Semantic-Structure-Based Search Engine*

Takashi Miyata
Core Research for Evolutional Science and Technology,
Japan Science and Technology Agency
miyata.t@carc.aist.go.jp

Kôiti Hasida
Information Technology Research Institute,
National Institute of Advanced Industrial Science and Technology
hasida.k@aist.go.jp

ABSTRACT
Our system represents the semantic structures of query and documents as graphs consisting of vertices as words and phrases and edges as dependency relations. Based on these semantic structures, the system provides the user with hints on how to revise his query. Preliminary experiments suggest that the system would reduce the user's time and effort for retrieving suitable documents. We will present the overview of the system and its extension to deal with a large amount of data such as Web pages.

Categories and Subject Descriptors
H.3.3 [**Information Systems**]: Information Search and Retrieval

General Terms
system design

Keywords
semantic structure

1. INTRODUCTION
In recent years, as various kinds of information are exchanged in machine-readable form, there has been widely required that the information be handled efficiently. Among such requirement, searching textual data particularly has been needed and provided in various situation, such as search engines for Web pages. Although searching textual data has been studied for a number of years, the requirement has not been completely satisfied yet.

One of the requirements that are not satisfied yet is to treat 'contents' of texts, which cannot be represented by mere sets of keywords. Searching textual data with 'contents' consists of two activities: extracting 'contents' from textual data and queries and calculating similarities among the extracted 'contents.'

Recent advances in computer performance and success in statistical approach in the field of natural language processing provide more accurate syntactic parsers. Obtaining propositional contents from simple sentences becomes quite easy in experimental use. On the other hand, knowledge acquisition and inference required in calculation of similarities among 'contents' have not been accurate nor easy yet even in experimental use.

Then, we approach to the problem in a way that a user and a system cooperatively search for the information; the system extracts propositional contents from an input query and provides the candidates and hints for revising his query to the user. The user revises his query based on the hints provided by the system. In the rest of the paper, we call 'contents' of texts *semantic structures* including not only propositional contents but also anaphora and rhetorical structures.

2. SEMANTIC STRUCTURES IN INFORMATION RETRIEVAL

2.1 Annotation for Semantic Structures
Since semantic structures can be represented by graph structures, they can be expressed by OWL or RDF. We, however, concern ourselves in representing semantic structures by *annotation* to a text written in a natural language. We adopt Global Document Annotation (GDA) [2], which is an instance of XML, as a framework to describe correspondence between texts and semantic structures. Figure 1 illustrates an annotation to Japanese sentence "*John-ga katta hon-o yabutta* (John tore a book which someone bought / Someone tore a book which John bought)." and its semantic structure.

The sentence itself is ambiguous. It means either "John bought a book" or "John tore a book." The annotation <np> in Fig. 1 displays that the meaning is "John bought a book" in this case. The attribute agt in the annotation <v> in Fig. 1 shows that the agent of tearing is Mary, who does not appear in this sentence.

2.2 Searching by Graph Matching
Our search engine converts documents and queries into the graphs in Fig. 1 and searches for the target(s) by graph matching. Figure 2 illustrates the case where the user inputs query graph representing "a Japanese business person has an accident in aboard" and the system found the portion of the graph of the documents representing "President Tanaka got involved in a vehicle crash in U.S.A.."

The left and right graphs in Fig. 2 represent the semantic structures of the query and the documents, respectively. Vertices in the query graph contain synonyms and related words, while vertices in the document graph are content words. Each vertex in the query graph matches the one in the document graph whose content word is one of the synonyms specified in the vertex.

*Work is supported in part by CREST grants, Japan Science and Technology Agency.

<su>
  <adp opr="obj">
    <np>
      <adp opr="agt">*John-ga* (John, MON)</adp>
      <v obj="mcn">*katta* (bought)</v>
      <n>*hon* (book)</n>
    </np>*o* (OBJ)</adp>
  <v **agt="Mary"**>*yabutta* (tore)</v>
</su>

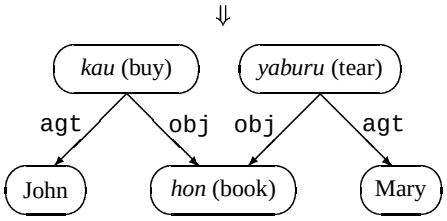

Figure 1: Annotation to "*John-ga katta hon-o yabutta* (John tore a book which someone bought / Someone tore a book which John bought)." (above) and Semantic Structure (below)

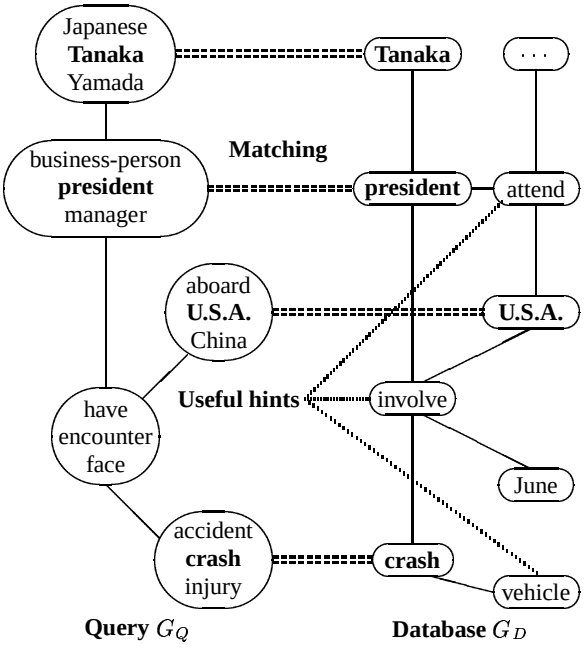

Figure 2: Information Retrieval by Graph Matching

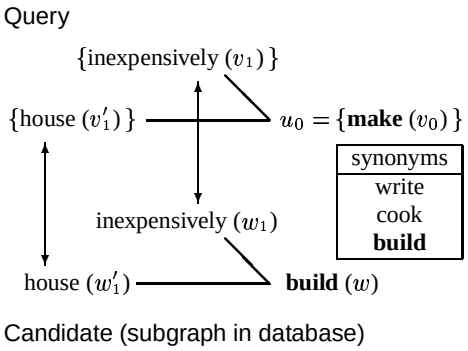

Figure 3: Structure-Sensitive Word Similarity

Note that graphs are matched approximately; vertices in a graph are not necessarily matched to the one in the other graph, as is the vertex 'vehicle' in the document. The problem of finding the specified sub-graphs in a given larger graph, *graph embedding*, is known to belong to the class NP-hard [1]. Query graphs, however, can be assumed small enough and currently naive implementation works within practical response time.

Note also that we simplify graphs as unlabeled and undirected ones due to the recall issues described in later.

2.3 Structure-Sensitive Word Similarity

It is difficult to think of keywords or expressions in an appropriate level of generality. For example, people sometimes say "making a house" to mean building a house. Semantic structures are useful in such cases. Suppose that a user inputs the above graph in Fig. 3 to obtain documents that describe "making a house inexpensively." Suppose also that there is a candidate like the graph below in Fig. 3. The two vertices in the query correspond to respective vertices in the candidate, but vertex u_0, which contains 'make,' does not[1].

In such a situation, the user could obtain a better candidate if the word 'build' were added to the synonyms for 'make.' This is a kind of local equivalence between words. 'Build' may be hard to relate to 'make' in a general context, but it should be related in the context of residential construction. We make use of this heuristics to rerank the synonyms which are provided users for revising their queries.

2.4 Merits and Demerits of Semantic Structures

Some of the merits and demerits of semantic structures in searching documents can be listed as follows:

- ○ **higher accuracy**
  Semantic structures can provide precise searching because the candidates will not be included which happen to contain the input keywords but do not have relation with user's intention[2].

[1]Similar configuration appears in Fig. 2; the vertex in the query containing 'aboard,' 'U.S.A.,' and 'China' corresponds to the vertex 'U.S.A.' in the document, and the vertex in the query containing 'accident,' 'crash,' and 'injury' corresponds to the vertex 'crash' in the document. In this case, the system will suggest the user to include 'involve' into the synonyms of 'have.'

[2]Detailed studies, however, exhibits that this is not always the case. See Sect. 5.

○ **detailed hints for query revision**
Matching semantic structures of the query and the documents can provide detailed hints for query revision. As is explained in the previous section, the vertices neighboring to the subgraph found are useful information to refine the query.

○ **proper representation of user's intention**
It is impossible even for human to guess what information the user needs only from a list of keywords. Graphs are as expressive as predicate logic, therefore users can express their intention appropriately.

× **lower recall**
Using a graph as condition for searching is more rigid than using a list of keywords. Therefore the recall is apt to become lower. To compensate the decrease of the recall, interaction between user and system is important[3].

× **difficulty providing 'proper' semantic structures**
We have found that the subjects sometimes input strange semantic structures in the experiment described in the next section. Some subjects said "it is difficult for me to input semantic structures." We, however, think that the real problem is interface and feedback of the system, not the complexity of semantic structures.

× **cost of parsing and indexing**
Actually parsing and indexing documents take quite resources, but we have an impression that we can practically manage within about 1,000,000 document each of which is a few kilo bytes with recent middle-class computers.

3. EFFECTS OF SEMANTIC STRUCTURE

We have conducted a preliminary experiment to verify the utility of semantic structures in information retrieval [7]. Eight subjects (five males and three females, age 20–35) who use Web browsers, search engines, and e-mail in daily activity were told to find articles that describe the following content from 100,000 Mainichi newspaper articles (news and editorials) in 1994 without time limitation:

1. A boy who beat Prime Minister Major in a vote
2. Subsidiary to be established in the future is evaluated better than its parent company.
3. Area in China where people obtain capital from aboard
4. Phone calls increase greatly when the party's name appears in mass media.

Table 1 shows the averages and the standard deviations of (1) the ranks of correct articles listed by the system when the subjects found them, (2) the time until the subject found articles or gave up, and (3) the average number of operations. The row named 'Keywords' shows the results of setting where the subjects uses only lists of keywords while the row named 'Structures' shows the results of setting where the subjects uses semantic structures. The numbers of subjects who could obtain correct articles are four for the keywords-setting and six for the structures-setting. Although the differences between the two settings are not statistically significant due to their large deviations, the table implies that using semantic structures can improve the efficiency of information retrieval.

[3]This problem is partially reduced by simplifying the graphs of semantic structures as unlabeled and undirected ones. This simplification makes 'a book which was bought by John' and 'John bought a book' have the same structures.

Table 1: Effects of Semantic Structure in Information Retrieval (average and standard deviation)

	Rank	Time (min)	# Operations
Keywords	32.71 (36.64)	18.01 (12.00)	27.62 (10.76)
Structures	1.50 (0.71)	7.62 (4.46)	13.62 (6.32)

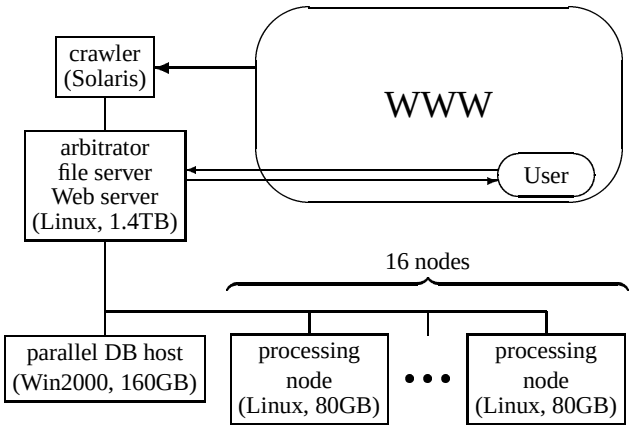

Figure 4: System Overview

4. ENHANCEMENT FOR LARGE-SCALE SET OF DOCUMENTS

We are currently enhancing the system to handle large-scale sets of documents. Figure 4 illustrates the overview of the system. The system consists of 19 computers, which are depicted by rectangles in Fig. 4. The number in parentheses in each rectangle is the amount of the storage of the computer.

A crawler is running on Solaris workstation to collect Web pages. The rest of the machines forms an isolated local network. The Linux machine depicted in the middle of Fig. 4 arbitrates the data flow and control of the system. It serves file system for the crawler and the processing nodes, gateway between the local network and outer network, and CGI services for interface to users. We adopt parallel database system SISA manufactured by Mitsubishi Electric Corporation [3], which consists of one host on Windows 2000 and arbitrary number of PCs (processing nodes) on Linux. Each processing node also reformats the collected Web pages, parses them, and makes index of the parsed documents for our original database.

In preparation for the database, the following processes are running; a crawler on the crawler machine and cleaners, filters, parsers, and indexers on processing nodes. Their invocations are arbitrated by the process on the arbitrator machine.

- crawler
  We adopt GNU `wget` and slightly modify the format of the log to be processed by other programs. The original functions such as retrying and downloading based on time stamp

Table 2: Performance of Document Collection and Indexing

# pages downloaded	1,500,000
# pages recognized as Japanese Text	1,260,000
Time for downloading	7 days
Time for indexing original DB	3.5 days
Time for indexing SISA	3 days

are used as they are.

- cleaner and filter
  The encoding of pages collected by the crawler are converted into euc-jp. Then, the patterns described in regular expressions are matched against the pages. This process strips HTML tags and determines whether the page is written in Japanese based on the ratio of the euc-jp characters and others (ASCII). Although the data described in Table 2 is obtained by the version which handles only plain texts and HTML, we currently combine converters from other format such as PDF, Microsoft Word, Excel, and PowerPoint to plain text into the system.

  These processes also split large pages into smaller pieces.

- parser and indexer
  Morphological and syntactic analyzers used here are statistical ones, which output some parses even for data not in Japanese. We have developed original database system based on B-tree and binary search to manage synonyms and neighboring words. Parallel database system SISA is used to perform filtering pages by keyword-based search and ranking the filtered pages by graph matching. In SISA, data for vertices and edges in each page is stored as 1 record and graph matching algorithm is implemented as a user-defined function.

Table 2 shows the time to collect and index pages which are traversed from 70,000 URLs with following links five steps further. All data is for initial registration.

Figure 5 shows a screenshot of the interface which is written by Flush and accessed through Web browser. The window consists of three panes. The first pane at the top of the window contains an input field. Users inputs his query as a Japanese sentence into the field. The query appeared in Fig. 5 is "*Okina gamen-de eiga-o miru* (Seeing movies on a large screen)."

Then the system parses the sentence and displays its semantic structure. The four boxes and three arcs in the second pane represent the semantic structure. Each box pop-ups a list of synonyms like the third box. User can choose appropriate synonyms and let the system search again by clicking the leftmost button. The synonyms are listed in the order of their score explained in Sect. 2.3.

The last pane contains a list of candidates in the order of their scores. Each candidate is displayed with matching words in bold face. User can browse the document itself by clicking of the candidate.

5. RELATED WORK

Mitra et al. [5] evaluated the effect of phrasal information in IR using about 210,000 documents from the Wall Street Journal, AP Newswire, and Ziff-Davis sections of TREC disk 2 and 50 queries in TREC. They report that the precision is not improved in the case where the documents can be ranked using keywords only. They also report that phrasal information does not work toward filtering top-ranked non-relevant documents.

The reason they inferred is that keyword-based ranking highly ranks documents by multiple aspects of the query. On the other hand, phrasal information emphasizes only one of these aspects. Therefore, the effect of phrasal information is weakened among highly ranked documents. They conclude that phrasal information should be used to re-rank the low-ranked documents.

Miyakawa et al. [6] obtained similar results from an experiment using 430,000 Mainichi news articles. These results suggest that the naive belief that linguistic structure improves the precision in IR should not hold.

Importance of interaction between user and system has been gradually noticed. Early conferences of TREC were mainly concerned with shared tasks without human interaction. TREC-6, however, started an interactive IR track [4] which compared the performance of 12 interactive IR systems with detailed experimental design and analysis. They have provided a guideline for experiments which must be performed by different locations and sets of subjects.

As XML is used widely, various researches for pattern matching against XML data arise. Difference between the settings in XML applications and our settings in matching semantic structures is the meaning of queries. Queries in XML applications represent a set of instances or *a class*, while our queries for semantic structures are instances. Therefore similarity between instances plays an important role in our application. Shasha et al. [8] survey algorithms and applications of tree and graph searching including general purpose ones.

6. SUMMARY

Interaction between user and system is important in information retrieval based on semantic structures. Our preliminary experiment indicates that the merit of using semantic structures in information retrieval is not the improvement of retrieval accuracy, but the improvement of efficiency in interaction. Although the cost of semantic-structure-based search might be thought quite higher than the keyword-based one, we have an impression that we can practically manage within about 1,000,000 document each of which is a few kilo bytes with recent middle-class computers.

7. ACKNOWLEDGMENTS

We would like to thank Mitsubishi Electric Corporation for cooperation to research and implement our system.

8. REFERENCES

[1] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and Approximation: Combinatorial Optimization Problems and their Approximability Properties*. Springer-Verlag, 1999. ISBN: 3-540-65431-3.

[2] K. Hasida. *Global Document Annotation*, 2003. `http://www.i-content.org/gda/`.

[3] M. Kori, Y. Yamagishi, H. Shimizu, and Y. Kaneko. Large-scale parallel full-text search implemented by storage system with search function. In *Technical Report*, number

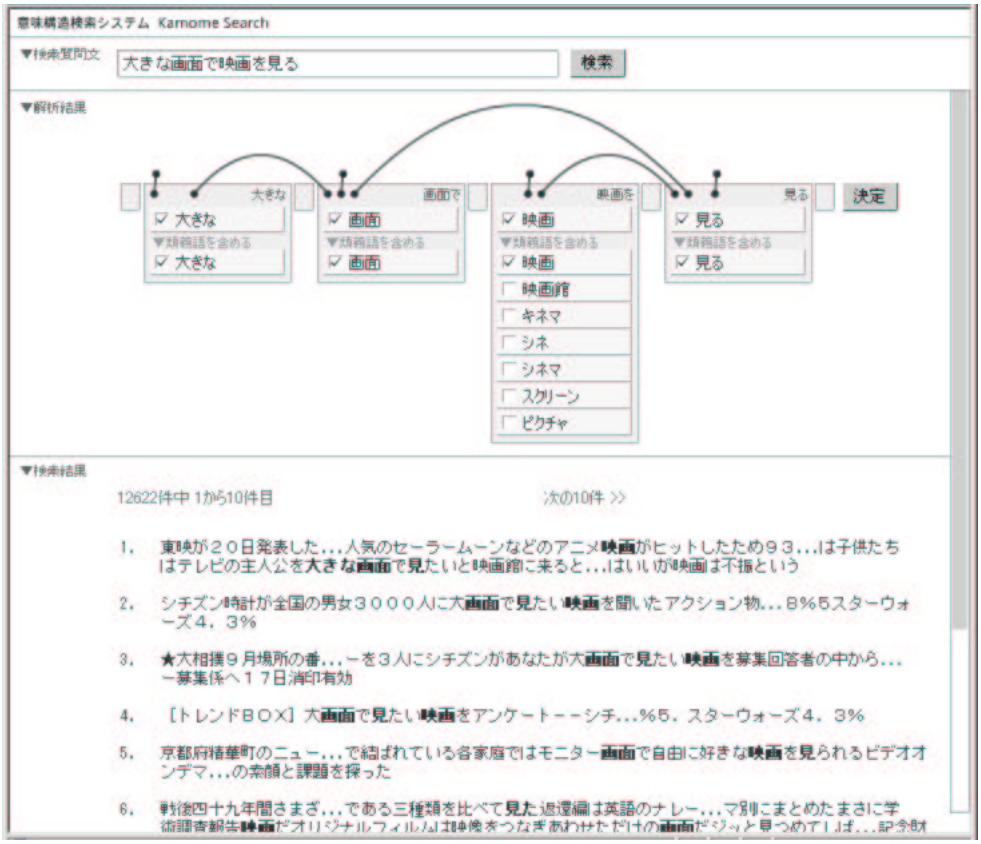

Figure 5: Screenshot

102(276) in CPSY2002-47, pages 41–46. The Institute of Electronics, Information and Communication Engineers, August 2002. (in Japanese).

[4] E. Lagergren and P. Over. Comparing interactive information retrieval systems across sites: The TREC-6 interactive track matrix experiment. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 164–172, Australia, 1998.

[5] M. Mitra, C. Buckley, A. Singhal, and C. Cardie. An analysis of statistical and syntactical phrases. In *RIAO '97*, pages 200–214, 1997.

[6] K. Miyakawa, T. Tokunaga, and H. Tanaka. Information retrieval using case frame. In *Proceedings of the Fourth Annual Meeting of the Association for Natural Language Processing*, pages 112–115. The Association for Natural Language Processing, March 1998. (in Japanese).

[7] T. Miyata and K. Hasida. Information retrieval based on semantic structures. In *Proceedings of the 2nd Language and Technology Conference*, Poznań, Poland, April 2005.

[8] D. Shasha, J. T. L. Wang, and R. Giugno. Algorithmics and applications of tree and graph searching. In *Proceedings of the Symposium on Principles of Database Systems*, pages 39–52, June 2002.