The Process of Information Extraction through Natural Language Processing

S Acharya sandigdhaacharya@yahoo.co.in
Lecturer/CSE Deptt,
Synergy Institute of Technology
Bhubaneswar,Orissa,India,752101

S Parija smita.parija@gmail.com
Asst.Prof/ECE Deptt,
Synergy Institute of Technology.
Bhubaneswar, Orissa, India, 752101

Abstract

Information Retrieval (IR) is the discipline that deals with retrieval of unstructured data, especially textual documents, in response to a query or topic statement, which may itself be unstructured, e.g., a sentence or even another document, or which may be structured, e.g., a Boolean expression. The need for effective methods of automated IR has grown in importance because of the tremendous explosion in the amount of unstructured data, both internal, corporate document collections, and the immense and growing number of document sources on the Internet.. The topics covered include: formulation of structured and unstructured queries and topic statements, indexing (including term weighting) of document collections, methods for computing the similarity of queries and documents, classification and routing of documents in an incoming stream to users on the basis of topic or need statements, clustering of document collections on the basis of language or topic, and statistical, probabilistic, and semantic methods of analyzing and retrieving documents. Information extraction from text has therefore been pursued actively as an attempt to present knowledge from published material in a computer readable format. An automated extraction tool would not only save time and efforts, but also pave way to discover hitherto unknown information implicitly conveyed in this paper. Work in this area has focused on extracting a wide range of information such as chromosomal location of genes, protein functional information, associating genes by functional relevance and relationships between entities of interest. While clinical records provide a semi-structured, technically rich data source for mining information, the publications, in their unstructured format pose a greater challenge, addressed by many approaches.

Keywords: Natural language Processing(NLP),Information retrieval, Text Zoning

1. INTRODUCTION

Natural Language Processing (NLP) [1]is the computerized approach to analyzing text that is based on both a set of theories and a set of technologies, and being a very active area of research and development, there is not a single agreed-upon definition that would satisfy everyone, but there are some aspects, which would be part of any knowledgeable person's definition.

Definition: Natural Language Processing is a theoretically motivated range of computational techniques for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis for the purpose of achieving human-like language processing for a range of tasks or applications. Several elements of this definition can be further detailed. Firstly the imprecise notion of 'range of computational techniques' is necessary because there are multiple methods or techniques from which to choose to accomplish a particular type of language analysis. 'Naturally occurring texts' can be of any language, mode, genre, etc. The texts can be oral or written. The only requirement is that they be in a language used by humans to communicate to one another. Also, the text being analyzed should not be specifically constructed for the purpose of the analysis, but rather that the text be gathered from actual usage.

The notion of 'levels of linguistic analysis' (to be further explained in Section 2) refers to the fact that there are multiple types of language processing known to be at work when humans produce or comprehend language. It is thought that humans normally utilize all of these levels since each level conveys different types of meaning. But various NLP systems utilize different levels, or combinations of levels of linguistic analysis, and this is seen in the differences amongst various NLP applications. This also leads to much confusion on the part of non-specialists as to what NLP really is, because a system that uses any subset of these levels of analysis can be said to be an NLP-based system. The difference between them, therefore, may actually be whether the system uses 'weak' NLP or 'strong' NLP. 'Human-like language processing' reveals that NLP is considered a discipline within Artificial Intelligence (AI). And while the full lineage of NLP does depend on a number of other disciplines, since NLP strives for human-like performance, it is appropriate to consider it an AI discipline. 'For a range of tasks or applications' points out that NLP is not usually considered a goal in and of itself, except perhaps for AI researchers. For others, NLP is the means for 1 Liddy, E. D. In Encyclopedia of Library and Information Science, 2nd Ed. Marcel Decker, Inc. accomplishing a particular task. Therefore, you have Information Retrieval (IR) systems that utilize NLP, as well as Machine Translation (MT), Question-Answering, etc. The goal of NLP as stated above is "to accomplish human-like language processing". The choice of the word 'processing' is very deliberate, and should not be replaced with 'understanding'. For although the field of NLP was originally referred to as Natural Language Understanding (NLU) in the early days of AI, it is well agreed today that while the goal of NLP is true NLU, that goal has not yet been accomplished. A full NLU System would be able to:

1. Paraphrase an input text
2. Translate the text into another language
3. Answer questions about the contents of the text
4. Draw inferences from the text

While NLP has made serious inroads into accomplishing goals 1 to 3, the fact that NLPsystems cannot, of themselves, draw inferences from text, NLU still remains the goal of NLP. There are more practical goals for NLP, many related to the particular application for which it is being utilized. For example, an NLP-based IR system has the goal of providing more precise, complete information in response to a user's real information need. The goal of the NLP system here is to represent the true meaning and intent of the user's query, which can be expressed as naturally in everyday language as if they were speaking to a reference librarian. Also, the contents of the documents that are being searched will be represented at all their levels of meaning so that a true match between need and response can be found, no matter how either are expressed in their surface form.

2. INFORMATION EXTRACTION

What is Information Extraction?

This volume takes a broad view of information extraction [2] as any method for filtering information from large volumes of text. This includes the retrieval of documents from collections and the tagging of particular terms in text. In this paper we shall use a narrower definition: the identification of instances of a particular class of events or relationships in a natural language text, and the extraction of the relevant arguments of the event or relationship. Information extraction there- fore involves the creation of a structured representation (such as a data base) of selected information drawn from the text.

The idea of reducing the information in a document to a tabular structure is not new. Its feasibility for sublanguage texts was suggested by Zellig Harris in the 1950's, and an early implementation for medical texts was done at New York University by Naomi Sager [5]. However, the specific notion of information extraction described here has received wide currency over the last decade through the series of Message Understanding Conferences [1, 2, 3, 4, 14].We shall discuss these Conferences in more detail a bit later, and shall use simplified versions of extraction tasks from these Conferences as examples throughout this paper the type of attack (bombing, arson, etc.), the date, location, perpetrator (if stated), targets, and effects on targets. Other examples of extraction tasks are international joint ventures (where the arguments included the partners, the new venture, its product or service, etc.) and executive succession (indicating who was hired or _red by which company for which position).

Information extraction is a more limited task than \full text understanding". In full text understanding, we aspire to represent in a explicit fashion all the information in a text. In contrast, in information extraction we delimit in advance, as part of the specification of the task, the semantic range of the output: the relations we will represent, and the allowable _leers in each slot of a relation. Identify specific pieces of information (data) in a unstructured or semi-structured textual document. Transform unstructured information in a corpus of documents or web pages into a structured database.

Applied to different types of text:

–Newspaper articles
–Web pages
–Scientific articles
–Newsgroup messages
–Classified ads
–Medical notes

In many application areas of text analysis, for instance, in information retrieval and in text mining, shallow representations of texts have been recently widely used. In in- formation retrieval, such shallow representations allow for a fast analysis of the in- formation and a quick respond to the queries. In text mining, such representations are used because they are easily extracted from texts and easily analyzed. Recently in all text-oriented applications, there is a tendency to begin using more complete representations of texts than just keywords, i.e., the representations with more types of textual elements. For instance, in information retrieval, these new representations increase the precision of the results; in text mining, they ext end the kinds of discovered knowledge.Many web pages are generated automatically from an underlying database. Therefore, the HTML structure of pages is fairly specific and regular However, output is intended for human consumption, not machine interpretation. An IE system for such generated pages allows the web site to be viewed as a structured database.

2.1 Techniques of Information Retrival

2.1.1 Text Zoning

Turns a text into a set of useful text segment(like headers,paragraphs,table) May be topic based using keywords or static. Depends on the structure of the text in domain of application. Discard unwanted segment of text.

2.1.2 Preprocessing

Take as input a stream of characters carried out tokenisation & sentence segmentations(convert txt segment into a sequence of sentence,disambiguaten fullstop) Part-of-speech tagging.named entity, spelling correction has been carried out.

2.1.3 Filtering

Throws away sentences considered to be irrelevant.Primary consideration is processing time[31,32]. Relevance decision can use manually or statisticallyderived keywords.

2.1.4 Preparsing

Ingoing from a sequence of words to a parse tree, some structure can be identify more reliably than other(noun,prepositional phrases,appositives) Uses finite state grammar & special word list.

2.1.5 Name recognition entity

Name may contain unknown words Identify of names simplify parsing. IE templates slots are typically filled with name.

Temporalexpreexpression(time,date,duration).Numberexpression(number,money,measure,sped,volume,temprature,percentage,cardinal).Simpleregularexpression(postalcode,studentid,telephoneno.)Entityname (person,organization,location)

2.1.6 Parsing

Takes as inputa sequence of lexical items and smallscale structure built by the parser.Produce as output a set of parse tree fragments, correspond to subsentential unit.Goal is to detrmine the major elements in the sentence (nounphrase,verbphrase)

2.1.7 Fragment combination

Take as input a set of parse tree fragments derived from a sentences.Tries to combine fragments into a representation for he entire sentence[6].

2.1.9 Semantic interpretation

Generate a semantic structure or logical form or event frame from a parse tree or a collection of parse tree fragment. What is a semantic structure An explicit representation of the relationship between participant in sentence. Goal is to map syntatic structure into structure that encode information relevance template filling[7].

2.1.10 Lexical disambiguation

Turns a semantic structure with ambigous predicate into unambigous predicate.This task may be carried out in a number of places in a system.In restricted domains this may not be an issue –the one sense per document assumption. Only one sense of the word is used in the complete domain.

2.1.11 Coreference Resolution

- Identify different description of he same entity in different parts of text and relates them in some way. identify,meronymy.,reference to events.

- Techniques number & gender agreement for pronoun(Ram met Shyam,he later stated.) semantic consistency based on taxonomic information(toyota motor corp.”the japenese automoter”.) some notion of focus(pronoun typically refer to something mentioned in the previous sentence).

2.1.12Template generation

Derive final output templates from the semantic structures.Carries out lowlevel formatting and normalization of data.

2.1.13 Evaluations

Precision=Ncorrect/Nresponse,

Recall=Ncorrect/Nkey

F= (2*precision*recall)/(precision+recall)

3. OBSERVATIONS

3.1. Strengths of SVM

- the solution is unique
- the boundary can be determined only by its support vectors, namely SVM[33] is robust against changes of all vectors but its support vectors
- SVM is insensitive to small changes of the parameters different SV classifiers constructed by using different kernels (polynomial, RBF, neural net) extract the same support Vectors**.**

Weaknesses of SVM

- It takes more time.
- SVM is used only for categorization of documents and user feedback

3.2 Conceptual Graph:

- It is more accurate than other two model.
- structured semantic matching can improve both recall and precision
- . Each relation associated with the entry induces a subgraph
- There are many graph to derive.

3.3 Boolean Retrieval:

- Process large document quickly.
- Allow more flexible matching.
- Need invarted index files.
- Used more memory space.
- More time take to access

	Standard Boolean
Goal	• capture Conceptual structure and Contextual Information
Methods	• Coordination:AND,OR,NOT • Proximity • Fields • Stemming/Truncation
(+)	• Easy to implement • Computationally Efficient =all the major online databases use it. • Expressiveness and Clarity Synonm specifications (OR –Clauses) and phrases (AND –Clauses)
(-)	• Difficult to Construct Boolean queries • All or Nothing. ANDBtoo severe ,and OR does not differentiate Enough • .Difficult to control output:Null output↔Overload. • No Ranking. • No weighting of index or query terms. • No uncertainty measure.

FIGURE 1. Boolean Retrieval

Parameters	Boolean Retrieval	Conceptual Graph	SVM
Types of Solution	More solutions	More solutions	Unique Solutions
Data Types	Linear Documents	Linear Documents	Multidimensional data
Performance	More accurate	Most accurate than other two models	More accurate
Evaluation	Simple	Simple	More complex
Space Complexity	Uses more memory	Uses more memory	Uses less memory
Efficiency	Retrieves large documents quickly	Retrieval of information	Categorization of documents

FIGURE 2. Comparision Study.

4. SURVEY ON PAPERS:

4.1 Paper**1**

The combined use of linguistic ontologies and structured semantic matching is one of the promising ways to improve both recall and precision. In this paper, we propose an approach for semantic search by matching conceptual graphs. The detailed definitions of semantic similarities between concepts, relations and conceptual graphs are given. According to these definitions of semantic similarity, we propose our conceptual graph matching algorithm that calculates the semantic similarity. The computation complexity of this algorithm is constrained to be polynomial. A prototype of our approach is currently under development with IBM China Research Lab

4.1 Paper2

As defined in this way, information retrieval used to be an activity that only a few people engaged in: reference librarians, paralegals, and similar professional searchers. Now the world has changed, and hundreds of millions of people engage in information retrieval every day when they use a web search engine or search their email.1 Information retrieval is fast becoming the dominant form of information access, overtaking traditional database style searching (the sort that is going on when a clerk says to you: “I’m sorry, I can only look up your order if you can give me your order ID”). Information retrieval can also cover other kinds of data and information problems beyond that specified in the core definition above. The term “unstructured data” refers to data that does not have clear, semantically overt, easy-for-a-computer structure. It is the opposite of structured data, the canonical example of which is a relational database, of the sort companies usually use to maintain product inventories and personnel records. In reality, almost no data are truly “unstructured.” This is definitely true of all text data if you count the latent linguistic structure of human languages. But even accepting that the intended notion of structure is overt structure, most text has structure, such as headings, paragraphs, and footnotes, which is commonly represented in documents by explicit markup (such as the coding underlying web pages). Information retrieval is also used to facilitate “semi structured”. search such as finding a document where the title contains Java and the body contains threading. The field of IR also covers supporting users in browsing or filtering document collections or further processing a set of retrieved documents. Given a set of documents, clustering is the task of coming up with a good grouping of the documents based on their contents. It is similar to arranging books on a bookshelf according to their topic. Given a set of topics, standing information needs, or other categories (such as suitability of texts for different age groups), classification is the task of deciding which class(es), if any, each of a set of documents belongs to. It is often approached by first manually classifying some documents and then hoping to be able to classify new documents automatically. Information retrieval systems can also be distinguished by the scale at which they operate, and it is useful to distinguish three prominent scales. In *web search*, the system has to provide search over billions of documents stored on millions of computers. Distinctive issues are needing to gather documents for indexing, being able to build systems that work efficiently at this enormous scale, and handling particular aspects of the web, such as the exploitation of hypertext and not being fooled by site providers manipulating page content in an attempt to boost their search engine rankings, given the commercial importance of the web. We focus on all these issues in

Chapters 19–21. At the other extreme is *personal information retrieval*. In the last few years, consumer operating systems have integrated information retrieval (such as Apple's Mac OS X Spotlight or Windows Vista's Instant Search). Email programs usually not only provide search but also text classification: they at least provide a spam (junk mail) filter, and commonly also provide either manual or automatic means for classifying mail so that it can be placed directly into particular folders. Distinctive issues here include handling the broad range of document types on a typical personal computer, and making the search system maintenance free and sufficiently lightweight in terms of startup, processing, and disk space usage that it can run on one machine without annoying its owner. In between is the space of *enterprise, institutional,* and *domain-specific search*, where retrieval might be provided for collections such as a corporation's internal documents, a database of patents, or research articles on biochemistry. In this case, the documents are typically stored on centralized file systems and one or a handful of dedicated machines provide search over the collection. This book contains techniques of value over this whole spectrum, but our coverage of some aspects of parallel and distributed search in web-scale search systems is comparatively light owing to the relatively small published literature on the details of such systems. However, outside of a handful of web search companies, a software developer is most likely to encounter the personal search and enterprise scenarios. In this chapter, we begin with a very simple example of an IR problem,Central inverted index data structure.We then examine the Boolean retrieval model and how Boolean queries are processed .

An example information retrieval problem

A fat book that many people own is *Shakespeare's Collected Works*. Suppose you wanted to determine which plays of Shakespeare contain the words Brutus and Caesar and not Calpurnia. One way to do that is to start at the beginning and to read through all the text, noting for each play whether it contains Brutus and Caesar and excluding it from consideration if it contains Calpurnia. The simplest form of document retrieval is for a computer to do this sort of linear scan through documents. This process is commonly grep referred to as grepping through text, after the Unix command grep, which performs this process. Grepping through text can be a very effective process, especially given the speed of modern computers, and often allows useful possibilities for wildcard pattern matching through the use of regular expressions. With modern computers, for simple querying of modest collections (the size of *Shakespeare's* Collected Works is a bit under one million words of text in total), you really need nothing more. But for many purposes, you do need more 1. To process large document collections quickly. The amount of online data has grown at least as quickly as the speed of computers, and we would now like to be able to search collections that total in he order of billions to trillions of words. 2. To allow more flexible matching operations.

5. CONCLUSION

After the survey of these two[33,34] papers we observed that the above models provides the best match. But we need the exact match for Information Extraction. Hence there is a further need of another model which should provide the exact match .The paper presented two approaches to extracting information Structures. While the manual approach is more accurate and can be engineered in a domain specific way, the automated approach is also advantageous because of its scalability.Our evaluation of the automated system on both the BioRAT[25] and GeneWays[23] datasets shows that our system performs comparably with other existing systems. Both the systems compared were built by manual rule engineering approach, and involve a repetitive process of improving the rules which take up a lot of effort and time. Our system is able to achieve similar results with minimal effort on part of the developer and user. While advantageous on this aspect, we realize that our system is also in need of improvements in tagging entities to boost the performance. Improvements in the interaction extractor module will also bring up the precision of the system. Nevertheless, we have proven that a syntactic analysis of the sentence structure from full sentence parses produces results comparable to many of the existing systems for interaction extraction[[29,30].

Semantic matching has been raised for years to improve both recall and precision of information retrieval.[9,14] finds first the most similar entities and then observe the correspondence of involved relationship. However, with this kind of simplification, matching on nodes is separate without the organization of sub graphs. In contrary, we try to retain sub graph structure in our matching procedure with as less cost as possible. OntoSeek [8,13] defines the match on isomorphism between query graph and a sub graph of resource graph where the labels of resource graph should be subsumed by the corresponding ones of query graph. The strict definition of match makes their system can't support partial matching. The assumption that user would encode the resource descriptions by hand also limits its popularization. Different from it, our method not only supports the partial matching but also introduces the weight to reflect user's preferences, which makes our method more flexible. Some previous work have discussed the issue of semantic distance, such as [5] [14] and [15]. The basic thought is to define the distance between two concepts as the number of arcs in the shortest path between two concepts in the concept hierarchy which does not pass through the absurd type. [14] modified the definition and defined the distance as the sum of the distances from each concept to the least concept which subsumes the two given concepts. We adopt their original thought and make some modifications to make it suitable to our work.

The measurement of concept similarity was also studied before. [17] builds their similarity definition on the information interests shared by different concepts, while [16] defines the similarity between two concepts as the information content (entropy) of their closest common parent, and besides take the density in different parts of the ontology into account. The measuring of concept similarity in our approach is different from them and is simpler. Of course, our approach is far from perfect. It needs further study based on collected experiment data in the future .Now a days, with the electronic information explosion caused by Internet, increasingly diverse information is available. To handle and use such great amount of information, improved search engines are necessary. The more information about documents is preserved in their formal representation used for information retrieval, the better the documents can be evaluated and eventually retrieved. Based on these ideas, we are developing a new information retrieval system. This system performs the document selection taking into account two different levels of document representation. The first level is the traditional keyword document representation. It serves to select all documents potentially related to the topic(s) mentioned in the user's query. The second level is formed with the conceptual graphs[20,21,22] reflecting some document details, for instance, the document intention. This second level complements the topical information about the documents and provides a new way to evaluate the relevance of the document for the query. the query and extracts from it a list of topics (keywords). The keyword search finds all relevant documents for such a keyword-only query. Then, the information extraction module constructs the conceptual graphs of the query and the retrieved documents, according to the process described in section 3. This information is currently extracted from titles [10] and abstracts [11] of the documents. These conceptual graphs describe mainly the intention of the document, but they can express other type of relations, such as cause-effect relations [12,13].

This graph indicates that the document in question has the intention of demonstrating the validity of the technique Then the query conceptual graph is compared – using the method described in this paper – with the graphs for the potentially relevant documents. The documents are then ordered by their value *s* of the similarity to the query. After this process the documents retrieved at the beginning of the list will not only mention the key-topics expressed in the query, but also describe the intentions specified by the user. This technique allows improving the retrieval of information in two main directions:

1.It permits to search the information using not only topical information, but also extratopical, for instance, the document intentions.

2. It produces a better raking of those documents closer to the user needs, not only in terms of subject. We have described the structure of an information retrieval system that uses the comparison of the document and the query represented with conceptual graphs to improve the precision of the retrieval process by better ranking on the results. In particular, we have described a method for measuring the similarity between conceptual graph representations of two texts. This method incorporates some well-known characteristics, for instance, the idea of the Dice coefficient – a widely used measure of similarity for the keyword representations of texts. It also incorporates some new characteristics derived from the conceptual graph structure, for instance, the combination of two complementary sources of similarity: the conceptual similarity and the relational similarity. This measure is appropriate for text comparison because it considers not only the topical aspects of the phrases (difficult to obtain from short texts) but also the relationships between the elements mentioned in the texts. This approach is especially good for short texts. Since in information retrieval, in any comparison operation at least one of the two elements, namely, the query, is short, our method is relevant for information retrieval. Currently, we are adapting this measure to use a concept hierarchy given by the user, i.e. an *is-a* hierarchy, and to consider some language phenomena as, for example, synonymy. However, the use of the method of comparison of the texts using their conceptual graph representations is not limited by information retrieval. Other uses of the method include text mining and document classification.

The information extraction system based on complex sentence processing is able to handle binary relations between genes and proteins, and some nested relations. However, researchers are also interested in contextual information such as the location and agents for the interaction and the signaling pathways of which these interactions are a part. Our tasks for future work include the following

- Handling negations in the sentences (such as ”not interact”, ”fails to induce”, ”does not inhibit”)
- Identification of relationships among interactions extracted from a collection of simple sentences (such as one interaction stimulating or inhibiting another)
- Extraction of detailed contextual attributes (such as bio-chemical context or location) of interactions and
- Building a corpus of biomedical abstracts and extracted interactions that might serve as a benchmark for related extraction systems. Attempts to improve the parse output of the Link Grammar System were also undertaken. The dictionaries of the Link Grammar Parser[18] were augmented with medical terms with their linking to polynomial.

Before discussing the complexity of the algorithm, we firstly consider the effect caused by cycles in requirements provided by Szolovit in his website. In spite of the improvement in performance of the Link Grammar Parser, this approach was discontinued in favor of the Pre-processor subsystem because of the increase in time taken to load the dictionaries and for parsing. Semantic[24] analysis based on proposed information extraction techniques would enable automated extraction of detailed gene-gene relationships, their contextual attributes and potentially an entire history of possibly contradictory sub-pathway theories from biomedical abstracts in PubMed thus allowing our system to generate more relevant and detailed recommendations. When applying graph matching algorithm, the greatest worry comes about the computation complexity, since it is well known that Maximum Sub graph Matching is a NP-complete problem. Fortunately, it can be expected in our algorithm that the computation complexity will be constrained graphs to our algorithm. Since the algorithm is recursive, the cycle in graph will lead to an unending recursion and will be fatal to our algorithm. So we must eliminate the cycles in graphs before we match them. We can handle it simply by duplicating the concept in cycles. Surely, this will increase the computation complexity, especially when the cycle is very complex. Fortunately, benefiting from the domain specific characters, cycles in graphs are very rare especially in commodity domain. So we ignore it here.In the following, we will discuss the complexity of our algorithm. Since cycles in graphs are very rare and the cycles can be eliminated simply, we will only concern the tree structure. Without losing generality, we can suppose that the query graph and the resource graph contain n arcs each and are both *l*-branch trees of *i* height, so there are more than *li* relations. We use C(*i*) to denote the time complexity of matching two trees both of *i* height. As shown in the algorithm, we will calculate the similarity between the two entries firstly We use a constant c to represent the time spent in calculating concept similarity. After this step, the time complexity is c; then we need to calculate the similarity between each sub graph pair. Since each entry will induce *l* sub graphs we need *l*2 times recursive invocations. These sub graphs are all *l*-branch trees of *i*-1 height, so in every invocation, the time complexity is C(*i*-1). Here we ignore the time to calculate similarity between relations. After these two loops, the time complexity will be c+*l*2*C(*i*-1). Once we determine the similarity between each sub graph pair, we should find out the best match from different mate combinations There exists *l*! combinations in these *l*2 sub graph pairs, so how to handle it efficiently is important. We translate the issue into a *maximum flow* problem and execute Bellman-Ford[26] algorithm *l* times to solve it4, whose computation complexity is *l*3, and the cumulative complexity is *l*4. So the complexity can be described as follows: From the formula, we can see that C(*i*) is about *l2i+2*. Generally, when *l* is not very small, the number of arcs n will approximate *li*, so the complexity will be n*2l2*. If *l*<<n, the complexity will be O(n2). For the worst case, suppose there is only 1 layer in the query graph, i.e. *l*=n, the complexity is O(n4). Since the algorithm combines syntactic and semantic context information in the whole process[27,28],the advantages over traditional keyword match technique can easily be seen. For example, a description is about 'soft collar shirt' and another is about 'soft shirt with straight collar'. They are both selected by keyword search when using 'soft 4 InThe initial observation in this paper is that binary decisions are not good enough for ontology evaluation, when hierarchies are involved. We propose an Augmented Precision and Recall measure that takes into account the ontological distance of the response to the position of the key concepts in the hierarchy**.**

6.REFERENCES

[1] Cohen K. Bretonel and Lawrence Hunter. Natural language processing and system biology, 2004.

[2] I.H. Witten, A. Moffat, and T.C. Bell. Managing Gigabytes: Compressing and IndexingDocuments and Images. Van Nostrand Reinhold, New York, 1999.

[3] Susan T. Dumais, George W. Furnas, Thomas K. Landauer, Scott Deerwester, and

[4] Richard Harshman. Using latent semantic analysis to improve access to textual information.

[5] In Proceedings of the Conference on Human Factors in Computing Systems.

[6] Motwani, and T.Winograd.: The Page Rank citation ranking: Bringing order to the web. Technical report, Stanford University, 1998. Available at http://www-db.stanford.edu/~backrub/pageranksub.ps.

[7] Lum et.al.: An architecture for a multimedia DBMS supporting content search. In the Proceedings of International Conference on Computing and Information (ICCI'90), LNCS Vol.468, Springer-Verlag, 1990.

[8]N. Guarino, C. Masolo, and G. Vetere.: OntoSeek: "*Content-Based Access to the Web. IEEE Intelligent Systems*" 14(3), pp.70--80.

[9]Y. A. Aslandogan, C. Thier, C. T. Yu, C. Liu, and K. R. Nair.: Design, implementation and evaluation of SCORE(A System for COntent based REtrieval of pictures). In Eleventh International Conference on Data Engineering, pages 280—-287, Taipei, Taiwan, March 199.

[10] J. F. Sowa.: Conceptual Structures: Information Processing in Mind and Machine, Addison-Wesley. 1984.

[11] Lei Zhang and Yong Yu.: Learning to Generate CGs from Domain Specific Sentences. In proceeding of the 9th International Conference on Conceptual Structures, (ICCS2001), LNAI Vol.2120, Springer-Verlag, 2001

[12] Jonathan Poole and J. A. Campbell.: A Novel Algorithm for Matching Conceptual and Related Graphs. In G. Ellisetaleds, Conceptua lStructures: Applications, Implementation and Theory, pp. 293-—307, Santa Cruz, CA, USA. Springer-Verlag, LNAI 954, 1995.

[13] George A.Miller.: WordNet: An On-line Lexical Database. In the International Journal of Lexicography, Vol.3, No.4, 1990.

[14] John F. Sowa.: Knowledge Representation: Logical, Philosophical, and Computational Foundations, Brooks Cole Publishing Co., Pacific Grove, CA, 1999.

[15] N. Kushmerick, Daniel S. Weld and Robert B. Doorenbos.: Wrapper Induction for Information Extraction. Intl. Joint Conference on Artificial Intelligence pp.729—-737.

[16] Jianming Li, Lei Zhang and Yong Yu.: Learning to Generate Semantic Annotation for Domain Specific Sentences. In the Workshop on Knowledge Markup and semantic Annotation, the First International Conference on Knowledge Capture (K-CAP2001),Victoria B.C., Canada, Oct.2001.

[17] T.H.Cormen, C.E.Leiserson and R.L.Rivest.: Introduction to Algorithms. The MIT Press, 1994.

[18] W. Daelemans, S. Buchholz, and J. Veenstra.: Memory-Based Shallow Parsing.In Proceedings of EMNLP/VLC-99, pages 239-246, University of Maryland, USA,June1999.

[19] Norman Foo, B. Garner, E. Tsui and A. Rao.: Semantic Distance in Conceptual Graphs. In J. Nagle and T. Nagle, editors, Fourth Annual Workshop on Conceptual Structures, 1989.

[20] A. Ralescu and A. Fadlalla.: The issue of semantic distance in knowledge representation with conceptual graphs. In Proceedings of Fifth Annual Workshop on Conceptual Structures, pages 141--142, 1990.

[21] R. Richardson, A. F. Smeaton and J. Murphy.: Using WordNet as a Knowledge Basefor Measuring Semantic Similarity between Words. In the Proceedings of AICS Conference, Trinity College, Dublin, Ireland, September 1994.

[22] D. R. Cutting, D. R. Karger, J. O. Pedersen, and J. W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In ACM SIGIR '92, pages 318-329, 1992. [8] J. J. Daniels and E. L. Rissland. A case-based approach to intelligent information retrieval. In ACM SIGIR '95, pages 238-245, 1995.

[23] S. Dao and B. Perry. Applying a data miner to heterogeneous schema integration. In Proceedings of First International Conference on Knowledge Discovery and Data Mining, pages 63-68, 1995.

[24]] S. Deerwester, S. T. Adumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. JASIS, 41(6):391^07, 1990.

[25] [D. Dubin. Document analysis for visualization. In ACM SIGIR '95, pages 199-204, 1995.

[26]] U. Fayyad and R. Uthurusamy. Data mining and knowledge discovery in databases. Communications of the ACM, 39(11), 1996.

[27]] R. S. Flournoy, R. Ginstrom, K. Imai, S. Kaufmann, G. Kikui, S. Peters, H. Schiitze, and Y. Takayama. Personalization and users' semantic expectations. In Query Input and User Expectations, Proceedings of SIGIR Workshop, pages 31-35, 1998.

[28] J. Hammer, H. Garcia-Molina, K. Ireland, Y. Papakonstantinou, J. Ullman, and J. Widom. Information translation, mediation, and mosaic-based browsing in the tsimmis system. In Exhibits Program of the Proceedings of the ACM SIGMOD International Conference on Management of Data, pages 483^87, 1995.

[29] [. Z. Hasan, A. O. Mendelzon, and D. Vista. Applying database visualization to the world wide web. SIGMOD Record, 25(4):45-49, 1996.

[30] M. Hemmje, C. Kunkel, and A. Willett. Lyberworld - a visualization user interface supporting fulltext retrieval. In ACM SIGIR '94, pages 249-259, 1994.

[31] D. A. Hull, J. O. Pedersen, and H. Shutze. Method combination for document filtering. In Proceedings of SIGIR, pages 279-298, 1996.

[32] M. Iwayama and T. Tokunaga. Cluster-based text categorization: A comparison of category search strategies. In ACM SIGIR '95, pages 273-280,1995.

[33] Survey Paper1, A Conceptual Graph Matching For Semantic SearchJiwei Zhong, Haiping Zhu, Jianming Li andYong Yu.

[34] Survey Paper 2, Boolean Retrival, Christopher D. Manning.