International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:8, No:5, 2014

Categorizing Search Result Records Using Word Sense Disambiguation

R. Babisaraswathi, N. Shanthi, S. S. Kiruthika

***Abstract*—**Web search engines are designed to retrieve and extract the information in the web databases and to return dynamic web pages. The Semantic Web is an extension of the current web in which it includes semantic content in web pages. The main goal of semantic web is to promote the quality of the current web by changing its contents into machine understandable form. Therefore, the milestone of semantic web is to have semantic level information in the web. Nowadays, people use different keyword- based search engines to find the relevant information they need from the web. But many of the words are polysemous. When these words are used to query a search engine, it displays the Search Result Records (SRRs) with different meanings. The SRRs with similar meanings are grouped together based on Word Sense Disambiguation (WSD). In addition to that semantic annotation is also performed to improve the efficiency of search result records. Semantic Annotation is the process of adding the semantic metadata to web resources. Thus the grouped SRRs are annotated and generate a summary which describes the information in SRRs. But the automatic semantic annotation is a significant challenge in the semantic web. Here ontology and knowledge based representation are used to annotate the web pages.

***Keywords*—**Ontology, Semantic Web, WordNet, Word Sense Disambiguation.

I. Introduction

A large volume of data available in the web is in unstructured or semi-structured format. Thus the data in the web is intended to be browsed by humans, not by machines. Most of the users use keyword-based search engines to retrieve information from the web. Since there is a dramatic increase in the web pages keyword-based search engines cannot help the users to find the most relevant and accurate information in an efficient way. The semantic web is an extension of the World Wide Web in which the information is given with well-defined meaning [8]. The idea of semantic web is leave the tasks and decisions to machines, so that machines and people to work in co-operation. This is applicable by adding knowledge to the information available in the web by understanding which language the machine can understand and introduce the software agents that able to process the information [7].

The machines in the semantic based environment have a common understanding from metadata tags and communicate to each other. In order to communicate with each other there is a need for repository which can define all the concepts. In semantic web, ontology acts as a shared repository. An ontology is an explicit, formal specification of a shared conceptualization. This means that an ontology describes the information in a machine understandable way. In other words, ontology is to be considered as a tool which defines additional meaningful tags to web pages and makes them available to be used by software agents and other applications. In the semantic based environment the resources are not only accessible by humans but also to automated processes. The automation of tasks elevates the status of the web from machine-readable to machine understandable. The data on the web is explicitly interpreted by the software agents rather than implicitly interpreted by humans. In order to realize this, there is a need to associate metadata with the resources. One such mechanism to associate the metadata with the resources is annotation.

The rest of the paper is organized as follows; Section II describes what word sense disambiguation algorithm is. Section III discusses about the accessing the web data using semantic annotation. Section IV discusses about the proposed method. Section V describes the conclusion and future work.

II. Word Sense Disambiguation

Word sense disambiguation is the process of identifying all possible meanings of a given keyword. It also includes the list of meanings such as those found in dictionaries and thesauri. WSD is a natural classification problem. The well-known and shared thesaurus provides a reliable set of meanings and allows the users to share with others the result of the disambiguation process. Moreover, the fundamental peculiarity of a thesaurus is the presence of a wide network of relationships between words and meanings.

Most of the words have more than one meaning. So, the user wants to disambiguate it. For example, consider a single query keyword “java”. Java has two senses. (1) It is an island in Indonesia (2) a platform-independent object oriented programming language. For mouse, there are two senses namely: (1) a small rodent (2) a computer mouse. Likewise most of the user keywords have different senses; the user wants to differentiate it. In the search results the results are mixed up with different meanings [6]. The users find it difficult to extract the information what they searched for. In order to disambiguate the different meanings of the search results WSD is used. In WSD approach WordNet is used to find the possible senses of the user keywords.

R. Babisarawathi is with the K.S.Rangasamy College of Technology, India (phone: +919865907718 e-mail: babisaraswathi@gmail.com).
Dr. N. Shanthi is with the Nandha Engineering College, India (e-mail: shanthimoorthi@gmail.com).
S. S. Kiruthika is with the Computer Science and Engineering, K. S. Rangasamy College Of Technology, India (phone: +918508293761 e-mail: sskiruthicse@gmail.com).

III. Semantic Annotation

The process of adding notes or comments to any resource is known as annotation. In the domain of web, annotation means adding the information like notes, comments or summary to an existing resource like documents without changing their original contents. These annotations are sharable over the network. The notes, comments and summaries are not sharable but it is useful. Semantic annotation means adding machine understandable metadata to resources. Basically, there are three types of annotation, namely: 1) informal annotations do not have formally defined constituents and not machine understandable 2) formal annotations are machine-readable but do not use ontological terms 3) ontological annotations have formally defined constituents and use ontological terms that are socially accepted and understood. Annotation can be done manually and automatically [2]. Manual annotation can be accomplished easily using the authoring tools such as Semantic Word which provides the integrated environment for simultaneously authoring and annotating the text. Manual annotation is an expensive process.

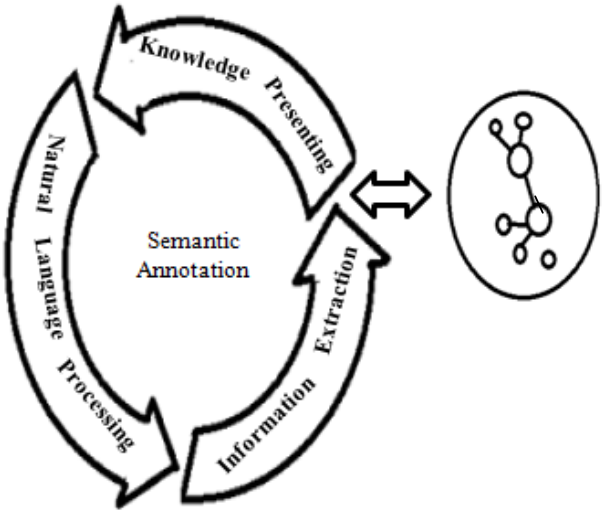

Fig. 1 An overview of semantic annotation and effective technologies

There are two reasons for an ineffectiveness of manual annotation namely: 1) Due to large amount of tasks and resources it is time consuming. 2) Different opinions can result in inconsistent knowledge [1].

The key element of semantic annotation is ontology. Ontological structures give additional value to semantic annotation. Documents are annotated with concept instances from the Knowledge Base by creating instances of the *Annotation* class, provided for this purpose [3]. *Annotation* has two relational properties namely, *instance* and *document*, by which concepts and documents are related together [3].

IV. Proposed System

The main goal of our system is to provide users with data that satisfy their information needs with little effort, even when they are looking for information not popular on the web. The proposed model consists of five modules:

- Downloading html pages using yahoo search engine
- Extraction of Snippet and URL
- WSD using WordNet
- Categorization of SRRs

A. Downloading HTML Pages using Yahoo Search Engine

In the first module, write a simple java program that connects to a yahoo web server and downloads web page and documents that the user is interested in. The java.net package contains URL and URL Connection classes to download the data and content from the Internet servers. There are many applications which makes use of client-side network programming for example writing web spiders or crawlers, robots to check and verify links on a website, web browsers, search engines, a tool to download a complete website or writing a simple tool to download webpages and parse it for further data analysis [9].

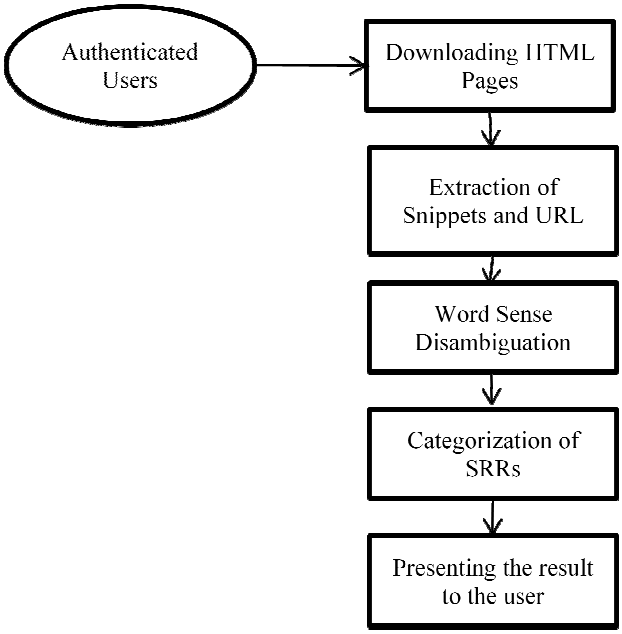

Fig. 2 Overview of the Proposed System

B. Extraction of Snippets and URL

For a user query, there are number of results will be retrieved and the results should be informative to the user. The users will not want to examine all the returned results. So we want to make the results list informative enough that the user can do a final ranking of the documents for themselves based on relevance to their information need. The standard way of doing this is to provide a snippet. Snippet is a short summary of the document, which is designed so as to allow the user to decide its relevance [10]. Typically, a snippet consists of a

document title and a short summary which is automatically extracted. The two basic kinds of summaries are static, which are always the same regardless of the query, and dynamic (or query-dependent), which are customized according to the user's information need as deduced from a query [10].

Fig. 3 Structure of a snippet

Generating snippets must be fast since the system is typically generating many snippets for each query that it handles. Rather than caching an entire document, it is common to cache only a generous but fixed size prefix of the document, such as perhaps 10,000 characters. For most common, short documents, the entire document is thus cached, but huge amounts of local storage will not be wasted on potentially vast documents.

In this phase, the snippets and the corresponding URL will be extracted using java program. Here, a parser is used to extract the snippet. The parser is a java library for working with real-world HTML. It provides a very convenient Application Programming Interface (API) for extracting and manipulating data, using the best of DOM, CSS, and jquery like methods[11]. The content in the web page is converted into string and then the snippet is extracted. Using the splt() function in String Class to split the strings and Pattern, Matcher class is used to find the URL, so that both the URL and snippets are extracted and stored in the database tbl_snippet.

Stop words are common words that do not have so much meaning in a retrieval system. Stop words are words which are filtered out prior or after processing the data. The reason that stop words should be removed from a text is that they make the text look heavier and less important for analysts and the stop words are not necessary for the analysis and so we do get some data reduction by eliminating stop words. A query done by using stop words would have a weak ability to categorize the text because of these words return each element of the data set as a result. There is no definite set of stop words which all tools use. Any set of words can be chosen as stop words for the given purpose. For some search machines, these are some of the most common, short function words, such as the, is, at, which, and on. In this case, stop words can cause problems when searching for phrases that include them, particularly in names such as The, Who, This or That. Other search engines remove some of the most common words including lexical words, such as "want" from a query in order to improve performance [4].

C. WSD Using WordNet

The goal of this step is to provide the user with the hits retrieved by a traditional search engine, such as Google or Yahoo!, classified in the categories defined by the different senses of the user keywords. Moreover, the categories will be ranked according to the user's interest. At run-time there are four phases for performing this process. At first, this phase requires performing a traditional search of hits on the web, by taking user query as input and using a traditional search engine such as Google or Yahoo!. This search returns a set of relevant ranked hits, with the web pages which contains the user keywords. The order of the hits (i.e., the ranking of the results provided by the search engine used) depends on the specific techniques used by the particular search engine for that task and its various internal parameters. Then, the hits returned by the search engine are given as input to the next phase incrementally, in blocks of hits of a certain size. In this way, number of new hits can be retrieved while the first blocks are being processed. Our prototype uses blocks of 100hits but the block size is a configuration parameter. This process can be performed in parallel with the Discovery of the Semantics of User Keywords step.

Each hit obtained from the previous phase (contains a title, a URL and a snippet) is automatically annotated lexically. Thus, firstly, each hit H_j goes through a cleansing process where stop words are filtered out (creating the filtered hit H'_j). After that, the relevant words of the title and the snippet of each filtered hit H'_j are considered to perform a lexical annotation of the hit. A lexical annotation is a piece of information added to a term that refers to a semantic knowledge resource such as a dictionary, a thesaurus, a semantic network, or any other resource which expresses, either implicitly or explicitly, a general ontology of the world or a specific domain[4].

Here, WordNet ontology is used to annotate the hits. The user keyword is given to WordNet which finds all the possible meanings and the senses of the corresponding meaning. By using this, the URLs and snippets are annotated and grouped into categories. First, based on the URLs the categories will be defined. In this the duplicates of the URLs are removed in the first step. After that find the URL starts with http:// or www. The URL which contains the user keywords are extracted and stored in a separate database. Next, the URL which are not in the http:// or www format will be annotated using the snippet.

For that the possible combination of user keywords or classes associated with it are identified for different meanings of the user keywords. Each and every combination of user keywords are used to search whether it can be find in the snippet or URL. If it is found that the corresponding snippet and URL are grouped into a category with the common name (i.e. the corresponding combination of user keywords).

The above processes are written in a java program using Riwordnet jar file. Some of the methods in Riwordnet are used in the program for annotation. The methods used are *getPos(), getSenseIds(), getDescription(), getSynset(), getBestPos() and getAllHyponyms()*.

For example, java has two different meanings. First one is an island in Indonesia and the second is a platform-independent object oriented programming language.

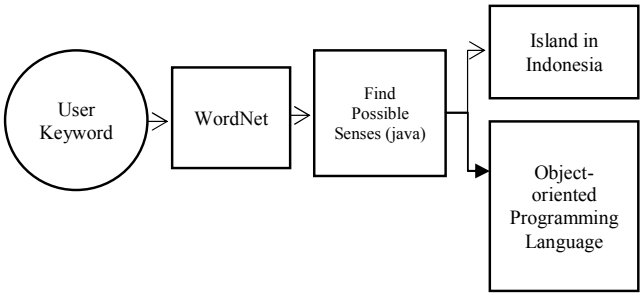

Fig. 4 Identifying different senses of java using WordNet

Here, yahoo search engine is used to retrieve information. For a single page there are 10 SRRs are given in yahoo. The following table denotes that the first page all SRRs are related to the second sense of java. In the second page 9 SRRs related to the second sense of java and 1 SRR related to the first sense of java.

TABLE I
NO. OF SRR'S FOR EACH SENSE OF A USER KEYWORD

HTML File	java-programming language (out of 10)	java island (out of 10)
File_1	10	0
File_11	9	1
File_12	10	0
File_13	7	3
File_14	10	0
File_15	9	1
File_16	10	0
File_17	10	0
File_18	9	1
File_19	10	0

Likewise for the first 10 pages graph is constructed for the possible senses of the user keyword 'java'.

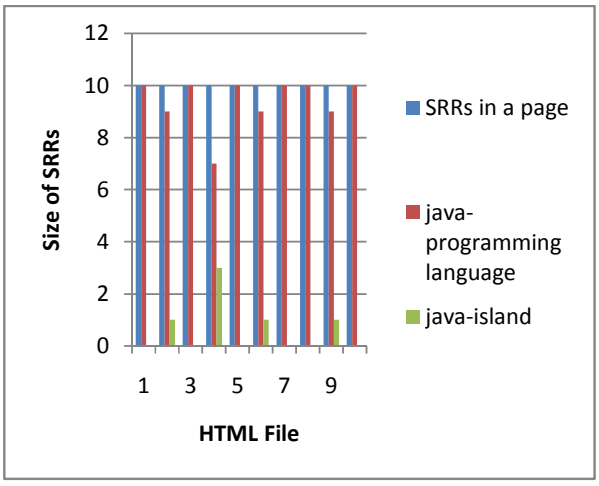

Fig. 5 Different senses of java

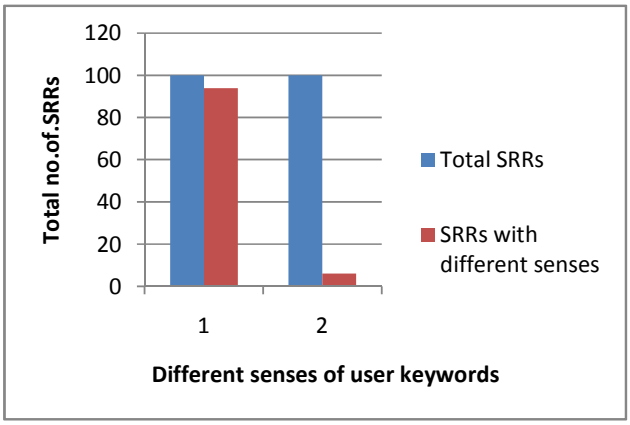

Fig. 6 No. of SRRs for each sense of user keywords

The x-axis denotes the html file which contains the search results The y-axis denotes the size of SRRs means that number of search result records in a web page..

From the above graph we realize that, if the user wants to find java island then they want to see every page. But the results related to it are very less. If the results are grouped based on the senses then it is easy for the users to find the relevant information.

Here, the second sense of java has more SRRs. Among those SRRs, some of the results are blogs and some of the links are repeated. After WSD process, based on the URL the search results are grouped based on the domain name.

D. Categorization of SRRs

In this phase, the hits (already annotated as a result of the previous process) are grouped into categories by considering their lexical annotations. Firstly, the system defines the possible categories that are going to be considered. Then, blocks of hits are classified. The potential categories are defined by considering all the possible combinations of possible keyword senses of the input keywords (i.e., the Cartesian product of the possible sense sets of the user keywords) [5]. Finally, the results of the Categorization of Hits phase are presented to the user.

V. CONCLUSION

This approach improves the effectiveness of searching, especially when users are searching for information which is not the most popular on the web. However, the user may still need to invest some effort to locate the hits relevant to his/her query, even when browsing the hits within the correct category (i.e., when the semantics of the keywords in both the hits and the user keywords have been identified)[4]. Thus, the user could be looking for hits corresponding to several different queries within a category. The search results returned by the traditional search engines are annotated by using lexical resources such as thesaurus and ontology. Semantic annotation automatically identifies the data items that are related and also identifies the relationship between the data. The annotated hits can be grouped by the semantic of the data which has a browsable summary describes what data the categorized hit

contains. By visualizing that summary the users can view the hits containing the relevant information of the given keywords.

Thus, we have adapted our work in other areas such as ontology matching and probabilistic word sense disambiguation for data integration systems to the context of web information retrieval. Moreover, current approaches do not deal with compound nouns (such as ‘‘creditcard’’) and proper nouns (such as ‘‘TowerBridge’’), that could appear in the title and/or the snippet of the retrieved hits. So, techniques that support the disambiguation of compound nouns and named entity recognition techniques will be adopted. In particular, we will study the possibility to apply or adapt the approach for compound nouns interpretation that we proposed in the context of schema matching.

REFERENCES

[1] Benjamin Donz., Dietmar Bruckner (2012), “External Semantic Annotation of Web-Databases”, IEEE Digital Library, pp 841-845.
[2] Fernando Gomez (2006), “Automatic semantic annotation of texts”, University of Central Florida, Orlando, FL 32816.
[3] Nadzeya Kiyavitskaya., Nicola Zeni., James R.Cordy., Luisa Mich and John Mylopoulos (2006), “Semi-Automatic Semantic Annotation for Web Documents”.
[4] Raquel Trillo., Laura Po.,Sergio Ilarri., Sonia Bergamaschi and Eduardo Mena (2011), “Using semantic techniques to access web data ”, Elsevier Information Systems, Vol.No.36, pp 117-133.
[5] Yiyao Lu., Hai He., Hongkun Zhao., Weiyi Meng and Clement Yu (2013), “Annotating search results from web databases ”, IEEE Transaction on Knowledge and Data Engineering, Vol.No.25, pp 514-527.
[6] Raquel Trillo., Jorge Gracia., Mauricio Espinoza and Eduardo Mena (2007), ―Discovering the semantics of user keywords”, Journal on Universal Computer Science, Vol.13, No.12, pp 1908–1935.
[7] https://answers.yahoo.com/question/index?qid=20091014064541AAd1CDt
[8] http://www.w3.org/RDF/Metalog/docs/sw-easy
[9] http://www.netcluesoft.com/downloading-webpages-and-html.html
[10] http://nlp.stanford.edu/IR-book/html/htmledition/results-snippets-1.html
[11] http://javarevisited.blogspot.in/2014/09/how-to-parse-html-file-in-java-jsoup-example.html

R. Babisaraswathi received Master of Engineering sdegree in Computer Science and Engineering from Anna University, Chennai. Now, she is working as an Associate Professor in K.S.Rangasamy College of Technology, Namakkal. Her research interest is Semantic Web and Ontology.

N. Shanthi received her doctorate in Computer Science and Engineering from Anna University, Chennai. Now, she is working as a Dean/CSE in Nandha Engineering College, Erode.

S. S. Kiruthika received her Bachelor of Engineering degree in Computer Science and Engineering from Anna University, Chennai. She is completed Master of Engineering in K.S.Rangasamy College of Technology, Tiruchengode. Now, she is working as an Assistant Professor in Sri Krishna College of Technology, Coimbatore.