different authors will express the same semantics in different ways. Therefore, there will be variations in the results that are retrieved by search engines. For example, one result may state that *AIDS* is caused by *HIV*, while another may state that the disease is caused by *Human Immunodeficiency Virus*. However, Entity Annotator will map them to the same resource using ontologies. Therefore, Relation Annotator will identify the same biological entity from the two search results. However, this may not be true for all snippets. For example, if one snippet states that *"Metabolic bone disease"* is caused by "the lack of Vitamin D3" and another states that it is caused by "Calcium deficiency," our annotators will not be able to match the two entities. Therefore, as shown in Figure 1, a hash map that has the resources that have the specified relation with the given resource along with the number of occurrences for each of them are returned from the *relationIdentifier* procedure.

3.2 Augmenting Ontologies

Several ontologies and knowledge bases have been developed for various domains. For example *Unified Medical Language System (UMLS)* is a consolidated repository of medical terms spread across multiple languages and disciplines. An essential section of UMLS is a **Semantic Network** that has 135 biomedical semantic classes like *Gene or Genome* and *Amino Acid, Peptide, or Protein* and 54 relationship categories like *treats*, *diagnoses*, and *part of*. In addition there are biological concepts each of which are associated with one or more semantic classes. Similarly, the TAP project [Guha and McCool 2003] has developed Semantic Web knowledge bases for several non-biomedical domains including Organization, Sports and Music. (Their technique of identifying an entity and its class is specified in Dill et al. [2003]).

However, most of these ontologies only encode simple relationships like *is-a* and *part-of*. Thus although the semantic classes of UMLS are linked by a set of 54 semantic relationships (like *prevents*, *causes*, etc.), there are no relationships between the biomedical concepts themselves. To develop a comprehensive Semantic Web, discovering relations between the resources is essential. We have utilized our technique for relation identification to augment Semantic Web ontologies and knowledge bases. Figure 2 shows the overall architecture of the system.

We assume that the ontologies have multiple classes and properties defined. They also have resources which belong to one or more classes. Given a Semantic Web class, the *Ontology Augmenter* first determines the properties appropriate for the class as well as the resources belonging to the class using a Semantic Web query engine. The Ontology Augmenter utilizes the *Relation Identifier* to determine the relations for the resources. Note that the Relation Identifier identifies the synonyms for the resource from the ontology. (We assume the synonyms of the resources are specified in the ontology using the *RDFS:label* tag). Similarly, the Entity Annotator uses the ontology to determine the entities in the search results. Note that if the resource is not in the ontology and is identified using the Entity Annotator Rule Engine, it has to be added to the ontology.

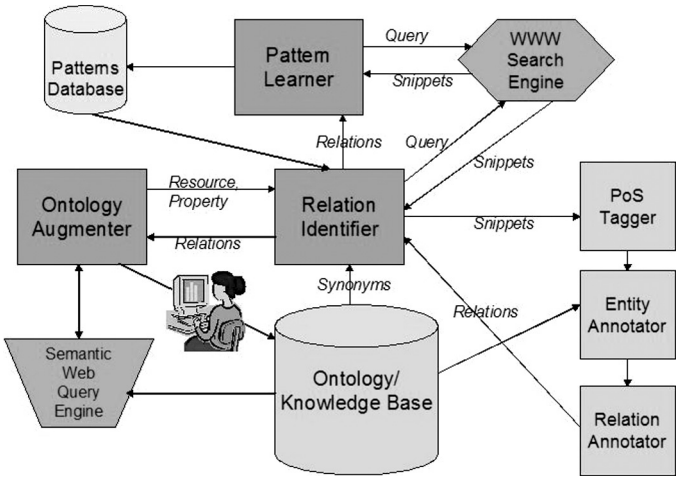

Fig. 2. Architecture of our system.

The Relation Identifier returns a hash map containing the potential relation resources along with the count of the number of snippets that contain the relation. If only one relation resource is identified or the count of one of these relation resources is much higher than the others then that resource is selected. However in other cases user intervention is required to determine the correct resource that has the relation with the given resource. (For many relations more than one relation resource may be appropriate. We discuss this issue further in Section 4).

The Ontology Augmenter adds triples to the Semantic Web ontology or knowledge base for each identified relation. Thus, if given a resource $r1$ and property $p2$, if $r3$ is identified as the relation resource, the triple $< r1, p2, r3 >$ or $< r3, p2, r1 >$ is added to the Semantic Web depending on whether the subject or the object of the relation is identified.

3.3 Learning Patterns

To determine the relation resources for a given property, the Relation Identifier determines the patterns relevant for the property from a Pattern Database. The *Pattern Learner* is an important component of our system that refines the patterns in the Pattern database. Often many of the manually specified patterns for a property may not be appropriate. For each pattern the Relation Identifier keeps a count of the number of snippets retrieved by Google as well as whether the snippets were able to identify a relation resource. This information is passed to the Pattern Learner which, over time, removes patterns that are not useful in determining the relation. This reduces the number of Google queries and improves the efficiency of the system.

A more important task of the Pattern Learner is to augment the Pattern Database with more promising patterns for the properties. Whenever a new relation is correctly identified by the Relation Identifier, that information is passed to the Pattern Learner. Figure 3 shows the algorithm that determines patterns between two given resources that are related by the given *property*. For

```
patternLearner(resource1, resource2, property) {
  synonyms1 = List of synonyms in the Ontology for resource1
  synonyms2 = List of synonyms in the Ontology for resource2

   for each s1 in synonyms1 {
      for each s2 in synonyms2 {
         queryString = s1 AND s2
         results = SearchResultSnippet(queryString)
         for each result in results {
            For each result which contains s1 and s2 in the same sentence {
                 pattern = Text Segment with s1 and s2 abstracted
                 resultPatterns{pattern, property}++
            }
       }
   }
  Add to Pattern Database patterns in resultPatterns with count > THRESHOLD
}
```

Fig. 3. Pseudocode to determine patterns that specify the relationship specified by *property* between *resource1* and *resource2*.

each synonyms for the given resources, a Boolean AND query is issued to Google. We utilize a Sentence Analyzer to determine the sentences in the snippets and only consider snippets where the given synonyms occur in the same sentence. From these snippets we determine the text segment which links the synonyms. We abstract the resources from these segments to create the relation patterns. This technique of generating patterns from example relations is based on the Snowball system [Agichtein and Gravano 2000].

For example, suppose we are determining patterns for the *hasCapitalCity* property. If the two resources are *China* and *Beijing*, we query Google with these two entities. If we determine that the text segment connecting the entities in a returned snippet is "China's capital is Beijing" we can abstract the pattern "RESOURCE's capital is."

We maintain a hash table keeping a count of the number of instances of the different patterns that are identified by the Pattern Learner for each property of interest. If the count for a pattern exceeds THRESHOLD, a predefined constant,[1] we add this pattern to the Pattern Database. This threshold idea is analogous to the notion of "support" for association rules. Note that if for some property we already know many instances of relation resource pairs, we can identify the patterns for that property from the Pattern Learner directly without having to initially handcraft these patterns.

4. EXPERIMENTS

4.1 Relation Identification

We have utilized our technique to identify various types of relationships between resources. However, a formal evaluation of our technique is difficult

[1]For our experiments we choose 10 as the THRESHOLD.

Table I. Some Relations for UMLS Resources Determined by Our Technique

Property	UMLS Resource	Relation Resource
causes	Typhoid	Bacterium Salmonella Typhi
diagnoses	Cyst	Ultrasonography
consists of	Butane	Liquefied Petroleum Gas
affects	Statin	Lipitor, Gemfibrozil, Niaspan
binds	Rhodopsin	Lys296, Transducin

Table II. Some Retrieved Relations for UMLS Resources from Pubmed

Property	UMLS Resource	Relation Resource
causes	Typhoid	Salmonella enterica serotype paratyphi
diagnoses	Cyst	Hypophysitis, Ciliary body melanoma
consists of	Butane	null
affects	Statin	Cholesterol, Angiogenic mediators
binds	Rhodopsin	Arrestin

because there are no test data sets that can be used for the evaluation. For determining the efficiency of our technique, we determined relations for resource for various domains relevant to the UMLS knowledge base and TAP Semantic Web. In the absence of domain experts, we did literature surveys and Web surfing to determine whether the relations identified by our system are correct.

For UMLS besides Semantic Network properties *causes*, *diagnoses*, *consists of* and *affects* we also extracted *binds* relations for several entities of UMLS class *Amino Acids, Peptides, or Proteins*. Table I and Table II show several biomedical relations determined by our technique. Thus we could identify the cause of *Typhoid* (*Bacterium Salmonella Typhi*) as well as entities that affect *Statin* (*Lipitor, Gemfibrozil, Niaspan*) [Mukherjea and Sahay 2006].

For each property, we determined relations for several entities of some particular UMLS class which has that property. To test the system impartially we have included common as well as rare concepts in our experiments. We calculated the following statistics for each property from our experiments:

—**N**: Total number of resources for which we tried to identify relations.
—**F**: The number of resource for which at least one relation was identified by our system.
—**C**: The number of resources for which at least one relation that was identified by our system is correct.
—**Coverage (CV)** $CV = \frac{F}{N}$
—**Correctness (CR)** $CR = \frac{C}{F}$

While *Coverage* measures the number of relations for which results could be obtained from the search engines, *Correctness* measures the ability of extracting the correct relation resource from the returned results. These metrics resemble recall and precision used in Information Retrieval. It is difficult to calculate exact recall as Google limits the searches to 10 results per search query and

Table III. Coverage and Correctness of the Relation Identifier for UMLS Resources

Property	Class	Coverage	Correctness
causes	Disease	0.85	0.82
diagnoses	Anatomical Abnormality	0.9	1.0
consists of	Organic Chemical	0.72	0.75
affects	Gene	0.76	0.8
binds	Amino Acid, Peptide, or Protein	0.75	0.83

Table IV. Coverage and Correctness of the Relation Identifier for UMLS Resources Using Pubmed Search

Property	Class	Coverage	Correctness
causes	Disease	0.9	0.88
diagnoses	Anatomical Abnormality	0.8	1.0
consists of	Organic Chemical	0.5	1.0
affects	Gene	0.8	1.0
binds	Amino Acid, Peptide, or Protein	0.8	1.0

we are only processing the top 100 results returned from Pubmed searches. Table III shows the results for each property and the corresponding UMLS class retrieved from Google search engine. Table IV shows the results for each property and the corresponding UMLS class retrieved from Pubmed database search.

There are several observations when we compare the results of Pubmed and Google searches. Medline abstracts are precise and technical accounts of facts and experiments reported through literature. They assume prior contextual knowledge and are highly domain specific in nature. We observed that they fail to identify the common answers to our queries as returned by Google search. However, they pick up some answers that are rare and can only be found through scientific papers. A good scheme of relation extraction aimed towards ontology construction would be to combine both these techniques to find common as well as rare relations for domain specific searches.

4.1.1 *Quality of the Relation Identifier.* The quality of the Relation Identifier is affected by various factors:

—The coverage is affected by Google's inability to identify complex class associations such as chemicals, genes, proteins and their relationships. For example, Google is unable to retrieve any results on our queries such as "binds Auxin Response Factor 1" or "Nephroptosis is diagnosed by."

—Sometimes the snippet returned by Google may not be able to identify the resource property. For example, one snippet retrieved was "Primary Hypertension is caused by abnormalities of" with the relevant cause of the disease stripped off.

—The Relation Identifier fails to identify complex relations embedded in large sentences or spanning multiple sentences (coreference and anaphora resolution).

—Setting a high threshold on relation identifier retrieves high precision results at the cost of recall.

4.2 Augmenting Ontologies

Triples for the identified relations need to be added to the Semantic Web. Ideally there should be limited user intervention at this stage. For this the Relation Identifier should identify only the correct relation resource or the count of the correct resource should be substantially higher than the others. Here are some observations from our experiments:

—For many properties there may be more than one relation resource. Thus, as shown in Table I, many entities may affect a gene. For such properties, many relation resources will have similar counts and user intervention may be required to determine the correct ones.

—For augmenting ontologies, we also need to correctly identify the categories for each class of the relationship triple. One concept in a given relation may belong to multiple class that we need to disambiguate automatically.

Our experiments show that UMLS is really comprehensive and has all biomedical resources and its variations. Therefore our system should be integrated with systems that identifies and classifies entities in Web pages like Know-it-all, PANKOW and SemTag [Dill et al. 2003] to create a comprehensive Semantic Web.

4.3 Learning Patterns

Our Pattern Learner determines new patterns for the properties. For example some of the patterns learnt by the system that were not present initially in the Pattern Database are:

—**causes**: causes of RESOURCE are, cause of RESOURCE is, RESOURCE causing, RESOURCE can cause

—**affects**: RESOURCE affects other, RESOURCE status in

Further experiments are needed to determine the efficiency of our Pattern Learning technique. We have to determine whether the new patterns identified by our technique are selective (that is, they do not generate incorrect relations) as well as have high coverage (that is, they are able to return many correct Google snippets). Our pattern learning is generating many domain specific patterns that we have to abstract using ontologies to retrieve non-specific patterns. We also conducted experiments where the Pattern Learner builds up a large pattern base automatically using the discovered relations to find more patterns. This algorithm is presently not converging and generated many resource specific patterns. For example, on a limited iterative search for *"smoking AND cancer"*, we retrieved these specific patterns:

—*smoking causes cancer*

—*chemicals causing mutations and cancer*

—*identifying environmental chemicals causing mutations*

Table V. Extraction of Semantic Network Relations for Domain Terms

Resource	Coverage	Correctness
Google	0.8	1
Pubmed	.68	1
Journal	0.63	1

4.4 Towards Domain Ontology Construction

The goal of this step was to construct an initial domain ontology for the Nuclear Cardiology subdiscipline where the experiments related to clinical trials and did not involve animal subjects. We used an expert provided list of 41 relevant domain terms for the Myocardial Perfusion Imaging domain (a subdomain of nuclear cardiology). Of these 41 terms, 21 of these did not have an exact match in the UMLS ontology. The Bioannotator Rule Engine was able to find most of the domain terms and their variants therefore we achieved higher coverage values for identified relations.

We performed large scale relation identification for the domain terms using three resources. These were Google search, Pubmed search and restricted Pubmed search restricted to Nuclear Cardiology journals. We created patterns for all UMLS Semantic Network relationships and performed queries corresponding to each domain term and semantic network relationship.

Table V shows coverage and correctness values for the extracted relations.

In order to prune the relations and increase precision, we performed a pattern match of extracted verbs with the Semantic Network relationships using WordNet ontology and categorized the relationships. This helped us create labels for the extracted verbs for the ontology and discard 'OTHER' relations that did not match any Semantic Network relations.

Some examples of relations with categorized links:

—Dipyridamole "AFFECTS" platelet thrombus growth
—Adenosine "BRINGS ABOUT" catecholamine
—Myocardial ischemia "INDICATES" stellectomy
—Ischemic complications "COMPLICATES" coronary angioplasty
—Ischemia "DISRUPTS" neuronal cytoskeleton
—Tc-99m sestamibi "EXHIBITS" parathyroid disease
—Dipyridamole "MEASURES" methotrexate

5. CONCLUSION

This paper introduced a new technique to automatically and efficiently discover relations between resources. It utilizes both general and specialized search engines, the most comprehensive sources of knowledge. Since Web Search engines have crawled and indexed most of the information, we query these engines with several lexico-syntactic patterns to retrieve relevant information. This information is used to discover relations between resources. The discovered relations are used to augment Semantic Web ontologies and knowledge bases. We have utilized our system to discover relations in various subdomains as well

as augment UMLS knowledge base. Our experiments show the promise of our techniques. For most relations, we are able to find at least one correct relation. Our results have high precision and we currently do not have resources to evaluate them. Even for uncommon relations like the binding of genes the technique had coverage and correctness values over 70%. Since we do not download Web pages our technique is very efficient and is suitable for large-scale and online Web mining to augment real-world ontologies. We have also developed a bootstrapping based pattern learning technique which will ensure that the system will become more effective over time. At present we are improving the system based on our observations during our experiments. Our ultimate objective is to utilize the discovered relations to develop a real-world Semantic Web which would store the "meaning" of numerous entities and concepts as well as relations between them. This will enable users and software agents to perform a single *Semantic Search* to retrieve all the relevant information about a resource.

REFERENCES

AGICHTEIN, E. AND GRAVANO, L. 2000. *Snowball*: Extracting relations from large plain-text collections. In *Proceedings of the ACM International Conference on Digital Libraries*. 85–94.

AGICHTEIN, E. AND GRAVANO, L. 2003. Querying text databases for efficient information extraction. In *Proceedings of the International Conference on Data Engineering (ICDE)*, U. Dayal, K. Ramamritham, and T. M. Vijayaraman, Eds. IEEE Computer Society, 113–124.

BERLAND, M. AND CHARNIAK, E. 1999. Finding parts in very large corpora. In *Proceedings of the Association for Computational Linguistics (ACL)*.

BRIN, S. 1999. Extracting patterns and relations from the World Wide Web. Lecture Notes in Computer Science. vol. *1590*, 172–??

CIMIANO, P., HANDSCHUH, S., AND STAAB, S. 2004. Towards the self-annotating web. In *Proceedings of the International WorldWide Web Conference (WWW)*, S. I. Feldman, M. Uretsky, M. Najork, and C. E. Wills, Eds. ACM, 462–471.

CIMIANO, P., LADWIG, G., AND STAAB, S. 2005. Gimme' the context: Context-driven automatic semantic annotation with C-PANKOW. In *Proceedings of the International World Wide Web Conference (WWW)*, A. Ellis and T. Hagino, Eds. ACM, 332–341.

COLLIER, N., NOBATA, C., AND ICHI TSUJII, J. 2000. Extracting the names of genes and gene products with a hidden markov model. In *Proceedings of the International Conference on Computer Linguistics (COLING)*. Morgan Kaufmann, 201–207.

DILL, S., EIRON, N., GIBSON, D., GRUHL, D., GUHA, R., JHINGRAN, A., KANUNGO, T., RAJAGOPALAN, S., TOMKINS, A., TOMLIN, J., AND ZIEN, J. 2003. Semtag and seeker: Bootstrapping the semantic web via automated semantic annotation. In *Proceedings of the World Wide Web conference.*

ETZIONI, O., CAFARELLA, M. J., DOWNEY, D., POPESCU, A.-M., SHAKED, T., SODERLAND, S., WELD, D. S., AND YATES, A. 2004. Methods for domain-independent information extraction from the web: An experimental comparison. In *Proceedings of the National Conference on Artificial Intellegence*, D. L. McGuinness and G. Ferguson, Eds. The MIT Press, 391–398.

FUKUDA, K., TSUNODA, T., TAMURA, A., AND TAKAGI, T. 1998. Toward information extraction: Identifying protein names from biological papers. In *Proceedings of the Pacific Symposium on Biocomputing (PBS'98).*

GIRJU, R. AND MOLDOVAN, D. I. 2002. Text mining for causal relations. In *Proceedings of the FLAIRS Conference*, S. M. Haller and G. Simmons, Eds. AAAI Press, 360–364.

GUHA, R. AND MCCOOL, R. 2003. TAP: A Semantic Web platform. *Comput. Netw. 42*, 5, 557–577.

HEARST, M. A. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the International Conference on Computer Linguistics (COLING)*. 539–545.

LEE, B., HENDLER, T., AND LASSILA, J. 2001. The semantic web. *Scientific Amer*.

McDonald, R. 2005. Extracting relations from unstructured text. Tech. rep., Department of Computer and Information Science, University of Pennsylvania.

Mukherjea, S. and Sahay, S. 2006. Discovering biomedical relations utilizing the World Wide Web. In *Proceedings of the Pacific Symposium on Biocomputing*, R. B. Altman, T. Murray, T. E. Klein, A. K. Dunker, and L. Hunter, Eds. World Scientific, 164–175.

Subramaniam, L. V., Mukherjea, S., Kankar, P., Srivastava, B., Batra, V. S., Kamesam, P. V., and Kothari, R. 2003. Information extraction from biomedical literature: Methodology, evaluation and an application. In *Proceedings of the ACM CIKM International Conference on Information and Knowledge Management (CIKM'03)*. ACM Press, 410–417.

Wong, L. 2001. Pies, a protein interaction extraction system. In *Proceedings of the Pacific Symposium on Biocomputing*, R. B. Altman, A. K. Dunker, L. Hunter, and T. E. Klein, Eds. Vol. 6. World Scientific, 520–531.

Zelenko, D., Aone, C., and Richardella, A. 2003. Kernel methods for relation extraction. *J. Mach. Learn. Res. 3*, 1083–1106.

Received November 2007; accepted December 2007