Towards Bridging the Web and the Semantic Web

Swarnim Kulkarni
Computing and Information Sciences
Kansas State University
Manhattan, KS, USA 66506
swarnim@ksu.edu

Doina Caragea
Computing and Information Sciences
Kansas State University
Manhattan, KS, USA 66506
dcaragea@ksu.edu

Abstract

The World Wide Web (WWW) has provided us with a plethora of information. However, given its unstructured format, this information is useful mainly to humans and cannot be effectively interpreted by machines. The Semantic Web provides information in computer understandable structures (e.g., RDF), but the amount of information on the Semantic Web is limited compared to the amount of information available on the Web. The problem of generating a bridge between the Web and Semantic Web has recently gained a lot of attention. In this paper, we propose a Concept Extractor and Relationship Identifier (CE-RI) system, which acts as a bridge between Web and Semantic Web by providing a "semantic" way of presenting the search results to the user. The Concept Extractor (CE) component of our system makes use of the power of existing search engines coupled with the elegance of PageRank to extract high quality concepts related to the given query. The Relationship Identifier (RI) component finds relationships between the extracted concepts and the given query and presents them to the user in the form of a graph. It also stores the generated results formally, in the form of RDF triples, to facilitate better inferences as compared to traditional search engines. We evaluate our system by comparing its components CE and RI with other similar "state of the art" concept detection and relationship identification systems, respectively. The results produced by our system are either similiar or better than those generated by other systems.

1. Introduction

Ever since its creation, the web has seen a tremendous growth, with thousands of new webpages added every day. The ever increasing diversity and complexity of WWW has made the retrieval of relevant information a challenging task. Modern search engines, such as Google^{TM} and Yahoo!^{TM}, have made this task a lot easier. However, the information crawled by search engines is stored in unstructured format (e.g., plain text or HTML) and is not machine interpretable. The dire need to make this information interpretable by machines has led to what is called - *The Semantic Web*.

The Semantic Web is an extension of the present web, where the information is represented in machine readable format. It can be seen as a platform for information and knowledge exchange for both humans and machines [1]. The Semantic Web involves adding "metadata" to the existing web information. The metadata can be expressed in formal structures, such as RDF. The use of RDF, schemas and inference engines makes it possible to represent data in the form of a large database that can be queried efficiently using languages such as SPARQL. Thus, WWW can be seen as the present, whereas the Semantic Web can be seen as the future. Both hold immense potential and therefore, the problem of developing a bridge between them has allured a lot of researchers in the recent years.

In this paper, we propose an approach that can contribute to filling in the gap between the Web and Semantic Web. Our approach has two main components: a Concept Extractor (CE) component and a Relationship Identifier (RI) component. The first component, CE, exploits the richness of the web and the power of the existing search engines to create an initial document corpus, and to extract concepts related to a given query concept. It then uses the PageRank score along with the document frequency of the extracted concepts, to rank them in the order of their relevance to the original query. Next, the RI component determines how each of these extracted concepts are related to the original query. Specifically, relationships between the query and its related concepts are identified and represented in the form of triples, such as <*Subject, Predicate, Object*>, where the *Subject* stands for the original query, the *Object* stands for a related concept and the *Predicate* represents the relationship between *Subject* and *Object*. The generated results are stored in the form of RDF, which can be queried using query languages, such as SPARQL. Moreover, instead of presenting the results in the traditional format (i.e., links) to the user, we present the results in the form of a semantic relationship graph that can be easily interpreted. We show that our approach, although very simple, is highly effective and generates results comparable to those of some of the existing "state of the art" techniques. It also addresses some of the limitations of these techniques. Overall, the results produced by our approach are indeed very encouraging.

The rest of the paper is organized as follows: Section 2 discusses related work. Section 3 gives a detailed description of the CE-RI system architecture and its underlying algorithms. Section 4 contains the experimental design and results. We compare our system with other similar systems in Section 5 and conclude the paper by providing conclusions and directions for future work in Section 6.

2. Related Work

Two main problems are addressed in this paper: (1) the problem of extracting concepts related to a query concept and (2) the problem of identifying relationships between the query concept and its related concepts. We present work related to each of these problems in what follows.

2.1. Extracting related concepts

The problem of extracting related concepts has received a lot of attention recently and, hence, has seen continuous improvements. The initial work on identifying semantic relatedness and deriving related concepts was done using existing lexical databases, such as Roget's Thesaurus [2] and WordNet [3], [4], [5]. More recently, Strube and Ponzetto [6] have shown that Wikipedia can easily outperform WordNet in finding semantic similarity, when a variety of approaches to semantic relatedness, including paths in graph and the text content in the articles, are employed.

In fact, Wikipedia has been used as a source of concepts by many researchers [7], [8], [9]. Syed et al. [7] have used the articles from Wikipedia along with the category and link graphs to identify concepts which are common to a set of documents. Gabrilovich and Markovitch [8] have proposed an approach called *Explicit Semantic Analysis* (ESA) in which concepts derived from Wikipedia are used to represent the meaning of any text and to compute the semantic relatedness between parts of natural language text. According to Gabrilovich and Markovitch [8], the ESA results are better than the results of other existing approaches. Milne [9] has used an approach similar to the ESA approach. In Milne's approach, the semantic relatedness between terms is computed on the basis of links found between Wikipedia articles corresponding to terms, but the underlying text within an article is not processed. Although successful, the above approaches make use of Wikipedia as knowledge base, and thus, face the following issues: (1) Not every word has a category graph associated with it; (2) A link from a relevant document to a similar document may not always exist.

2.2. Identifying relationships between concepts

Accurately identifying relationships between concepts can be useful in a wide range of applications, including query expansion [10], [11], [12], community mining [13], [14], natural language processing tasks (e.g., language modeling) [15], and word sense disambiguation [16], among others. Many approaches for identifying relationships between concepts have exploited manually compiled semantic databases, such as WordNet [17], [18], [19]. However, due to limitations on the size of the vocabulary of WordNet, researchers have started exploiting the Web as knowledge base. Sun and Zheng [20] have used the web to acquire semantic relationships between concepts. Turney [21] has identified synonyms based on the number of hits returned by a web search engine. Matuso et al. [22] have used a similar approach to determine the similarity between concepts.

Web search engines have been used by Sahami and Heilman [23] to address the problem of determining similarity between short texts (e.g., snippets) that have few or no words in common. Bollegela and Matuso [24] have exploited the page counts and text snippets generated by a Web search engine to determine the semantic similarity between concepts. PageRank-like techniques have also been used to calculate the similarity. Chen et al. [25] have proposed to exploit the text snippets returned by a Web search engine as an important measure in computing the semantic similarity between two words. In their approach, the text snippets for the two words, A and B, are collected and the occurrences of word A are counted in the snippet of word B, and vice versa. One drawback of this approach is that a word may not always appear in the snippet of another similar word.

2.3. Bridging the gap between Web and Semantic Web

Some work has also been done on the problem of bridging the gap between the Web and Semantic Web. Angeletou et al. [26] have proposed the enrichment of folksonomy tags by harvesting the Semantic Web for explicit relations. They have used online ontologies to dynamically collect and combine bits of knowledge. To the best of our knowledge, an approach to bridge the gap between Web and Semantic Web by making use of web search engines and relationship identification algorithms has not been proposed yet.

3. System Architecture

A pipeline approach has been used to extract concepts for a given query and to represent the results as an RDF graph. The architecture for our system (CE-RI) is shown in Figure 1. The system consists of two main modules, *Concept Extractor* and *Relationship Identifier*, and two modules for result generation, *Graph Builder* and *RDF Generator*. The system receives a query from the user via the *User Interface*. The query is fed to the *Concept Extractor* module that extracts the concepts related to the given query. After all the concepts have been extracted for a given query, the query and its generated concepts are provided to the *Relationship*

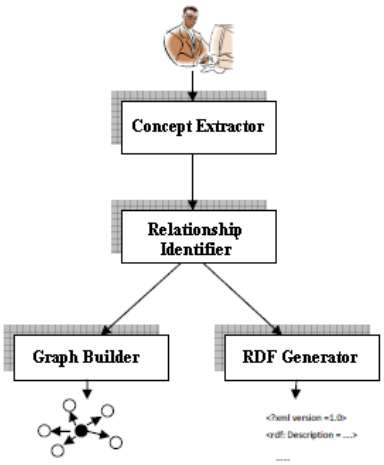

Figure 1: CE-RI system architecture

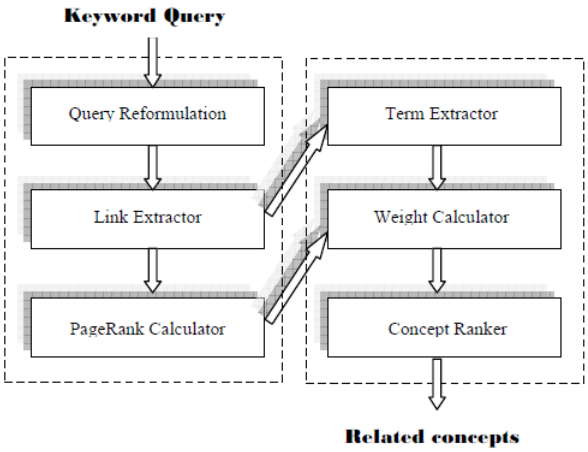

Figure 2: *Concept Extractor* architecture

Identifier module, which determines relationships between the query and its associated concepts (one for each pair <query concept, related concept>).

The *Graph Builder* takes as input the relationships identified by the *Relationship Identifier* and outputs a semantic graph that is presented to the user. The *RDF Generator* interprets the identified relationships in the form of "triples" and creates an RDF document. Next, we will describe each of the modules of the processing pipeline in detail.

3.1. Concept Extractor

The steps of the *Concept Extractor* component are shown in Figure 2. The process starts by taking a query (i.e., set of keywords) from a user via the *User Interface*. The *Query Reformulation* module constructs all possible combinations of keywords for the given query. By considering combinations of keywords, we ensure that a larger area in the concept space, relevant to the given query, is covered. Once the initial query is reformulated, in the next step of the process, the *Link Extractor* feeds all possible query reformulations to a search engine and extracts the resulting links, thus creating a document corpus (which is stored locally). For the results in this paper, Yahoo!^{TM} Search engine was used for this step. The number of pages to be extracted for each possible query formulation can be set by the user. The created document corpus is accessed by the *PageRank Calculator* module, which extracts the links associated with the documents in the corpus and first creates a graph from them (note that this graph might not be very well connected). It then calculates the PageRank scores from this graph (for pages that don't have any links in the graph, a default score is returned). Next, the *Term Extractor* module extracts the terms from the top *n* pages (i.e., those with highest PageRank scores). The parameter *n* can also be set by the user. Its default value is 5. Rather than extracting all terms from the text of a webpage, we exploit the meta information of a webpage. More precisely, we extract the meta keywords along with the title of a webpage. The terms collected by the *Term Extractor* are next sent to the *Weight Calculator* module. This module uses the PageRank scores of the corresponding web pages to calculate the weights of the terms. Specifically, the weight of a term is calculated as follows: Let t be a term and w_t the weight corresponding to the term. Let P_i denote the PageRank score of the webpage i containing the term t. Let k be the number of web pages containing the term t. Then: $w_t = \frac{\left(\sum_{i=1}^{k} P_i\right)}{N}$, where N denotes the total number of documents in the document corpus. Finally, the *Concept Ranker* module ranks the terms in the descending order of their weights. The number of concepts to be displayed depends on the threshold value set by the user.

The *PageRank Calculator* module uses the PageRank algorithm [27] to calculate the PageRank score for each webpage recursively, as follows: If A ia a webpage in the collection, then PageRank score of A is:

$$PR(A) = \frac{d}{N} + (1-d)\sum_{i=1}^{n}\frac{PR(w_i)}{C(w_i)}$$

where $w_1, w_2, \cdots, w_n$ specify the pages that point to the page A, $PR(w_i)$ denotes the PageRank score of the webpage w_i and $C(w_i)$ denotes the number of outgoing links from the page w_i. The PageRank algorithm is based on a random surfer model. This model assumes an imaginary surfer who randomly goes from one page to another through links. Once in a while, the surfer gets bored and jumps to a random page with a probability d, where d is a dumping factor whose default value is set to 0.15.

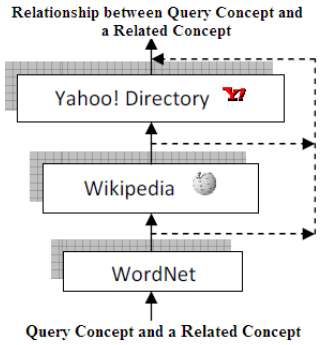

Figure 3: *Relationship Identifier* architecture

3.2. Relationship Identifier

The basic architecture and knowledge bases exploited by the *Relationship Identifier* are shown in Figure 3. As can be seen, three well known knowledge bases are used, namely WordNet, Wikipedia and Yahoo!^{TM} Directories, in this order, to identify relationships between concepts.

The *Relationship Identifier* module is provided with the concepts detected by the *Concept Extractor* module and it identifies relationships between the query concept and its related concepts. More precisely, at each iteration, a pair *<query concept, related concept>* is provided as input and a relationship between the concepts in the pair is determined. The process starts by searching the *related concept* in the WordNet database (which was installed locally, for efficiency reasons). If a relationship can be determined, the relationship is added to the semantic graph to be presented to the user and the current iteration of the *Relationship Identifier* terminates. Next, another *<query concept, related concept>* pair is considered. However, if the *related concept* is not found on WordNet or it is found, but a relationship could not be determined, then the system moves to exploring Wikipedia for a relationship. If Wikipedia also fails to determine a relationship, we use the category structure of *Yahoo!^{TM} Directory* for the relationship identification task. In what follows, we will use several examples to demonstrate how the unique capability of each data source explored can be useful in developing a system that has the power to function accurately and efficiently in many scenarios.

Determining relationships using WordNet. Consider the following example: The system receives as input the query *Flu*. The *Concept Extractor* identifies the following related concepts for *Flu*: *Influenza, Flu Shot, Oseltamivir, Avian Influenza, Hong Kong Flu, Cold* and *Acetaminophen*. Then, the *Relationship Identifier* module receives as input the pair *<Flu, Influenza>* and tries to determine whether the concept *Influenza* is a synonym, hypernym or hyponym of the original query concept Flu. To do that, the concept *Influenza* is searched on WordNet. As this concept is present on WordNet, all its synonyms, hypernyms and hyponyms are extracted and matched against the query concept to identify a relationship. Using the extracted information, *Flu* is found to be a synonym of *Influenza*. Hence, in this particular case, a relationship is determined efficiently from WordNet, without the need to search larger data sources.

In general, if the *related concept* in a *<query concept, related concept>* pair is found on WordNet, but a relationship between those concepts cannot be identified, the WordNet information (i.e., synonyms, hypernyms, hyponyms and antonyms) is stored locally for possibly future use (more details below). A major limitation of WordNet is its limited relationship coverage: not many relationships can be determined using WordNet alone.

Determining relationships using Wikipedia. Consider the following query: *Kansas State University*. One of the related concepts found by our system is *K-State*. We want to find a relationship between the pair of concepts *<Kansas State University, K-State>*. We first use Wikipedia to search for a synonymy relationship, as follows: if the two concepts retrieve the same page, when used as queries on Wikipedia, then we consider them to be synonyms. In our case, when *K-State* and *Kansas State University* are searched on Wikipedia, the same page is retrieved. Hence, we conclude that *K-State* is a synonym of *Kansas State University*.

In general, if a synonymy relationship cannot be determined this way, the first sentence of the Wikipedia article corresponding to the *related concept* is extracted. Next, we search for the *query concept* in the sentence extracted to determine a relationship between the two concepts. If a relationship cannot be determined, but there is information about the *related concept* available from WordNet (although a relationship could not be found using WordNet), the system tries to use this information as a last attempt to find a relationship, before moving to the last data source available to the system (mainly, Yahoo! Directory).

As an example, we consider the concept pair *<Watery Eyes, Conjunctivitis>*. The concept *Conjunctivitis* is found on WordNet, but a relationship with the query concept *Watery Eyes* cannot be identified. Therefore, we store the WordNet information available for *Watery Eyes*, specifically, its synonym, *Pinkeye*. Then, by exploring the information in Wikipedia, we are able to determine a relation between *Watery Eyes* and *Conjunctivitis* from the sentence *Watery eyes is a symptom of Pinkeye* because the system "knows" that *Pinkeye* is a synonym of *Conjunctivitis*. Although very comprehensive, the Wikipedia approach has some limitations: we can only identify relationships to concepts whose corresponding articles can be found on Wikipedia.

Determining relationships using Yahoo! Directory. Again, we consider the query: *Kansas State University*. One of

its related concepts is *Kansas*. To identify a relationship using the category graph structure of Yahoo! Directories, we pose *"Kansas State University" "Kansas"* as a query onto Yahoo!^{TM} Directories and extract the displayed categories from the results. We then search for the *query concept* and the *related concept* in the set of extracted categories and try to identify a relationship between them. In our example, we find the following relationship: *Kansas State University* $\rightarrow$ *Manhattan* $\rightarrow$ *Kansas*. A limitation of this approach is that the determined relationships are limited to the terms present in the category graph of Yahoo!^{TM} Directories.

However, the triple layer exploited in our approach can be effectively used to determine relationships in many cases. Specifically, we use WordNet to determine synonymy, hypernymy and hyponymy relationships. We use Wikipedia mainly to determine synonymy relationships that cannot be identified based on WordNet. Finally, we use Yahoo! Directories to determine relationships between concepts that are not in a synonymy, hyponymy or hypernymy relation.

3.3. Graph Builder

Unlike traditional search engines that display the results in the form of links to documents relevant to the user query, we propose a more "semantic" way of displaying the results. Specifically, we present the results in the form of a Semantic Relationship Graph (SRG). The SRG is a graph structure that contains the query and its related concepts as nodes, and relationships between the query and the related concepts as edges. The graph has the query node in the center. An SRG is similiar to a Semantic Network with the difference that it only shows relationships between the query and its related concepts (but it does not show relationships among the related concepts themselves). The *Graph Builder* module in our system takes the concepts and their relationships as input and builds an SRG that is presented to the user.

3.4. RDF Generator

The last step in our pipeline is to generate an RDF database from the relationships identified by the system, so that the information produced by CE-RI system can be effectively queried using inference engines. Thus, relationships are represented in the form of triples *<Subject,Predicate,Object>*, where the *Subject* is the given query, *Object* is a related concept and the *Predicate* is a relationship between *Subject* and *Object*. The *RDF Generator* takes the concepts and their relationships as input and creates an RDF document.

4. Experimental Design and Results

To evaluate our approach, we have performed experiments on the two main components of our system, namely the *Concept Extractor* and the *Relation Identifier*. The results of these experiments will be described in what follows.

4.1. Extracting related concepts

We tested the *Concept Extractor* module on several types of queries, including: simple-concept queries, name queries, set-based queries, spelling-error queries and also word sense disambiguation (WSD) queries. Table 1 shows the results of the experiments performed. For simple concept queries, such as *Sun Microsystems* or *Flu*, the system found the most closely related concepts and listed them according to their relatedness rank. The system was also able to find concepts related to a name query. Thus, when provided with the query *Dan Brown* (a well known author), the system identified related concepts, such as the *author* concept and the names of Brown's most famous fiction books. Furthermore, our system was also able to disambiguate the meaning of an ambiguous concept, *Leopard*, based on its neighboring terms. We also provided a misspelled term as a query and, as expected, the system was able to handle the error and returned concepts related to the corrected query. Finally, the system was tested on a query formed with similar keywords belonging to a single specific domain. The purpose of this experiment was to simulate the behavior of Google Sets. For example, for the query *Taurus Leo Aries*, the common domain is "Astrological Signs". Our system returned the names of other astrological signs, from the same domain.

4.2. Identifying relationships

As described above, the concepts extracted by the *Concept Extractor* module, together with their corresponding query concept, are used to form pairs of concepts of the form *<query concept, related concept>*. Such pairs are provided as input to the *Relationship Identifier* module. The results produced by this module are displayed in Table 2. Due to space limitations, we display results for only four pairs of concepts. In the table, *Subject* refers to the *query concept* within a pair and *Object* refers to its *related concept* in the pair. *Subjects* are linked to *Object* through relationships called *Predicates*. As the results in Table 2 show, our approach is able to determine accurate relationships between *Subjects* and *Objects*. These relationships are not limited to synonyms, hypernyms and hyponyms. More complex relationships can be identified as well.

4.3. Building a semantic relationship graph

The *Graph Builder* module takes as input the query and its related concepts, together with the corresponding relationships, and it displays them as a graph. The graph obtained for the query *Harry Potter* is shown in Figure 4. The user can explore the graph by clicking on a particular

Query Type	Query	Extracted Concepts
General	Sun Microsystems	Java, OpenOffice, Sparc, Unix, Jini, Server, Solaris, Training
	Kansas State University	K-State, Kansas State, CIS, Ahearn field house , Kansas, Powercat, Courses
	Flu	Influenza, Flu shot, Oseltamivir, Avian influenza, Hong Kong flu, Cold, Acetaminophen
	Italy	Rome, Venice, Abruzzo, Naples, Tuscany, Umbria, Coat of arms of italy
Name-based	Harry Potter	Daniel Radcliffe, Rupert Grint, Emma Watson, Hermione granger, Hogwarts, JK Rowling
	Dan Brown	Author, Da vinci code, Angels and Demons, Biography, Deception point
WSD	Leopard OS	Apple, Mac OS X, Mac OS History, Operating System, 32-bit, 64-bit, Software
	Leopard Animal	Animals, Leopard, Snow Leopard, Mammals. Wildlife, Aardwolf
Set-based	Taurus Leo Aries	Aries, Taurus, Astrology, Gemini, Cancer, Pisces, Virgo, Capricorn
Spelling-error	Contectiivtis	Conjunctivitis, Pinkeye, Allergic conjunctivitis, Chlamydia, Eye infection, Infectious, Bacterial conjunctivitis

Table 1: CE results: related concepts extracted for a given query.

Subject	Predicate	Object	Subject	Predicate	Object
Sun Microsystems	programming languages	Java	Flu	synonym	Influenza
Sun Microsystems	microprocessors	Sparc	Flu	immunizations	Flu shot
Sun Microsystems	computer networking protocols	Jini	Flu	immunizations	Oseltamivir
Sun Microsystems	software	OpenOffice	Flu	super type	Avian Influenza
Sun Microsystems	computer hardware	Server	Flu	super type	Hong Kong flu
Sun Microsystems	unix	Solaris	Flu	diseases and conditions	Cold
Sun Microsystems	computer training	Training	Flu	drugs and medications	Acetaminophen
Subject	**Predicate**	**Object**	**Subject**	**Predicate**	**Object**
Kansas State University	synonym	K-state	Harry Potter	english actor	Daniel Radcliffe
Kansas State University	synonym	Kansas state	Harry Potter	english actor	Rupert Grint
Kansas State University	departments and programs	CIS	Harry Potter	fictional character	Albus dumbeldore
Kansas State University	athletics	Ahearn field house	Harry Potter	british author	JK Rowling
Kansas State University	manhattan	Kansas	Harry Potter	setting	Hogwarts
Kansas State University	athletics	Powercat	Harry Potter	fictional character	Hermione granger
Kansas State University	departments and programs	Courses	Harry Potter	english actor	Emma Watson

Table 2: RI results: relationships (predicates) determined between query concepts (subjects) and the related concepts (objects).

surrounding concept, in which case the clicked concept becomes the new query and the process is reiterated to build the graph centered on this query.

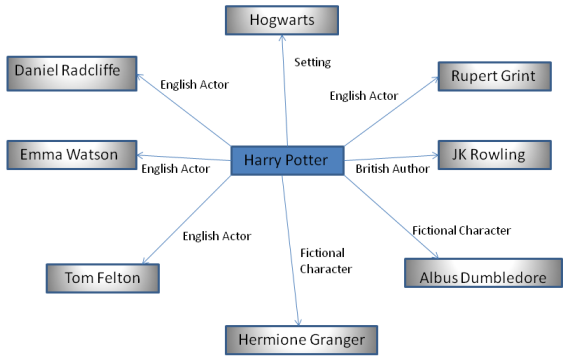

Figure 4: Graph Builder Output

4.4. Representation in RDF format

Similar to the *Graph Builder* module, the *RDF Generator* takes as input the query and its related concepts, together with their corresponding relationships, and generates an RDF database, which is saved for future use. More precisely, the generated RDF can be queried using formal query languages, such as SPARQL. For example, the system can answer queries such as: *"What English actors have acted in Harry Potter?"* or *"Who is Albus Dumbeldore?"* by converting them into triplets of the form <X,EnglishActor,Harry Potter> and <X,Y, Albus Dumbeldore>, respectively. Using the generated RDF and an inference engine, the system can infer the values of the variables.

5. Comparison with other approaches

In addition to the evaluation of the individual modules presented in the previous section, we have also compared our system's main modules, *Concept Extraction* and *Relationship Identifier*, with similar existing systems.

First, we have compared the *Concept Extractor* module with two other similar existing "state of the art" systems, specifically ESA [8] and WikiRelate! [6], along several dimensions. The comparison is shown in Table 3. As can be seen, our *Concept Extractor* not only provides results that are comparable to the results provided by other similiar concept detection systems, but it is also able to handle some of the limitations of these approaches. The main strength of our CE approach is that it is not dependent on a knowledge base. Instead, information from the whole web is used to

identify related concepts. Therefore, in theory, it should work on an "infinite" vocabulary. Moreover, as the results show, CE is capable of handling errors in the user query. It also has the capability to generate results for many name queries, compared with other concept detection algorithms that cannot do that, due to their limited vocabulary (e.g., names described in Wikipedia).

Criteria	**CE**	**ESA**	**Wiki!**
Knowledge Base Specific	No	Yes	Yes
Error Correction	Yes	No	No
Length of Query	Multiple	Multiple	Single
Handle Name Queries	Almost all	Few	Few

Table 3: Comparison of the CE module with ESA and Wiki!

Second, we have compared the performance of the *Relationship Identifier* module to WikiC!, a Wikipedia based relationship extraction system [28]. The comparison was performed using several queries from [28] and several new queries. The results are shown in Table 4.

Concept Pairs	**RI**	**WikiC!**
<Apple, Fruit>	is	is
<Cat, Mammal>	is	is
<Bird, Biped>	no relation	is
< Computer, Machine>	is	is
<Jimmy Snuka, Wrestler>	is	is
<Colorado, U.S. states >	is in	is one of
<Sharon Stone, Model>	actor	is
<Flintstones, Animated TV Show>	is	no relation
<Pat Benatar, 1980's music>	rock and pop	no relation

Table 4: Comparison between relationships found by our RI module and relationshipds found by WikiC!

As can be seen, in many cases, RI is able to identify relationships similiar to those identified by WikiC! RI could not find a relation for the <Bird, Biped> pair (when searching for *Biped* the first sentence in the corresponding Wikipedia document did not have have enough information to help identify a relationship and the concept was not present in Yahoo! Directories). However, RI outperforms WikiC! when at least one of the concepts in the pair considered is not present on Wikipedia. For example, considering the concepts *Flintstones* and *Animated TV Show*, we can find a Wikipedia document for *Flintstones*, but not for *Animated TV Show*. Hence, in this case WikiC! cannot return a relationship. As *Animated TV Show* is present in Yahoo! Directories, RI returns a relationship as *Flintstones is Animated TV Show*. As another example, consider the terms: *Pat Benatar* and *1980's music*. Wikipedia has no article for *1980's music*, and hence WikiC! cannot return a relationship. However, RI returns the relationship as *rock and pop*. Such examples show that there are cases when our relationship identifier algorithm outperforms other similiar relation detection algorithms.

6. Summary and Discussion

In this paper, we have proposed a two-phase approach towards bridging the gap between the Web and Semantic Web. Our approach makes use of the power of existing search engines to extract concepts related to a query concept, and exploits well known knowledge bases to determine relationships between the query and the extracted concepts. We have evaluated our approach by comparing it with some of the existing "state of the art" approaches that use Wikipedia as knowledge base for identifying concepts and relationships between concepts. Our approach has the following advantages:

- The concept extraction module is not specific to a particular knowledge base, such as Wikipedia or WordNet; the whole web is used as knowledge base.
- Spelling mistakes made by the user can be easily handled. Also, there is no limit on the length of the query to be fed in by the user.
- The inclusion of PageRank scores in the calculation of weights of concepts makes our approach resistant against techniques like "keyword stuffing", which are sometimes employed by web pages to achieve a higher rank in search results.
- The system can identify not only relationships between strongly related concepts, but also relationships between weakly (or loosely) related concepts.
- The graph representation of the results makes it easy for the users to interpret the results.
- Storage of machine interpretable RDF triplets facilitates further querying using inference engines.
- Although simple, our approach provides high quality results comparable to the results of other "state of the art" approaches.

While our approach can use category graphs in an effective way to determine relationships between two concepts, it also has some drawbacks. The major drawback is that the relationships identified depend heavily on the phrases used in the category graph structure. Hence, in some cases, the identified "relationship" phrase might loosely capture the relationship between the two concepts. For example, when trying to determine a relationship between *Sparc* and *Sun Microsystems*, RI returns the relationship *microprocessors*. This is because the *microprocessors* term is present as a category name in the category graph on Yahoo! Directories. However, the precise relationship is that *Sparc* is a *processor architecture* designed by *Sun Microsystems*. Hence, in this case, our system is able to capture a loose relationship, but not an exact relationship. More precise relationships can be identified by exploiting all the information present in Wikipedia documents (as opposed to only the first sentence). The use of search engines such as DBPedia (dbpedia.org) and Powerset (www.powerset.com) can be used to extract

better information from Wikipedia and might help improve our RI results.

Future work plans include more experimentation with the CE-RI system to identify the best settings for parameters. Furthermore, we will also explore languages for querying the RDF database and compare the results with the results of the traditional search engines. Finally, along with the identification of a relationship, we also aim to assign a statistical score to the query concept and its related concepts in order to determine their degree of semantic relatedness.

Acknowledgements

This work was funded by the National Science Foundation grant number NSF 0711396 to Doina Caragea.

References

[1] E. Minack, W. Siberski, and G. Zenz, "Suits4rdf: Incremental query construction for the semantic web," in *Proc. of the Int. Semantic Web Conf. - Posters and Demos*, ser. CEUR Workshop Proceedings, vol. 401, 2008.

[2] M. Jarmasz and S. Szpakowicz, "Rogets thesaurus and semantic similarity," in *Proc. of RANLP-03*, 2003, pp. 212–219.

[3] Z. Wu and M. Palmer, "Verb semantics and lexical selection," in *Proc. of ACL-94*, 1994.

[4] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, S. Solan, G. Wolfman, and E. Ruppin, "Placing search in context: The concept revisited," *ACM Transactions on Information Systems*, vol. 20, no. 1, pp. 116–131, 2002.

[5] S. Banerjee and T. Pedersen, "Extended gloss overlap as a measure of semantic relatedness," in *Proc. of the Int. Joint Conf. on AI (IJCAI-03)*, 2003, pp. 805–810.

[6] M. Strube and S. Ponzetto, "Exploiting semantic role labeling, wordnet and wikipedia for coreference resolution," in *Proc. of the ACL Conf. on HLT*, NJ, USA, 2005.

[7] Z. Syed, T. Finin, and A. Joshi, "Wikipedia as an ontology for describing documents," in *Proc. of 2nd Int. Conf. on Weblogs and Social Media*, 2008.

[8] E. Gabrilovich and S. Markovitch, "Computing semantic relatedness using wikipedia-based explicit semantic analysis," in *Proc. of the Int. Joint Conf. on AI (IJCAI-07)*, 2007, pp. 1606–1611.

[9] D. Milne, "Computing semantic relatedness using wikipedia link structure," in *Proc. of the New Zealand CS Research Student Conf. (NZCSRSC'07)*, Hamilton, New Zealand, 2007.

[10] C. Buckley, G. Salton, J. Allan, and A. Singhal, "Automatic query expansion using smart: Trec 3," in *Proc. of 3rd Conf. on Text Retrieval*, 1994, pp. 69–80.

[11] M. Mitra, A. Singhal, and C. Buckley, "Improving automatic query expansion," in *Proc. of 21st Annual Int. ACM-SIGIR Conf. on Research and Development in Information Retrieval*, 1998, pp. 206–214.

[12] B. Vlez, R. Wiess, M. Sheldon, and D. Giord, "Fast and effective query refinement," in *Proc. of 20th Annual Int. ACM-SIGIR Conf. on Research and Development in Information Retrieval*, 1997, pp. 6–15.

[13] P. Mika, "Ontologies are us: A unified model of social networks and semantics," in *Proc. of the ISWC*, 2005.

[14] Y. Matsuo, J. Mori, M. Hamasaki, K. Ishida, T. Nishimura, H. Takeda, K. Hasida, and M. Ishizuka, "Polyphonet: An advanced social network extraction system," in *Proc. of 15th Int. WWW Conf.*, 2006.

[15] R. Rosenfield, "A maximum entropy approach to adaptive statistical modelling," in *Computer Speech and Language*, 1996.

[16] P. Resnik, "Semantic similarity in a taxonomy: An information based measure and its application to problems of ambiguity in natural language," in *Journal of Aritificial Intelligence Research*, 1999.

[17] D. Lin, "Automatic retreival and clustering of similar words," in *Proc. of the 17th COLING*, 1998.

[18] J. Jiang and D. Conrath, "Semantic similarity based on corpus statistics and lexical taxonomy," in *Proc. of the Int. Conf. on Research in Computational Linguistics ROCLING X*, 1998.

[19] D. Lin, "An information-theoretic denition of similarity," in *Proc. of the 15th Int. Conf. on ML*, 1998, pp. 296–304.

[20] X. Sun, Q. Zheng, H. Dang, Y. Hu, and H. Bai, "An approach to acquire semantic relationships between words from web document," in *Lecture notes in Computer Science*, 2007.

[21] P. D. Turney, "Minning the web for synonyms: Pmi-ir versus lsa on toefl," in *Proc. of ECML*, 2001.

[22] Y. Matsuo, T. Sakaki, K. Uchiyama, and M. Ishizuka, "Graph-based word clustering using web search engine," in *Proc. of EMNLP*, 2006.

[23] M. Sahami and T. Heilman, "A web-based kernel function for measuring the similarity of short text snippets," in *Proc. of 15th Int. WWW Conf.*, 2006.

[24] D. Bollegala, Y. Matsuo, and M. Ishizuka, "Measuring semantic similarity between words using web search engines," in *Proc. of the Int. WWW Conf.*, 2007.

[25] H. Chen, M. Lin, and Y. Wei, "Novel association measures using web search with double checking," in *Proc. of the COLING/ACL 2006*, 2006.

[26] S. Angeletou, M. Sabou, L. Specia, and E. Motta, "Bridging the gap between folksonomies and the semantic web," in *Proc. of the Workshop on Bridging the Gap between Semantic Web and Web 2.0 at the 4th ESWC*, 2007.

[27] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web." Stanford InfoLab, Technical Report 1999-66, 1999.

[28] K. Nakayama, T. Hara, and S. Nishio, "Wikipedia link structure and text mining for semantic relation extraction," in *Proc. of the Workshop on Semantic Search at the 5th ESWC*, vol. 334, 2008.