PROCEEDINGS OF THE INTERNATIONAL CONFERENCE
ON FIFTH GENERATION COMPUTER SYSTEMS 1988,
edited by ICOT.

KNOWLEDGE REPRESENTATION WITH ABSTRACTIVE LAYERS FOR INFORMATION RETRIEVAL

Takuo Koguchi†, Hidefumi Kondo†
Masahiro Oba‡*, Hidenori Itoh‡

† Systems Development Laboratory, Hitachi, Ltd.
1099 Ohzenji , Asao-ku, Kawasaki-shi, 215 Japan
‡ Research Center, Institute for New Generation Computer Technology
4-28, Mita 1-chome, Minato-ku, Tokyo, 108 Japan

ABSTRACT

This paper describes a knowledge representation scheme for knowledge about the contents of data (for example, "text") and abstracted information of data. In the proposed scheme, the contents of data are initially represented by a kind of semantic network in which the case frame structure of the data is expressed explicitly, and abstracted information of the data which is extracted from the semantic network is also represented in the form of a semantic network. By step-by-step abstraction, abstractive layers of knowledge about the data are formed, the components of which are eventually linked with corresponding data. The objective of the proposed scheme is to offer a general view of the contents of the data stored in an information retrieval system. In an interactive environment, it allows users to move as they wish from the general to the particular as well as to change the viewpoint of retrieval. Stored knowledge about the data serves as a phrase index for retrieval which is more desirable than a single keyword index as far as precision of retrieval is concerned. Since items of abstracted information which are common among the information of lower layers can be related to each other, some identification based on the relation makes it possible to locate information similar to the specified data.

1 INTRODUCTION

The need for intelligent information retrieval systems is widely recognized [Teskey 1987], [Sakamoto 1987] and [Rau 1987].

Using a keyword index is a simple and traditional way of retrieving the items which meet the specified requirement from a large-scale document database. The full text search method has become practical through the recent growth of efficient hardware. In retrieval with a keyword index, keywords to represent the contents of the primary information are attached and stored as secondary information. In retrieval, only the secondary information is searched and corresponding items of the primary information are located. In full text search, there is no secondary information about the contents of the primary text information like keywords. In retrieval, the primary text information is searched for matching patterns directly and the items containing such patterns are located.

In both these methods, retrieval conditions are usually specified by Boolean expressions which consist of keywords or matching patterns and logical connectives. When users want to specify semantic relations of keywords as query conditions, they cannot describe their intended retrieval conditions precisely because no relationships among the keywords other than logical connection can be described in such an expression. If semantic relations of keywords are expressed in users' retrieval requests as in a natural language query, secondary information such as simple keywords attached to the primary information is not expressive enough to answer the requests. Furthermore, the degree of retrieval intended by users is so widely diversified that more sophisticated secondary information is necessary for information retrieval systems to fulfil the users' retrieval requests.

Our approach to these problems is to build a system using a knowledge base to store such sophisticated secondary information about the

*Current address: above-mentioned †

contents of data as knowledge in some levels of abstraction. The knowledge representation scheme is a crucial point in building such a system. Many knowledge representation schemes have been proposed, for example, frames [Bobrow 1977] and semantic networks [Janas 1979]. Semantic networks lack representations of the relationships between the contents represented by nodes and semantic relation among them and the simplified contents. Inheritance, which is the main feature of frame systems, does not operate effectively on abstraction structures of individual data. We propose a knowledge representation scheme with abstractive layers which include high-level secondary information consisting of (1) extracted conceptual information which reflects the literal meaning of the primary information and (2) abstracted conceptual information which is taken from underlying conceptual information. These two types of information will be closely related and structured so that they can be combined.

Primary information which is expressible in natural language is in the scope of the present article. In the following, it is assumed that primary information is given in the form of texts.

Section 2 of this paper presents the proposed structure of a knowledge base for information retrieval. Section 3 presents a knowledge description of the knowledge base. Section 4 describes how to construct knowledge in a layered structure. Section 5 presents some illustrations of retrieval using the knowledge base.

2 LAYERED STRUCTURE OF THE KNOWLEDGE BASE

We focus on the ability to answer widely diversified degrees of retrieval requests which is one of the requirements for intelligent information retrieval systems. It means (1) offering information in various levels according to the degree of retrieval intended by a user, (2) offering information according to changes of the degrees, from the general to the particular, and vice versa, and (3) offering information according to changes of the users' viewpoints. We propose an architecture of the knowledge base, which describes the contents of primary information in many levels corresponding to the degrees of users' retrieval requests to fulfil the requirement.

Figure 1 shows an architecture for storing and retrieving knowledge. This architecture has seven elements: the abstraction function (h in fig. 1), which is not in the general retrieval model [Kondo 1987]; and the other six elements in the general retrieval model, namely, individual objects to be stored (p in fig. 1), a function for storing individual objects (g in fig. 1), the set of stored objects (M in fig. 1), queries for the set of stored objects (Q in fig. 1), a function for retrieving objects from the set of stored objects in accordance with queries (f in fig. 1), and the retrieval results of queries (A in fig. 1).

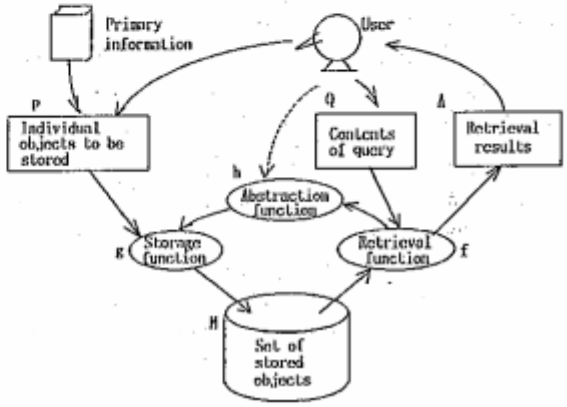


Fig. 1 An architecture for storing and retrieving knowledge

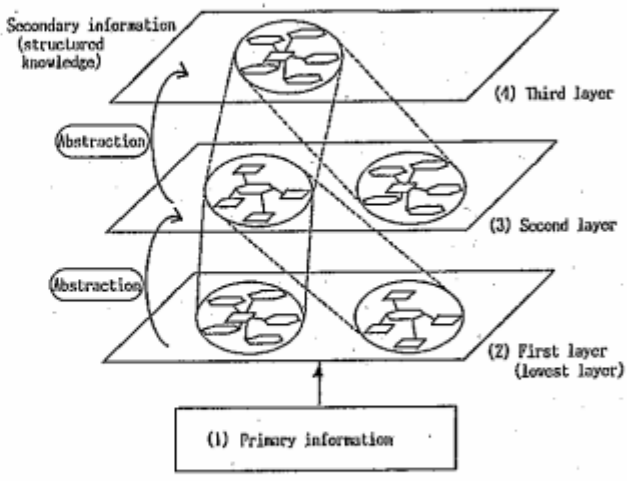


Fig. 2 Layered structure of the knowledge base

The relationship among these elements can be expressed as follows:

$$M=\{m \mid m=g(p,\_) \text{ or } m=g(h(f(M, Uf), Uh), Ug)\} \quad (1)$$

$$A=f(M,Q) \quad (2)$$

where Uf, Uh and Ug stand for the interaction of users, if any, on applying functions f, h and g. Equation (1) shows that the set of stored objects, M, is a set of individual objects, p, and objects which are extracted by the abstraction function, h, and the retrieval function, f, and stored by the storage function, g. Equation (2) shows that the retrieval results of query, A, are the consequence of retrieval function, f, operating on the set of stored objects, M, and the contents of query, Q.

Figure 2 shows the basic structure of knowledge in the proposed knowledge representation scheme with abstractive layers. In figure 2, (1) indicates original data items (texts) which are primary information, and (2), (3) and (4) indicate abstractive layers containing structured knowledge which we call high-level secondary information about the contents of primary information. An acceptable viewpoint (for example, a technical viewpoint of a specific field of stored information) provides the basis of abstraction. Each step of abstraction from a layer generates structured knowledge in the next layer. Since the contents of primary information can generally be expressed in various degrees of abstraction, structured knowledge will form many layers in the knowledge base.

Structured knowledge in the first layer, which is the lowest level of the abstractive layers, consists of extracted conceptual information. Extracted conceptual information is a fragmentary item of knowledge which presents the literal meaning of corresponding text information. Structured knowledge in higher layers consists of abstracted conceptual information. Abstracted conceptual information is knowledge which combines knowledge in the lower layers, and is eventually linked to the corresponding text information via lower layers.

As a representation of structured knowledge in an abstractive layer, the proposed scheme uses a kind of semantic network which explicitly expresses the case frame structures of sentences in text information. The reason is that relationships among words or concepts derived from the case frame structures play an important part in representing the contents of text information and the retrieval process, and the commonality of these relationships shows a possible semantic similarity.

The relationship between structured knowledge in different layers is knowledge about abstraction operation, and represents abstraction function mapping from a lower layer onto a higher layer.

From the basic considerations described above, we have developed a description of structured knowledge, including construction of abstractive layers and utilization of structured knowledge in abstractive layers for the retrieval process.

3 DESCRIPTION OF STRUCTURED KNOWLEDGE

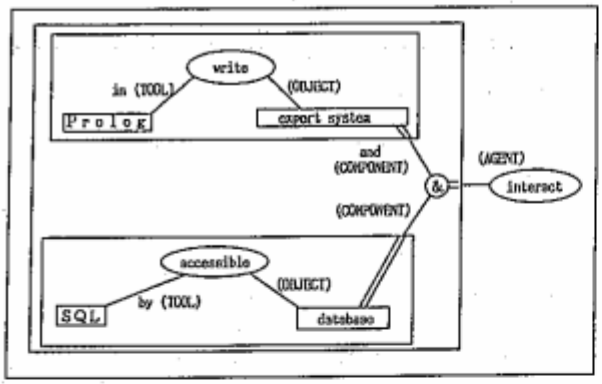
Although our current interest is not in natural language processing itself, we designed simple notations of structured knowledge that represent the contents of text information and abstracted information of texts as bases of a knowledge base for information retrieval systems. This section describes two sorts of knowledge notation, symbolic and graphical, which are common to all the levels of abstractive layers.

Graphical notation of structured knowledge is not only used as an intermediate form when describing structured knowledge in symbolic notation in order to input to the system, but is also used for showing structured knowledge in the knowledge base when composing structured knowledge in higher layers in reference to existing knowledge.

For these purpose, the notation should possess comprehensibility (the contents of the text information must be easily understood from it), constructibility (representation of the text information by the notation must be simple), and formalized representation (potential for this transformation must be automatic in future). For the reason given in the previous section, we use a graphical notation of networks in which relationships among words or concepts are instinctively comprehensive.

In the lowest abstractive layers, nodes (simple nodes) of the network designate words in the text and arcs of the network present a case relationship among those nodes. Certain groups of nodes often behave as single nodes. In such a case, the structure of the group is called a composite node, in contrast to a simple node. The relationships between structured knowledge are represented by arcs which connect couples of the nodes in both items of structured knowledge.

In graphical notation of structured knowledge (see figure 3), labelled ovals in the diagram represent nodes designating verbs which have case frame structures, labelled boxes in the diagram represent nodes designating the item filling the case slots in the case frame, and lines labelled with case names in the diagram represent the case relationships between the verbs and items filling the case slots. Composite nodes are shown by boxes enclosing the constituent nodes and arcs. An entry node of a composite node is a node corresponding to the noun eventually modified by other constituent nodes of the composite node or a node corresponding to a verb when the composite node represents a noun clause. When a case slot is filled with a composite node, the entry node of the composite node is indicated by a double line from a box representing the composite node.




```
$'885021786': [
    $1: [
        [write, 'Prolog':[in, $'TOOL'], 'expert system': $'OBJECT']= @1,
        [accessible, 'SQL':[by, $'TOOL'], database: $'OBJECT']= @2,
        [&, @1 /'expert system':and, @2 /database]= @3,
        [interact, @3: $'AGENT']
    ]
].
```

Fig. 3 Example of structured knowledge description
Case labels are capitalized in the notation. This figure describes the phrase, "An expert system written in Prolog and a database accessible by SQL interact".

Figure 3 shows an example of a graphical description of structured knowledge in the first layer. This structured knowledge represents the phrase, "An expert system written in Prolog and database accessible by SQL interact". In the part "An expert system written in Prolog", the node "write" has a case frame, the node "expert system" fills the case slot "OBJECT", and the node "Prolog" fills the case slot "TOOL" in the case frame. In the part "database accessible by SQL", the node "accessible" has a case frame, the node "database" fills the case slot "OBJECT", and the node "SQL" fills the case slot "TOOL" in the case frame. These two parts form composite nodes, whose entry nodes are "expert system" and "database", respectively. These two composite nodes are joined by the node "&" to form a composite node, which fills the case slot "AGENT" in the case frame of the node "interact".

A symbolic notation which is readable as a part of ESP (Extended Self-contained Prolog [Chikayama 1984]) programs gives a formal description of structured knowledge described in graphical notation. The BNF syntax of the symbolic notation is shown in figure 4. In the figure, elements in quotation marks are terminal symbols and elements enclosed by angle brackets < > are non-terminal symbols. Square brackets [ ] are used to indicate optional constructs.

A <text>, which is an item of structured knowledge corresponding to an abstract consisting of some sentences, consists of a <text_identifier>, designating an abstract and a <net_list>, representing the contents of the abstract. A <net_list> is a set of <net>s corresponding to individual sentences. Each <net> consists of a <net_identifier>, designating a sentence in an abstract, and <net_contents>, representing the contents of the sentence. A <net_contents> is a set of

```
<text> ::== "#" <text_identifier> ":" "[" <net_list> "]" "."
<text_identifier> ::== <quoted_atom>
<net_list> ::== <net> | <net> "," <net_list>
<net> ::== <net_identifier> ":" <net_contents>
<net_identifier> ::== "$" <integer>
<net_contents> ::== "[" <net_contents_list> "]"
<net_contents_list> ::== <net_contents_atom> ";" |
          <net_contents_atom> ";" <net_contents_list>
<net_contents_atom> ::== "[" <predicate> "," <cases> "]"
          [ "=" <atom_identifier> ]
<atom_identifier> ::== "@" <integer>
<predicate> ::== <predicate_atom> [ ":" <predicate_attributes> ]
<predicate_atom> ::== "$" <predicate_index> |
          "*" | "&" | <predicate_string>
<predicate_attributes> ::== <predicate_attribute_atom> |
          "[" <predicate_attribute_list> "]"
<predicate_attribute_atom> ::== <ESP_term>
<predicate_attribute_list> ::== <predicate_attribute_atom> |
       <predicate_attribute_atom> "," <predicate_attribute_list>
<cases> ::== <case> | <case> "," <cases>
<case> ::== <term_label> [ ":" <roles> ]
<term_label> ::== [ <reference_text_identifier> "/"
   [ <reference_net_identifier_list> "/" ] <atom_identifier> "/" ] ]
       <term_string>
<reference_net_identifier_list> ::== <reference_net_identifier> |
       <reference_net_identifier> "/" <reference_net_identifier_list>
<roles> ::== <role_atom> | "[" <role_list> "]"
<role_atom> ::== "$" <role_index> |
       <role_string>
<role_list> ::== <role_atom> | <role_atom> "," <role_list>
<reference_text_identifier> ::== "#" <text_identifier>
<reference_net_identifier> ::== <net_identifier>
<role_index> ::== " AGENT" | "OBJECT" | "TOOL" | ...
<predicate_index> ::== "equivalent" | "is_a" | ...
<predicate_string> ::== <string>
<role_string> ::== <string>
<term_string> ::== <string>
<data> ::== "data(" <text_identifier> "," "``([" <net_list> "])" ")" ";"
```

Fig. 4 Syntax for structured knowledge description

<net__contents__atom>s describing composite nodes which constitute a sentence.

A <net__contents__atom> describing a composite node consists of a <predicate>, which has a case frame, and a <cases> which is a set of <case>s (case slots). When a composite node is referred to in order to fill the case slot of the case frame in another <net__contents__atom>, an <atom__identifier> is given to the referred <net__contents__atom>. Identifiers are arranged so that every composite node can be uniquely identified by the combination of its <text__identifier>, <net__identifier> and <atom__identifier>. A <predicate> corresponding to a verb which has a case frame consists of a <predicate__atom> which represents the meaning of the <predicate>, and a <predicate__attribute>, representing some other information (tense, modality, etc.), which this paper does not deal with. A <predicate__atom> is a <predicate__string> which is a verb in the phrase, a <predicate__index> which represents an equivalence or subsumption relationship between composite nodes, a symbol "&" which joins composite nodes to fill a case slot, or a symbol "*" which forms a composite node in the absence of a verb. A <case> consists of a <term__label> which shows the item that fills the case slot and a <roles> which represents the case relationship. A <term__label> is a <term__string> which is a word in the sentence if the item that fills the case slot is a simple node, otherwise it is a <term__string> qualified by a <text__identifier>, a <net__identifier> and an <atom__identifier>. A <roles> is a <role__atom> or a set of <role__atom>s. A <role__atom> is a <role__index> which designates the name of the case relationship or a preposition in the sentence.

An example of structured knowledge in symbolic notation is shown in figure 3, which corresponds to the example of graphical notation.

Each item of structured knowledge is described in the form of <data> in figure 4 to be embedded in an ESP program by restraining the macro expander. In addition, the following operator definitions are necessary:

```
add__operator((@), fx, 40).
add__operator(($), fx, 50).
remove__operator((*)).
remove__operator((/)).
add__operator((/), xfy, 95).
```

We have examined the notations described above by using them to represent more than 100 abstracts of technical documents written in Japanese. It is proved that the practical meanings of abstracts are not changed.

4 CONSTRUCTING ABSTRACTIVE LAYERS

Abstracted conceptual information, which is one of the two types of structured knowledge

stored in the knowledge base as secondary information, is obtained by abstraction of the structured knowledge in the lower abstractive layers, and constitutes the higher abstractive layer. Figure 5 shows an example of stored knowledge in the proposed knowledge representation scheme with abstractive layers.

Abstraction is a set of operations on a network structure of knowledge, consisting of the following three steps (see figure 6):
(1) extraction, the step in which, according to certain viewpoints, items of structured knowledge are retrieved from a lower layer as the objects on which a single step of continual abstraction operation is performed,
(2) abstraction, the step in which retrieved items are contracted in some different ways depending on the viewpoints, and
(3) integration, the step in which contracted items are stored in the higher layer and bidirectional relationships are established to arrange the items so that they are accessible from both layers.

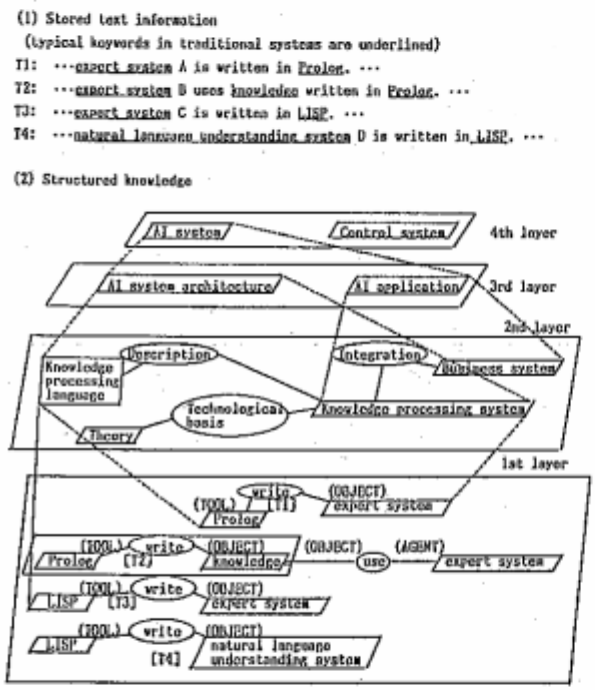


Fig. 5 Example of stored knowledge

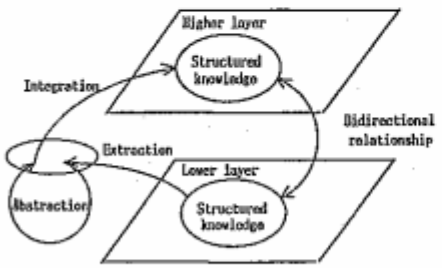


Fig. 6 Transformations and bidirectional relationship between structured knowledge

In the extraction step, items of structured knowledge which have similar structures are collected together. Items of structured knowledge have a similar structure if there are (1) a <predicate_atom> which is common to each item and (2) a set of case slot fillers which is a non-empty subset of the set of case slot fillers of the items. In this matching process, two fillers of the case slots in different case frames are identified if they are related by equivalence or subsumption, or if they are already connected to a certain composite node in the higher layer. A lack of case slot fillers in a few items is tolerated when no similar structures can be found in a strict matching process. Items of structured knowledge also have a similar structure when respective case frame structures in the items are reducible to a similar structure described above by applying transformational rules peculiar to a <predicate_atom> or a verb.

In the abstraction step, the contents of a similar structure which is found in the extraction step are summarized by simplification and generalization. Simplification means replacing components of structured knowledge which represent particular concepts with simpler structures which designate the same concepts. For example, the node "AI application" in the third layer (in figure 5) is the result of simplification of the structure, "Knowledge processing system" - "Integration" - "Business system". A transformational rule such as "Integration of a system with another is an application of the system" applies to this simplificative operation.

Generalization means extracting the entry node of a composite node or replacing a node which represents a particular concept with a simple node which designates a more general concept. For example, the structure "Knowledge processing language" - "Description" - "Knowledge processing

system" in the second layer (in figure 5) is the generalized structure of the structures in the first layer, "Prolog" - "write" - "expert system", "LISP" - "write" - "expert system" and "LISP" - "write" - "natural language understanding system", using the generalization-specialization relationships, "Prolog and LISP are knowledge processing languages" and "Expert systems and natural language understanding systems are knowledge processing systems".

The integration step includes intra-layer operations and inter-layer operations. In intra-layer operations, a simplified or generalized structure of structured knowledge is inserted in the higher layer and similar structures in it are related to each other. In inter-layer operations, bidirectional relationships are established to arrange the items so that they are accessible from different layers.

5 RETRIEVAL USING STRUCTURED KNOWLEDGE

This section illustrates typical ways of retrieval using structured knowledge, (1) retrieval by browsing views of existing relationships of keywords in the stored text information, (2) similarity-based retrieval, and (3) retrieval utilizing the relationships between keywords expressed in a query, in contrast to traditional retrieval with simple keywords. In this section, the knowledge base shown in figure 5 is used as an example.

The proposed system offers retrieval by browsing views of existing relationships of keywords in the stored text information. This enables the users to pick out part of the contents of stored text information and to specify the retrieval conditions to retrieve desired results stepwise more precisely or more appropriately. Figure 7 shows an example of this kind of retrieval. In processes (1)-(5), the user is offered knowledge about the contents of stored text information and moves step by step from an ambiguous retrieval condition (1) about "AI system" to more particular conditions. Although the retrieval process in this example is rather straightforward, offering knowledge to the users as views of stored data is more effective as users move more freely, changing the viewpoints by imposing a new condition or modifying the condition.

Traditional information retrieval systems have thesauri to show what is stored in a system and what combination of keywords will produce the desired results. Since an ordinary thesaurus does not represent specific relationships of keywords to the stored data but universal relationships, it is not always possible to browse the contents of stored data effectively.

The similarity-based retrieval presented here is associative retrieval in which a retrieval condition is specified by an instance of stored texts and texts which are semantically similar to the given instance are retrieved as retrieval results. Any similarity between the texts is regarded as a similarity between structured knowledge extracted from the texts. A text which has a common structure with the given instance in a certain abstractive layer is considered as a

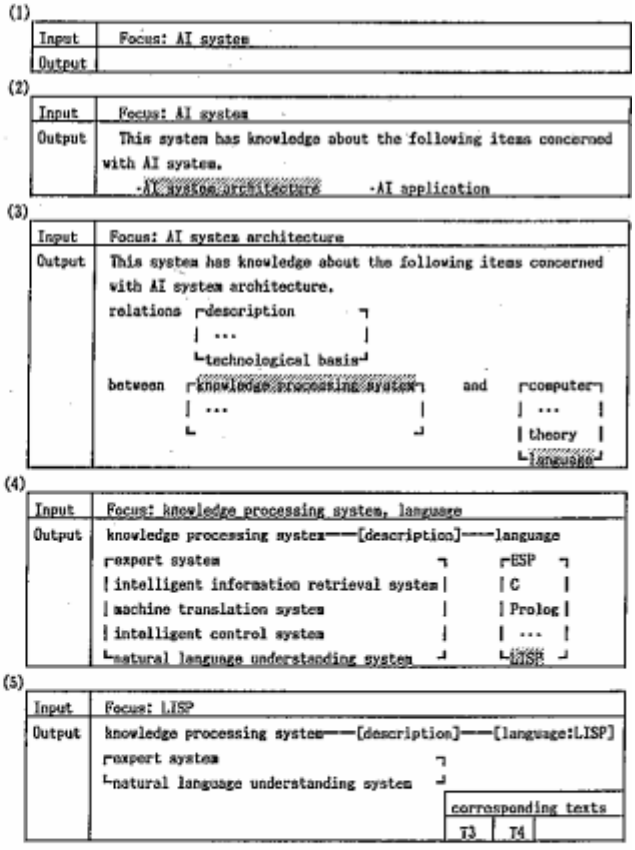


Fig. 7 Retrieval by browsing views of relationships existing in the stored information

candidate of retrieval results. Similar structures are ordered by levels of abstractive layers in which common structures are found and there is an equivalence or subsumption relationship between words in the structures.

In traditional retrieval with simple keywords, any similarity between the texts is measured by common portion of respective sets of keywords attached to the texts. As this measurement neglects the variety of relationships of keywords, it offers only a rough approximation.

Using structured knowledge as a phrase index is another feature of retrieval of the proposed system, in which the relationship between keywords expressed in a query is utilized to locate retrieval results. For example, a query "find a text about an expert system which is written in Prolog" is expressed as the relationship "description" between "expert system" as the item that fills the case slot "OBJECT" and "Prolog" as the item that fills the case slot "TOOL". Structured knowledge corresponding to text T1 in figure 5 is determined to match the query correctly. In a traditional information retrieval system with simple keywords, text T2 is also included as extraneous items in the retrieval results, because text T1 and text T2 are not distinguishable as far as both keywords are concerned.

6 CONCLUSION

The advantages of the proposed system which constructs two kinds of secondary information before retrieval are that it utilizes them:
(1) to help users make the intended retrieval requests by showing the contents of the stored information in the desired degree of detail,
(2) to answer the retrieval requests described in various levels of generality and particularity, and
(3) to realize a kind of similarity-based retrieval.

We give a formal description for knowledge and outline of how to construct knowledge in a layered structure.

ACKNOWLEDGMENTS

This work is part of a research and development project on fifth generation computers, conducted under a programme set up by the Ministry of International Trade and Industry.

We wish to express our thanks to Dr. Kazuhiro Fuchi, Director of the ICOT Research Center, Mr. Shingi Domen, General Manager, and Mr. Koichiro Ishihara, Department Manager of the Systems Development Laboratory of Hitachi Ltd., who provided the opportunity for us to conduct the present research.

REFERENCES

[Bobrow 1977] Bobrow, D. G. and Winograd, T., "An Overview of KRL, a Knowledge Representation Language", Cognitive Science, 1, 1, pp. 3-46, 1977

[Chikayama 1984] Chikayama, T., "Unique Features of ESP", Proceedings of the International Conference on Fifth Generation Computer Systems 1984, pp. 292-298, 1984

[Janas 1979] Janas, J. M. and Schwind, C. B., "Extensional Semantic Networks: Their Representation, Application, and Generation", pp. 267-302, Associative Networks, Nicholas V. Findler (eds.), Academic Press, 1979

[Kondo 1987] Kondo, H. and Koguchi, T., "A New Approach to Knowledge Base Management Systems", Methodologies for Intelligent Systems, pp. 232-239, Zbigniew W. Ras and Maria Zemankova (eds.), North-Holland, 1987

[Rau 1987] Rau, Lisa F., "Spontaneous Retrieval in a Conceptual Information System", Proceedings of the Tenth International Joint Conference on Artificial Intelligence, pp. 155-162, Milan, Aug. 1987

[Sakamoto 1987] Sakamoto, T., "The History of Information Retrieval: Recollection and Future View", Proceedings of the 7th Advanced Database Systems Symposium (in Japanese), pp. 1-8, Information Processing Society of Japan, Dec. 1987

[Teskey 1987] Teskey, F. N., "Enriched Knowledge Representations for Information Retrieval", Proceedings of ACM SIGIR International Conference on Research and Development in Information Retrieval, pp. 43a-43g, 1987