Knowledge-based Natural Language Interface to Database

(+) K. NARUMI, (+) S. HASHIMOTO, (*) T. YOSHINO, (*) T. FUTAMURA

(+) AI Dev. Dep.
Fujitsu Ltd.
140 Miyamoto, Numazu,
Shizuoka, 410-03,
JAPAN

(*) Software Laboratory
Fujitsu Laboratories Ltd.
1015 Kamikodanaka, Nakaharaku,
Kawasaki, 211
JAPAN

1. INTRODUCTION

As databases have come into widespread use, the range of user application domains has broadened and become greatly diversified. The number of amature computer users performing nonroutine jobs is steadily inclcasing. To aid such users, We have been developing a Japanese language interface in a front-end database system. This paper explains system requirments and outlines the prototype interface we have developed.

A natural-language interface has four major requirments:

1) Expandability and robustness in sentence analysis

Many studies of natural-language and its diversification have demonstrated the practical impossibilitiy of constructing a body of syntax rules that can accept all sentences. [e.g. Marcus 80] Syntax rules muse be added and modified to suit the users and application domains. To handle the use of colloquial expression, syntax rules must be able to cover ellipses, coversational statements, and ungrammatical sentences to the point of processing gramamatical mistakes. Sentences that the system cannot accept must be analyzed in such a way that the system can explain the why to the user and teach the user the analystical processes for the user to modify syntactic rules. A system capablity of any thing less than this does not provide an acceptable level of service to users.

2) Independence from the application domain

The system must possess two types of knowledge. One that is domain-dependent and one that is domain independent. Domain-dependent knowledge must be classified clearly and replaced easily in order to guarantee the system is transportable to a new domain. Knowledge engneers (KEs) are required to structure domain-dependent knowledge for each domain, a process often requiring long hours. Yet this type of knowledge is vital to the system's language analysis because it plays an important role in semantic processing.

3) Independence from the database system.

The system must be capable of expansion and change required when the database structure is modified. The user must be able to operate the system without regard for the database system itself or schema information.

4) Easy structuring and editing of knowledge.

The different type of knowledge stored in the system must be easily structured and easily edited regardless of whether the user is a first-timer in the area or an expart in the domain.

To attain expandability and robustness in sentence analysis, we collected and expressed knowledge of Japanese language syntax in a language model composed of class-level objects. A set of syntax rule is included in class-level objects and modularized, while at the same time retaining system expandabilty. Because the parser is an expert system, it has explanation, trace and re-execution functions from the specified analysing state. To enhance robustness, the system analyzes sentences by placing emphasis on semantic analysis.

To attain independence from the application domain, we structured domain-oriented knowledge as a world model separate from other components and types of knowledge. We also simplified the expression of the domain knowledge to reduce the length of time required to construct knowledge of a domain as it relates to the database interface. More over, we designed semantic processing that uses knowledge of the domain.

To attain independence from the database system, we provided the system with knowledge of the database structure and knowledge of mapping to enable it to associate domain related knowledge with knowledge of the database structure. This enabled us to design a database independent interface. This interface enables the user to access the databse without having to be familia with schema information, for example.

To attain easy structuring and editing of knowledge, we equipped the system with a variety of knowledge base editor with interfacing corresponding to the user's level, enabling the system to be constructed and expanded easily. The prototype system and the types of knowledge it uses are covered in more detail later.

2. SYSTEM OVERVIEW

Figure 1 gives the system overview. The system uses four types of knowledge: a language model, a world model, knowledge of the database structure, and knowledge of mapping. The language model collects knowledge of the language involved Jpanese-language syntax. The world model collects knowledge of the application domain. Knowledge of the database structure includes database schema information of the database. The knowledge of mapping associates the world model with knowledge of the database structure. In system

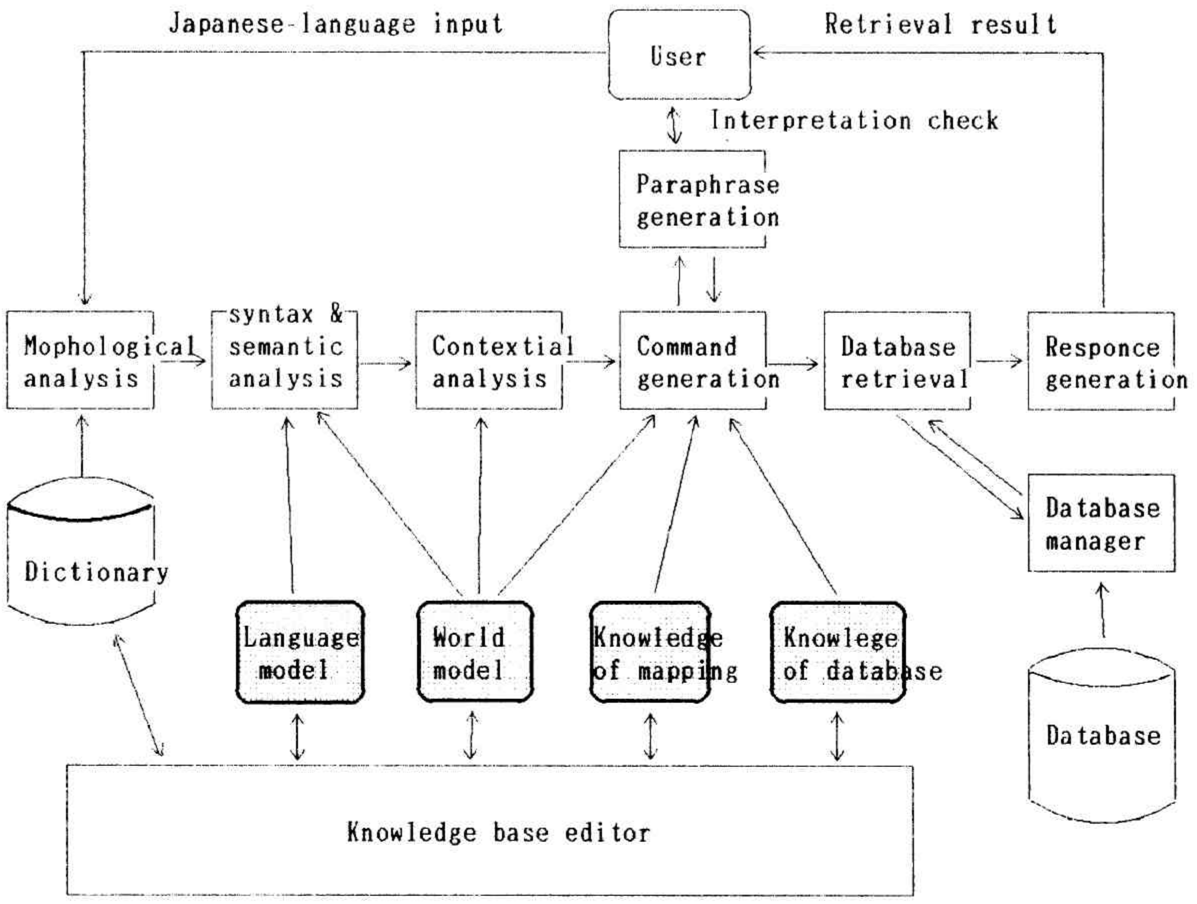


Figure 1 SYSTEM OVERVIEW

processing, The user starts by inputting a Japanese-language sentence. During morphological analysis, the system divides the input sentence into words, referencing the dictionary. During syntax and semantic analysis, the system generates a syntax analysis tree and semantic structures, referencing the language and world models. During contextual analysis, the semantic structure obtained from the input sentence is qualified by the semantic structure obtained from the previous sentence.

In command generation, the system generates command character string, referencing to the knowledge of the database structure and of mapping. During paraphrase generation, the system notifies the user of the interpretation of the input sentence thus far and asks the user weather the interpretation is OK or not. During database retrieval, the system accesses to database. During data response generation, the system notifies the user of retrival results. The system is described using the object-oriented language MINERVA. [Sato 86] The different types of knowledge are defined as class-level objects of the language.

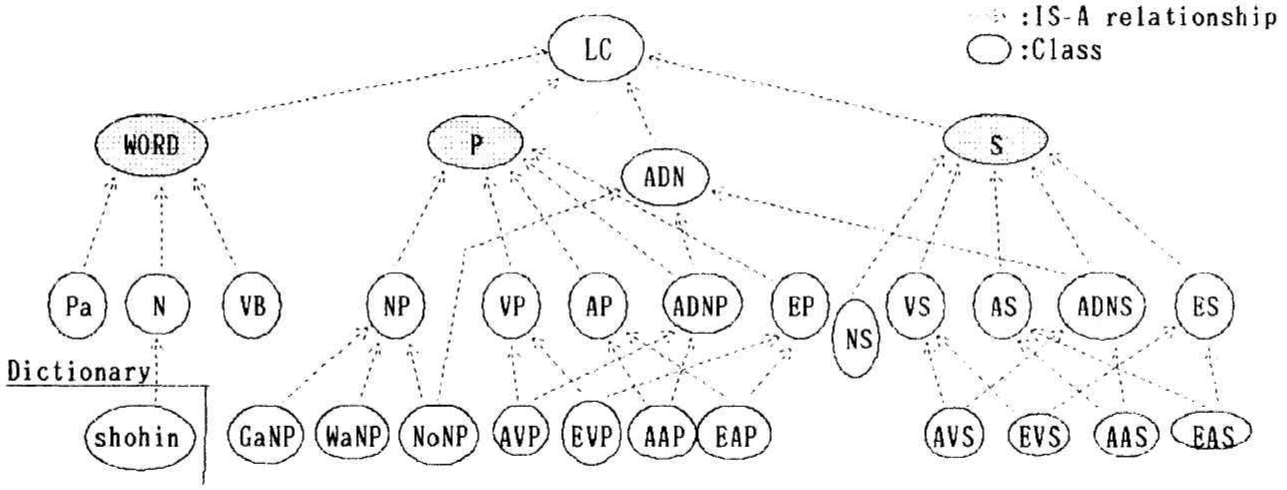
To optimize system construction, the language model is edited useing a rule editor, the world model a world-model-editor, dictionary a word editor, and MINERVA an object editor.

3. ANALYSIS AND KNOWLEDGE OF SYNTAX AND SEMANTICS

This chapter explains the two types of knowledge -- the language model, world model -- and semantic rules used by the syntax and semantic analysis.

3.1 Language model

The language model (LM) is a hiararchical arrangement of linguistic concepts, such as words, phrases, and sentences, classified based on their attributes. It is also a hiarachical set of rules for parsing input sentences. Figure 2 shows a part of the language model, which consists of class-level objects, each of which represents a linguistic concept. Dashed arrow indicates IS-A relationships.



AAP: adnominal adjective phrase
AAS: adnominal adjective sentence
ADN: adnominal
ADNP: adnominal phrase
ADNS: adnominal sentence
AP: adjective phrase
AS: adjective sentence
AVP: adnominal verb phrase
AVS: adnominal verb sentence
EAP: adnominal phrase with end-form
EAS: adnominal sentence with end-form
EP: phrase with end-form
ES: sentence with end-form
EVP: verb phrase with end-form
EVS: verb sentence with end-form
GaNP: "ga" noun phrase
LC: linguistic concept
N: noun
NoNP: "no" noun phrase
NP: noun phrase
NS: noun sentence
P: phrase
Pa: postpositional particle
S: sentence
VB: verb
VP: verb phrase
VS: verb sentence
Wanp: "wa" noun phrase
WORD: word

Figure 2 Part of a language model

The class "linguistic concept" (LC in Figure 2) has three subclasses: the word (WORD), the phrase (P), and the sentence (S). These are basic concepts recognizable as units that form Japanede-language sentences.

Japanese-words are classified based on their grammatical categories. That is, the LM has, as subclasses of the word class, a verb subclass (VB), a noun subclass (N), and a postpositional particle subclass (Pa). A Japanese word such as "shohin (merchandise)" is represented as the "shohin class" linked as a subclass to the N class.

Phrases are classified based on the independent word in the last bunsetsu. The phrase class (P), followed by its subclass -- the noun phrase (NP), verb phrase (VP), and adjective phrase (AP), - refrects the classification. Phrases are also classified based on the conjugation of the words that end them. Thus, the system has an adnominal phrase class (ADNP) and a phrase with end-form class (EP). The noun phrase (NP) is classified into a " ga" noun phrase (GaNP), or "no" noun phrase (NoNP) -- that is, the particle (ga, wa, or no) that terminates a phrase.

A set of syntactic rule is linked to some LM classes. Such a class usually contains two or three rules. A rule consists of two components -- a conditional component that checks whether the rule can be applied to the input parse tree having a meaning structure, and an action component that transforms the input parse tree and produces a new meaning structures. Figures 3.1 and 3.2 gives examples of LM class rules.

```
("no"-noun-phrase

   (meta class     (rule:set))
   (super class    (noun-phrase)(adnominal))
   (control        (doall)
   (rule
        ("no"-noun-phrase-with-L1-neq-adnominal ;RULE NAME
          (if   (and (L1 neq adnominal)
                     (R1 eq noun-phrase)
                     (semantic check between C and R1)))
          (then  generation of new state
                 transformation of syntactic tree
                 setting a phrase priority))
        ("no"-noun-phrase-with-L1-eq-adnoun    ;RULE NAME
          (if
                        . . . . . .
          (then
                        . . . . . .
                                             ))))
```

Figure 3.1 Rules in the "no" noun phrase class

```
(noun-phrase
          . . . .
(rule
  (noun-phrase   ;RULE NAME
    (if      (there is noun phrase ahead))
    (then    creation of a new state
             left shift for buffer
             setting a parse priority)))
```

Figure 3.2 Rule in the noun phrase class

In this language model, there are two types of rule inheritance. In the first, the system searches for rules in its super class if there are no rules in an activated class object. In the second, rules of a particular class are combined with those of its super class. When the "no" noun phrase class (Figure 3.1) is activated, its rules and its super calss rule (Figure 3.2), which is in a noun phrase class, are merged to form a set of rules. The rule in the noun phrase class is also used for the "ga" and "wa" noun phrase classes which, having no rules, inherit the rule in the noun phrase class.

3.2 World model

The world model expresses knowledge of the application domain, and consists of nodes and links. The node expresses "things" and "events" of the domain. These are expressed as class-level objects and are called classes. The links express relationships between things and events. To obtain knowledge easily, however, our system has only two types of relation -- attribute and IS-A -- from the viewpoint of expressing domain knowledge for database retrieval. Figure 4 is an example of the world model for sales. A system tool generates a prototype of the world model from the database schema. The KE can then complete the world model simply by modifying the prototype of the world model.

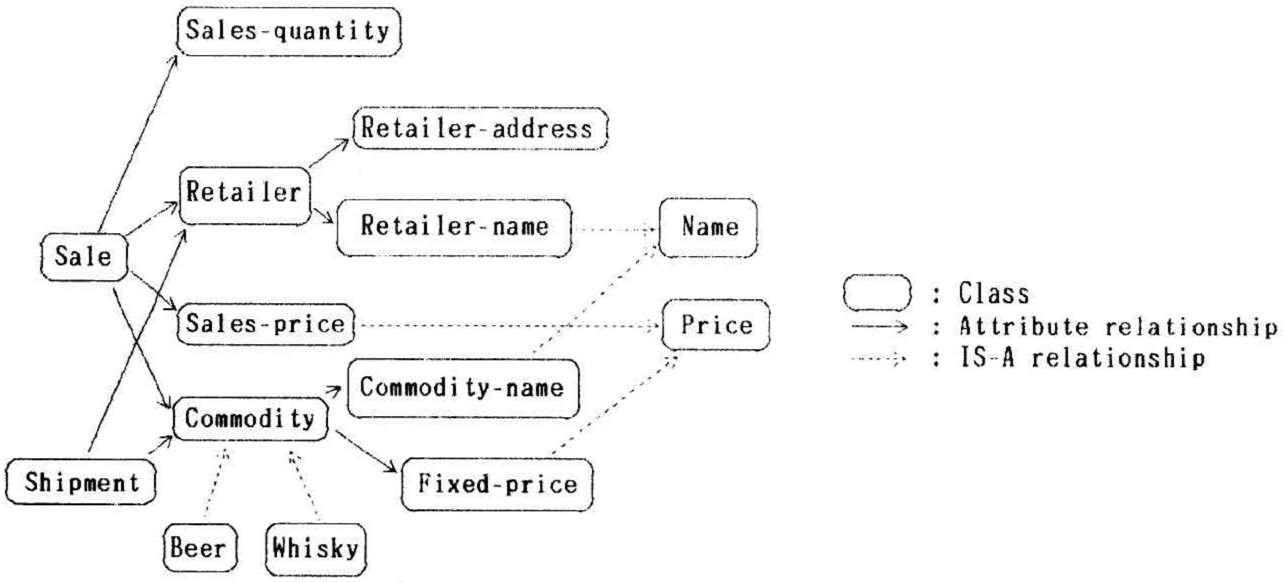
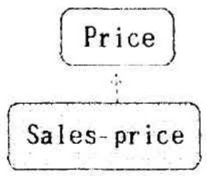


Figure 4 Example of world model

3.3 Semantic interpretation rule

In our system, the world model is used to express not only domain knowledge, but also a semantic interpretation. Semantic interpretation is represented by a network. The network consists of world model class instances corresponding to the phrases in the input sentences based on the relationship to the attributes in the world model. Semantic interpretation has two basic rules -- specialization and connection -- that check the relationship between the classes corresponding phrases and interpret the meaning of a new phrase which has the phrases as subtrees.

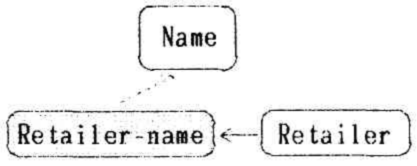
The specilization rule references the IS-A relationship in the world model. In the input sentence, there are two phrases where a syntactic qualification is established and each phrase corresponds to a class in the world model. If there is an IS-A relationship between these two classes, the specialization rule selects the lower class for the interpretation of the combined phrase, because the subclass has a more specific, and restricted meaning than the super class. Figure 5 is an example of the specialization rule in which the phrase "hanbaikakaku" corresponds to the sales price class, and "200 yen" corresponds to the price class. The specilization rule selects the sales price class as



Hanbaikakaku ga 200 yen

(The sales-price is 200 yen.)

Figure 5 Example of the specialization rule



Hanbaiten no namae

(Retailer name)

Figure 6 Example of the connection rule

the overall meaning.

The connection rule references the attribute relationship in the world model. If there is a syntactic qualification between two phrases of the input sentence and there is an attribute relationship between two classes corresponding to the above phrases, the connection rule selects the class corresponding to the syntactically qualified phrase for the interpretation of the combined phrase. Figure 6 is an example of the connection rule, in which the phrase "hanbaiten" corresponds to the retailer class, and "namae" corresponds to the name class. The connection rule selects the retailer name class as the overall meaning.

4. COMMAND GENERATION KNOWLEDGE

The database system uses two types of knowledge -- that of the database structure and of mapping -- to generate command strings. Knowledge of the database structure includes the database schema information. Knowledge of mapping is knowledge of corresponding to the world model class to the database structure.

4.1 Knowledge of the database structure

In knowledge of database structure, database configuration, table configuration, field data attributes, join information, and information for paraphrase generation, are described. This knowledge consists of class-level objects like other types of knowledge. These classes are classified into

- a class having a knowledge of the database , called the database class;
- a class having knowledge of tables, called the table class; and
- a class having knowledge of fields, called the field class.

The followings are described for the database class: Database name, the table class names corresponding to the tables in the database, and the database management system name that manages the database. The followings are described for the table class: Table name, the field class names corresponding to the tables, and join requirments of the other tables. The field name and data attributes are described for the field classes. Figure 7 is an example of the table class, and Figure 8 is an example of the field class.

```
(defobject  TABLE%GOODS
   (@level   (class-level))
   (@super   (DB%TABLE))
   (@class   (%ENTITY%))
   (@field   (FIELD%GOODS%CODE
              FIELD%GOODS%NAME
              IELD%GOODS%PRICE))
   (@name    ("GOODS")
   (join     (TABLE%SALE
               (FIELD%GOODS%CODE =
                FIELD%SALE%MCODE))))
```

Figure 7 Example of a table class object

```
(defobject  FIELD%GOODS%PRICE
   (@level       (class-level))
   (@super       (DB%FIELD))
   (@class       (%ENTITY%))
   (@p-table     (TABLE%GOODS))
   (@name        ("CODE"))
   (@type        (SINT))
   (@size        (2))
   (@dcp         (0))
   (scale        (1))
   (paraphrase (" 定価 ")))
```

Figure 8 Example of a field class object

4.2 Mapping knowledge

The world model classes correspond to (1) a table, (2) one field, or (3) two or more fields.

The correspondence is described in mapping classes. The world model class corresponds to nothing or to one mapping class, because the world model class includes the class expressing things and events that have nothing to do with the database.

The corresponding table or field class names are described in the mapping classes. A mapping class for a table also has field class names which are available when the WM class for it is a user's retrieval target. Figure 9 is an example of the mapping class.

```
(defobject  MAP%TABLE%SALE
    (@level      (class-level))
    (@super      (DB%STORAGE))
    (@class      (%ENTITY%))
    (map         (TABLE%SALE))
    (return      (FIELD%SALE%CODE  FIELD%SALE%DATE  FIELD%SALE%VOLUME)))
```

Figure 9 Example of the mapping object

5. APPLICATION AND EVALUATION

We applied our prototype system to five domains, which included domains on - real estate information, - sales information, and - drug test database, and so on

Our system was able to meet the four requirment for natural language processing outlined in section 1. Taking the domain involving the drug test database as an example, we attained the following results:

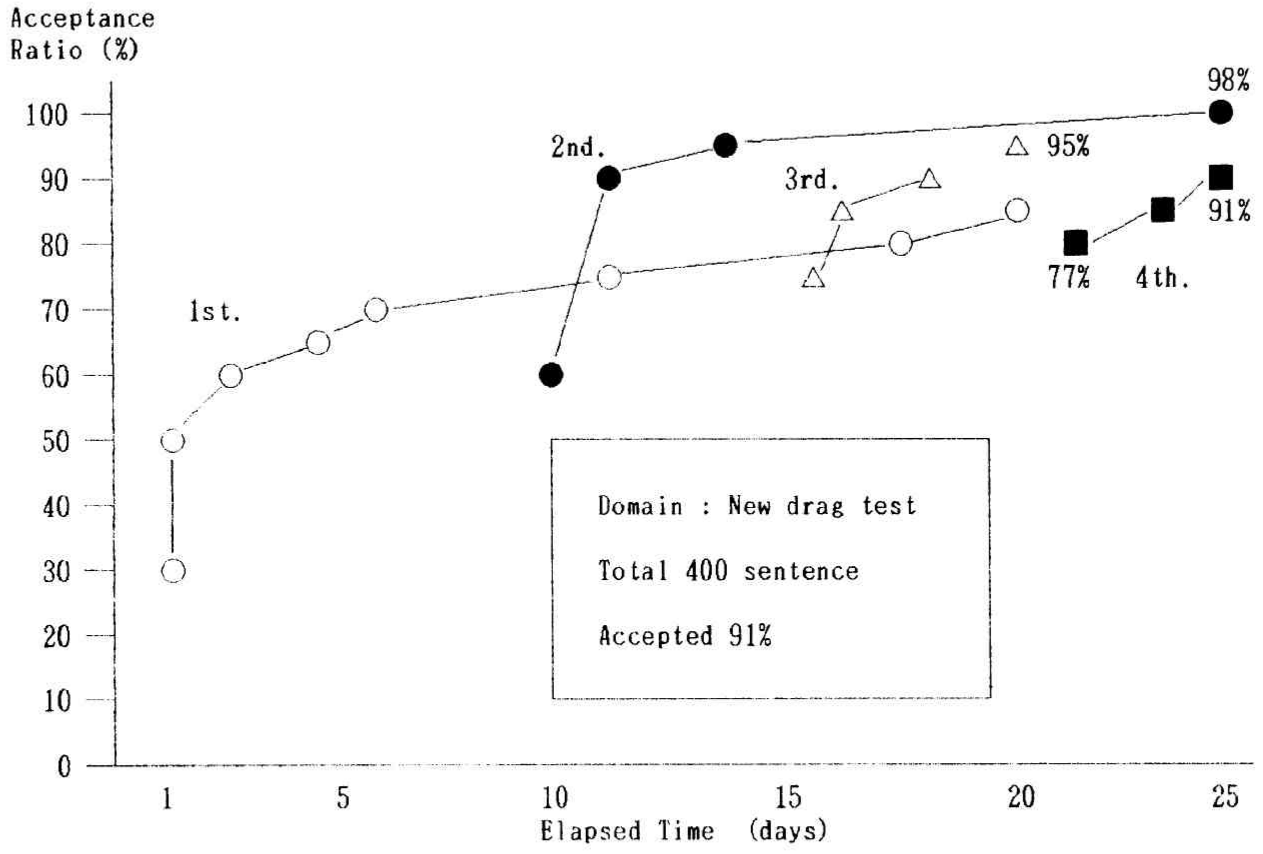


Figure 10 Evaluation of natural language interface

The world model for drug tests consists of about 140 classes. Its database consists of 32 tables. Some 400 interrogative sentences are used for evaluation, together with about 360 words. It took about half of a month to construct these types of knowledge, including education of the world model.

In the system evaluation, 400 sentences were divided into four groups, and blind test was conducted to check the number of successfull sentences without system modification. The full-capability test was repeatedly conducted to pass unsuccessful sentences. Figure 10 is an evaluation of our system.

The success ratio for the sentences reached 91% in less than one-month of testing. The initial success ratio of four tests has risen each time for a short period. This result is very good, considering the system's expandability and practical use.

6. SUMMARY

6.1 Related Work

This section compares our system with several database retrieval systems using a natural language that have been developed.

Many systems (eg., Walz 78 and Hendrix 78) include domain knowledge in sentence analysis rules, which prevents complete system transportability. To overcome this problem, we clearly separated liguistic knowledge from domain knowledge. Ginsparg (GINSPARG 83) uses a similar approach, in that ungrammatical sentences are semantically interpreted using the semantic model. The semantic interpretation rule, however, is not clarified in Ginsparg's system. To make semantic interpretation rules clear, we prepared two rules. There are commercialy available systems, such as INTELLECT [Harris 78] and Q&A. [Walden 86] For INTELLECT, it is difficult for the user to customize the system because the user cannot reference detailed processing. Q&A is superior in acquiring phrases and vocabularies but cannot simultaneosly access one file at the same time.

6.2 Conclusion

We described a Japanese-language database retrieval system that uses a language model, world model, knowledge of the database structure, and knowledge of mapping. The language model contributes to the expandability and robustness of sentence analysis. The world model contributes to the system's independence from the application domains. Knowledge of the database structure and of mapping guarantee the interface system's independence from the database system.

Our system provides superior explanations in its analysis and debugging because its analysis component is constructed as an expert system. The system also supports several knowledge base editors that make it easier for the KE to structure knowledge.

[Ginsparg 83] Ginsparg, J. M., 1983 A robust Portable Natural Language Data Base Interface, proc. Conf. Applied Natural Language Processing, PP.39 - PP.45

[Harris 78] Harris, L. R. 1978 The ROBOT System: Natural Language Processing Applied to Database Query, Proc. ACM Annual Conf., Vol.1, ACM New York, N.Y., PP.165 - PP.172

[Hendrix 78] Hendrix, G. G. Sacerdoti, E. D., Sagaluwics, D. and Slocum, J. 1978 Developing a Natural Language Interface to Complex Data, ACM TODS, Vol.3, No.2, pp105 - pp147

[Marcus 80] Marcus, M.P 1980 A Theory of Syntactic Recognition for Natural Language . MIT PRESS, Cambridge, Mass. and London

[Sato 86] Sato,S and Sugimoto,M 1986 Artificial Intelligence, FUJITSU Scientific and Technical Journal Vol 22, No.3, pp 139 - pp181

[Walden 86] Walden, J. 1986 Getting the Most from Q&A, Osborne McGraw-Hill, Berkely, Calif.

[Walz 78] Walz, D. L. 1978 An English Language Question Answering System for a Large Relational Database, Communication of ACM, Vol.27, No.7, pp526 - pp539