

Word Segmentation based on Database Semantics in NChiq1*

Xiaofeng Meng(孟小峰), Shuang Liu(刘爽), and Shan Wang(王珊)

College of Information, Renmin University of China, Beijing 100872
xfmeng@public.bta.net.cn, swang@mail.ruc.edu.cn

Abstract

In this paper we present a novel word-segmentation algorithm to delimit words in Chinese natural language queries in NChiq1 system, a Chinese natural language query interface to databases. Although there are sizable literatures on Chinese segmentation, they can not satisfy the particular requirements in our system. The novel word-segmentation algorithm is based on the database semantic, namely Semantic Conceptual Model (SCM) for specific domain knowledge. Based on SCM, the segmenter labels the database semantics to words directly, which eases the disambiguation and translation (from natural language to database query) in NChiq1.

Keywords: database query, natural language processing, word segmentation, and disambiguation

1. Introduction

A natural language interface to databases (NLIDB or NLI) is a system that allows a user to access information stored in a database by typing requests expressed in some natural languages, such as Chinese and English. Since the early 1960s, much of the research on NLIDB has been motivated by its potential use for communicating with DBMS. NChiq1, the Chinese natural language interface to databases, is developed for Chinese novice users.

The initial step of any language analysis task is to tokenize the input statement into separated words with certain meaning. For many writing systems, using white space as a delimiter for words yields reasonable results. However, for Chinese and other systems where white space is not used to delimit words, such trivial schemes will not work. Therefore, how to segment the words in a Chinese natural language becomes an important issue.

Most of literatures [2,3,5] on Chinese segmentation are rooted in the natural language processing (NLP). However, NLIDB, as one of the typical application domain of NLP, has its own processing features. It is very possible to apply the achievement in NLP to NLIDB, but it may not be the best way.

The goal of language processing in NLIDB is to translate natural language to database query. So it's not necessary to understand the deep structure of sentences, as long as we can reach the

* This work is supported by the National Natural Science Foundation of China under grant No 69633020.

translation goal. Generally, the conventional segmentation methods mark the word with Part of Speech (POS) such as noun, verb, adjunct, and pronoun etc. These methods can not reflect the database semantic associated with the words. However, in NLIDB we do not care which class a word belongs to, but care what semantic it represents in database. According to this design principle, we give a novel word segment method based on the database semantic. The advantage of the word-segmenter is simple and efficient. The performance was 99% above precision to our real test queries [8].

This paper is organized as follows: In Section 2, we introduce the semantic conceptual model and its extracting in NChiqI, which is the foundation of our novel segmenter. In Section 3, the word segmenter based on database semantic is presented. In Section 4, the system structure and implementation is shown. In Section 5, we give the conclusion about our method.

2 Semantic Conceptual Model and Its Extracting

We contend that knowledge is needed to interpret natural language queries to a database. In this section we argue in favor of a knowledge-based approach to natural language database access.

Actually, there are many places where it is necessary to use domain knowledge in order to interpret the queries, such as word segmenting, ambiguity resolving, ellipsis handling, etc. Some AI research has considered this problem for the database domain. The database community has not dealt with this problem yet[4]. In this paper we give a kind of knowledge representation and its extracting approach from the database perspective. In next section we will explain how the knowledge to support the word segmentation in NChiqI.

In NChiqI, we depict a semantic conceptual model, which combine the linguistic knowledge and database knowledge.

Definition 1 a Semantic Conceptual Model (SCM) is a specific domain knowledge set, formally,

$SCM := (C, L)$ where C is a set of database concepts such as entity, attribute and relationship; L is the set of linkages among them. Generally, there are two types of modifying relationship and three types of verb relationship. To be specific, the two types of modifying relationship are:

- (1) The Direct modifying relationship among entities, including the attribute-modify-entity and the entity-modify-entity.
- (2) Reference modifying among entities, which is engendered by foreign key reference relationship among entities.

The three types of verb relationship are the verb combination of relations vs. entities, attributes vs. entities, and entities vs. entities.

Modifying relationship in SCM can mainly determine the combination relationship of nouns, while verb relationship presents the usage of verb phrase. It is extremely useful once provided with these semantic relationships in the analyzing of natural language queries.

SCM extraction is the process of acquisition of sufficient domain knowledge and

reconstruction of domain information especially the semantic and usage of words in order to extend the ability of language processing. Extraction of SCM means to make up the gap between the realistic world knowledge and database schema. Its goal is to provide necessary semantic information for the language processing in NChiq[6,7].

All the result of extracting is stored in dictionaries, which are the structure representation of SCM. NChiq divides these dictionaries into two types: general and specific.

- General dictionaries are independent on application and it is also the core dictionary of the system. General dictionary records the semantic of most commonly used words, such as pronoun, conjunction, quantifier, interrogative words, etc.

- Specific dictionaries record the semantics and usage of words, which are usually used in a specific application domain. When the system is being transported to another domain, these dictionary must be reconstructed. Generally, the specific dictionary includes:

- Dictionary of entity semantic, which records table name and its synonym; the semantic of the modifying attributes, etc.

- Dictionary of the attribute semantic, which records the attribute name, synonym, hint, modifiers and the semantics of constraint, etc.

- Dictionary of the verb semantic, which records the semantic and usage of the verb, etc.

- Domain knowledge rules.

The relationship of these dictionaries is shown in Figure 1.

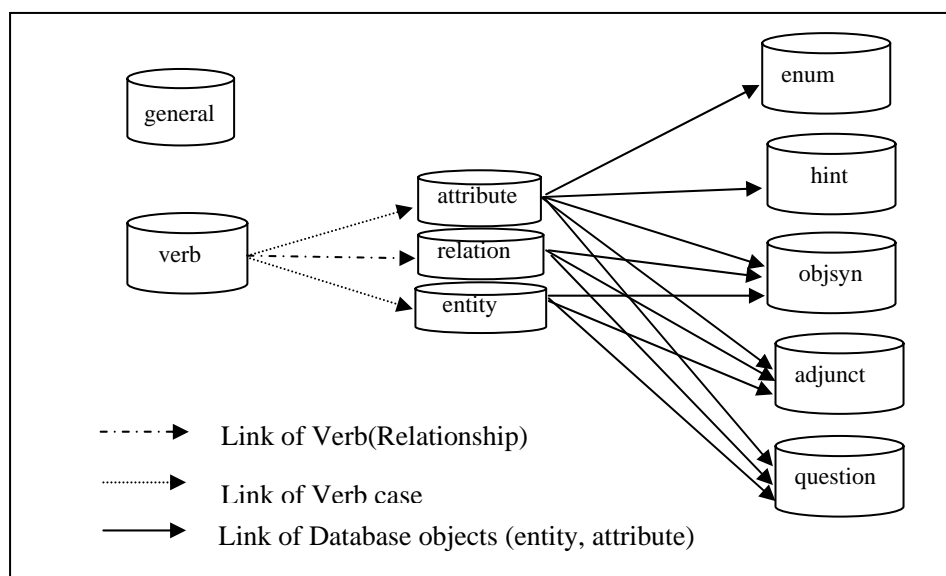


Fig. 1 Relationship among the Dictionaries in NChiq

3 Chinese Word Segmentation Based on SCM

Basically, the content in database is explicit and limited, and a database query is to operate the objects and data in database. So the most of words in natural language queries can find their counterpart in database, e.g., a noun word in a query must be related to entity, attribute, value, or

their synonym; verbs may be commands (query, modify, delete etc.) or domain relationship. For example, noun "student" vs. the student entity in database concept; verb-phrase "take a course" vs. the SC relationship, which contains attributes sno and cno representing the subject (student entity) and object (course entity) of the verb respectively. It's obvious that the words used in natural language queries are limited in database concept model and have the one to one relations with the database objects. As mentioned above, SCM establishes such specific relationship among the linguistic knowledge and database knowledge. So using SCM to conduct the word segmentation is reasonable and feasible. Furthermore, it can assign the database semantic to words to ease the disambiguation and translation.

3.1 The Token and Semantic Description of Words

As we known, there are two different types of words: general words and specific words in our semantic segmentation system (they are stored in general dictionaries and specific dictionaries respectively). General words have no special database semantics such as numeral, measurement, preposition, conjunction and auxiliary word. The tokens of these words represent as NUMERAL, MEASURE, PREPOSITION, CONJUNCTION, and AUXILIARY. In addition, some words are regarded as general one too such as query word, be, has, aggregation operator, compare operator and punch. The tokens of such words are QUERY, BE, HAS, AGG, COMP, PUNCH.

Specific words include entity, attribute, relationship (verb), database value, attribute hint, pronoun (representing entity) and question word (representing entity and attribute). The tokens of them are TABEL, ATTR, VERB, VALUE, HINT, PRONOUREN and QUES_TABLE respectively.

Let Ω denote all the tokens described above.

Definition 2. The database description of a given word in a specific application domain (DOM) is defined as $D(\omega:DOM)=(o,[t,c])^1$, where

ω represents the given word;

o represents the corresponding database object (entity, relation, attribute) of ω ;

t represents the data type (data type, length, precision) of a database object;

c represents the verb case of a database object.

$D(\omega:DOM)$ also denoted as $D(\omega)$ or D for short. In $D(\omega:DOM)$, the information represent by o determine t and c . For convenience, we can use the database object of a word ω to indicate its database description.

In a natural language query, a word may have more a few database descriptions. For example, the database object of word "name" can be either student's name or teacher's name. To handle the case, we have the following definition.

Definition 3 Database semantics set of a specific word ω is the set of database descriptions

¹ The items in [] are optional.

that the word ω corresponds to based on the SCM of a specific domain (DOM), denoted as:

$$S(\omega:DOM)=\{D_1,D_2,\dots,D_n\}.$$

For simpleness, we can define the database semantic set by word's database objects:

$$S(\omega)=\{O_1(\omega),O_2(\omega),\dots,O_n(\omega)\}$$

Based on the aforesaid definitions, we can give the token description in NChiq1 as follows.

Definition 4 The database semantic token description of a word ω consists of the corresponding word token and database semantic set based on SMC for a specific domain. It can be denoted as:

$$\Theta(\omega:DOM)=(\lambda,S), \text{ where } \lambda \in \Omega, \text{ and } S \text{ is the database semantics of } \omega, \text{ namely } S(\omega).$$

For example, “找出所有 19 岁学生的学号和姓名。”(Find the registration number and name of all nineteen years old students). Its semantic token in NChiq1 is shown in figure 2.

查找 # {(QUERY,)} 所有 # {(QUALIFY, ALL)} 19 # {(NUMERAL, 19)}
 岁 # {(HINT, dba, student, age, INT, 4), (HINT, dba, teacher, age, INT, 4)}
 学生 # {(TABLE, dba, student)} 的 # {(AUXILIARY,)}
 学号 # {(ATTR, dba, student, sno, INT, 4)} 与 # {(CONJ, AND)}
 姓名 # {(ATTR, dba, student, name, CHAR, 10), (ATTR, dba, teacher, name, CHAR, 10)}
 。 # {(PUNCH,)}

Fig.2 An example of database semantic token

Obviously, it is helpful for the system to translate the natural language query to database query based on the semantic token in NChqil.

3.2 Word Segmentation Algorithm

The word segment algorithm adopts the minimum matching method. In order to improve the precision of segmentation, we use the backtracking and correlation semantics determining mechanisms to process the “unknow” case. The basic idea of the algorithm is as follows: The input sentence is first divided into several substrings according to the morphology, one substring is made up of a few words. Through looking up the dictionaries, these substrings are separated into words. The backtracking and correlation semantic determining mechanism are applied to solve the ambiguity of words.

There are three functions in the algorithm: getword(), search() and retrace(). Function Getword() fetches a substring to provide to search(); function search() is responsible for looking up the lexicons to find the matched word; function retrace() tries to combine the characters in the unprocessed word with the pre-word, to find new segmentation. If failures, the word will be labeled UNKNOW.

Figure 3 shows the processing flow of search().

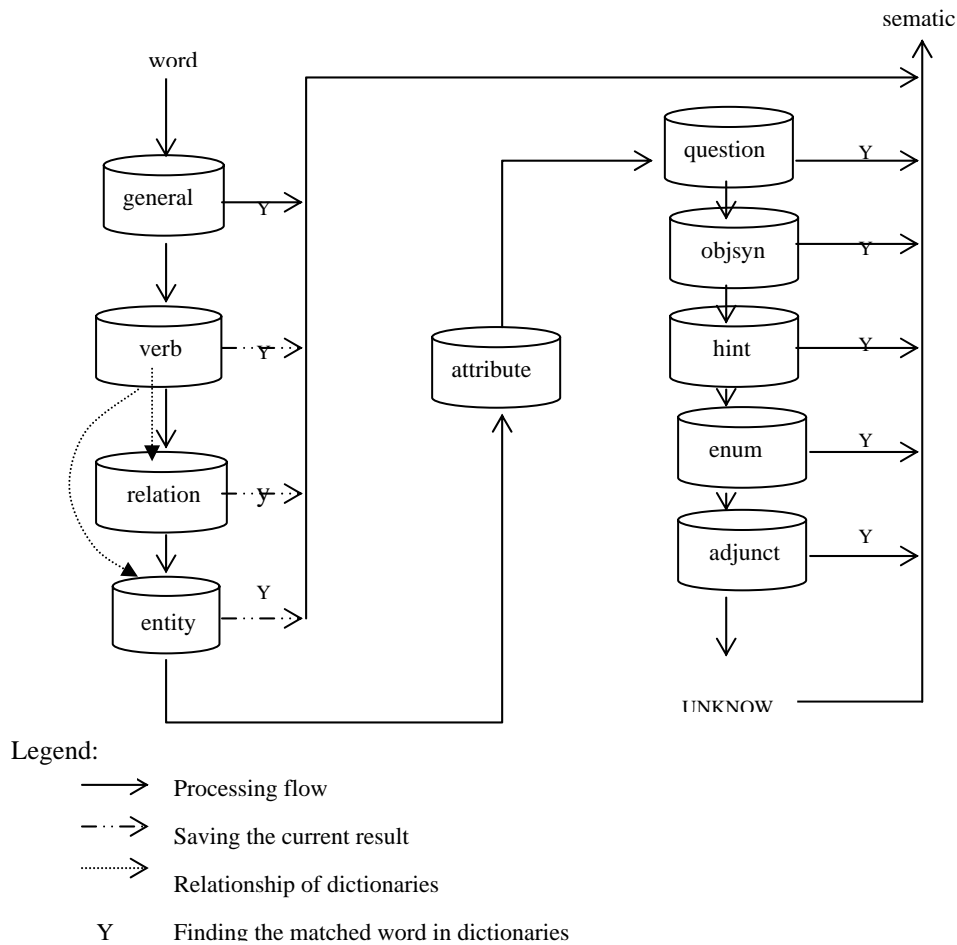


Fig. 3 the Processing Flow of search() Function

The word segmenter may meet many ambiguous and UNKNOW words in the statement. In the next part of this paper, we will introduce the correlation semantics determined method to disambiguate the word, and identify the database semantics of the unknown word.

3.3 . Disambiguation in NChiq1

There are two types of ambiguities in natural language query:

1. Syntactic ambiguity

One string may have different segmentation result. But under certain context, only one result is right. We call such ambiguity as syntactic ambiguity.

For example: in the query “查找供应商品的供应商”(Find the supplier who supplied some items), “供应商” has two possible segmentation: “供应(supply)/商(UNKNOW)” and “供应商(supplier)”. This is a syntax ambiguity. Based on the minim segmentation method, the”供应商” is segmented as:”供应/商”, through the backtracking mechanism we can get the new word”供应商”.

2. Database semantic ambiguity

The DB(database) semantic ambiguity means that one word may correspond to several DB

semantics, namely there may be several elements in the semantic set $S(\omega:DOM)$. In natural language query, those words having DB semantic ambiguity are all specific words. For example, the word “name” corresponds both to teacher.name and student.name. But in a query statement, the word “name” should have one and only one DB semantic.

Correlative semantic disambiguation method tries to disambiguate the word by reference the context of the ambiguity word in query statement. Before introducing the method, we give some related definitions first.

Definition 5 Ambiguity Word. A word ω is an ambiguity one in DB, if semantic set $S(\omega)$ has more than one elements, denoted as ω^a .

Definition 6 Normal Query Statement. One query statement is normal, if there are no such ambiguity word ω^a existing: ω^a do not modify any un-ambiguity word, at the same time there is no un-ambiguity word modify ω^a . A normal query statement is denoted as ξ .

For example, in the query statement” Find name”, the word “name” corresponds both to teacher.name and student.name, there are no other words which have definitive DB semantics existing to identify the “name’s” DB semantics, so this query statement is not a normal one.

Definition 7 Correlated Word. Word ω_1 correlates with word ω_2 , when the DB objects $O(\omega_1)$ and $O(\omega_2)$ of the word ω_1 and ω_2 have linkage in SCM, denoted as $\gamma(\omega_1) = \omega_2$.

Definition 8 The correlated word of word ω in a query statement are the word which correlates and has directive dependent relation with word ω in normal query statement ξ , denoted as $\gamma_\xi(\omega)$. Generally, a word ω has more than one correlated words in a normal query statement, called **correlated word set in a query statement**, defined as: $\Gamma_\xi(\omega) = \{\gamma_{\xi 1}(\omega), \gamma_{\xi 2}(\omega), \dots, \gamma_{\xi k}(\omega)\}$.

Definition 9 Correlated Semantic Set. The correlated semantic set of a word ω is a collection composed of the DB object(s) $O(\omega)$ of the correlated word set $\Gamma_\xi(\omega)$, defined as: $\mathfrak{S}(\omega) = \{O(\gamma_{\xi 1}(\omega)), O(\gamma_{\xi 2}(\omega)), \dots, O(\gamma_{\xi k}(\omega))\}$.

Definition 10 The minimum semantic set of ambiguity word ω^a in a normal query statement is the intersection of the correlated semantic set of every correlated words. Defined as: $\Lambda(\omega^a) = \mathfrak{S}(\gamma_{\xi 1}(\omega^a)) \cap \mathfrak{S}(\gamma_{\xi 2}(\omega^a)) \cap \dots \cap \mathfrak{S}(\gamma_{\xi k}(\omega^a))$

Lemma 1 If a query statement is normal, the ambiguity word in this statement must have correlated words which can be used to determine the DB semantics of the ambiguity word.

Proof: See Definition 5 and Definition 6.

Theorem 1 The minimum semantic set $\Lambda(\omega^a)$ of ambiguity word ω^a is not null in normal query statement ξ .

Proof:

For ξ is normal, the ambiguity word ω^a must has some definitive DB semantics, it is presumed as $O(\omega^a)$.

Let the correlated words of ω^a are $\gamma_{\xi 1}(\omega^a), \gamma_{\xi 2}(\omega^a), \dots, \gamma_{\xi k}(\omega^a)$. Then $\gamma_{\xi 1}(\omega^a), \gamma_{\xi 2}(\omega^a), \dots, \gamma_{\xi k}(\omega^a)$ must exist according to lemma 1.

The correlated semantic set of $\gamma_{\xi 1}(\omega^a), \gamma_{\xi 2}(\omega^a), \dots, \gamma_{\xi k}(\omega^a)$ are $\mathfrak{S}(\gamma_{\xi 1}(\omega^a)), \mathfrak{S}(\gamma_{\xi 2}(\omega^a)), \dots, \mathfrak{S}(\gamma_{\xi k}(\omega^a))$ respectively.

$\gamma_{\xi 1}(\omega^a), \gamma_{\xi 2}(\omega^a), \dots$ and $\gamma_{\xi k}(\omega^a)$ correlate with word ω^a . Word ω^a also correlates with $\gamma_{\xi 1}(\omega^a), \gamma_{\xi 2}(\omega^a), \dots$ and $\gamma_{\xi k}(\omega^a)$.

So $O(\omega^a) \in \mathfrak{S}(\gamma_{\xi 1}(\omega^a)), \mathfrak{S}(\gamma_{\xi 2}(\omega^a)), \dots, \mathfrak{S}(\gamma_{\xi k}(\omega^a))$ respectively.

So there are at least one element, $O(\omega^a) \in \Lambda(\omega^a) = \mathfrak{S}(\gamma_{\xi 1}(\omega^a)) \cap \mathfrak{S}(\gamma_{\xi 2}(\omega^a)) \cap \dots \cap \mathfrak{S}(\gamma_{\xi k}(\omega^a))$

$\therefore \Lambda(\omega^a) \neq \emptyset$ □

Theorem 2 The intersection of Semantic set $S(\omega^a)$ and minimum correlated semantic set $\Lambda(\omega^a)$ of ambiguity word ω^a is not null.

Proof: $S(\omega^a) = \{O_1(\omega^a), O_2(\omega^a), \dots, O_n(\omega^a)\}$

$\Lambda(\omega^a) = \mathfrak{S}(\gamma_1(\omega^a)) \cap \mathfrak{S}(\gamma_2(\omega^a)) \cap \dots \cap \mathfrak{S}(\gamma_k(\omega^a)) = \{\dots O(\omega^a) \dots\}$

$O(\omega^a) \in S(\omega^a)$

$\therefore S(\omega^a) \cap \Lambda(\omega^a) \neq \emptyset$ □

According to the definitions and theorems given above, an ambiguity word in a normal query statement has at least one correlative word, the correlative semantic set of the correlated word set can help to determine one DB semantic for the ambiguity words.

The algorithm of disambiguating is as follows.

DISAMBIGUITY (ω^a) :

Input: $\omega^a, S(\omega^a), \xi, SCM$

Output: $O(\omega^a)$

Step:

Step 1 Get the correlated words $\gamma_{\xi 1}(\omega^a), \gamma_{\xi 2}(\omega^a), \dots, \gamma_{\xi k}(\omega^a)$ of word ω^a in ξ based on SCM;

Step 2 Find the correlated semantic set of $\gamma_{\xi 1}(\omega^a), \gamma_{\xi 2}(\omega^a), \dots, \gamma_{\xi k}(\omega^a)$:

$\mathfrak{S}(\gamma_{\xi 1}(\omega^a)), \mathfrak{S}(\gamma_{\xi 2}(\omega^a)), \dots, \mathfrak{S}(\gamma_{\xi k}(\omega^a))$;

Step 3 Computing: $\Lambda(\omega^a) = \mathfrak{S}(\gamma_{\xi 1}(\omega^a)) \cap \mathfrak{S}(\gamma_{\xi 2}(\omega^a)) \cap \dots \cap \mathfrak{S}(\gamma_{\xi k}(\omega^a))$;

Step 4 If there is only one element in $\Lambda(\omega^a)$, this is the DB semantic of the ambiguous word; else go to step 5.

step 5 Do $S'(\omega^a) = S(\omega^a) \cap \Lambda(\omega^a)$, if there is one element in $S'(\omega^a)$, this is the right DB semantic of ambiguous word ω^a ; else call INTERACT(see Figure 5) to interact with the user.

3.4 Determining the Semantics of the Unknown Words

Besides ambiguous words, the system also includes UNKNOWN words, DIGITAL_VALUE words and TIME WORDS. Although the latter two kinds of words have definite types, there is still little chance to know their database semantic. Therefore, they belong to UNKNOWN words too. The existence of these UNKNOWN words is resulted from the limited content of the lexicon. Given the enormous values in the database, it is impossible to store all of them into the dictionary

for limited storage and efficiency. We could only put typical values and hints into the dictionary. Usually we search the "values", combined with relevant semantic ascertainment methods to determine the UNKNOWN words. The processing is shown as Figure 4.

As shown in Figure 4, we start searching from the HINT dictionary and ENUM dictionary sequentially. If the matching word with unique meaning is found, the process stops. Otherwise, for example, no matching word found or some ambiguous word appearing, the process should call relevant semantic methods to ascertain the UNKNOWN word. If there is still no way to do that, it should search the database or interact with users.

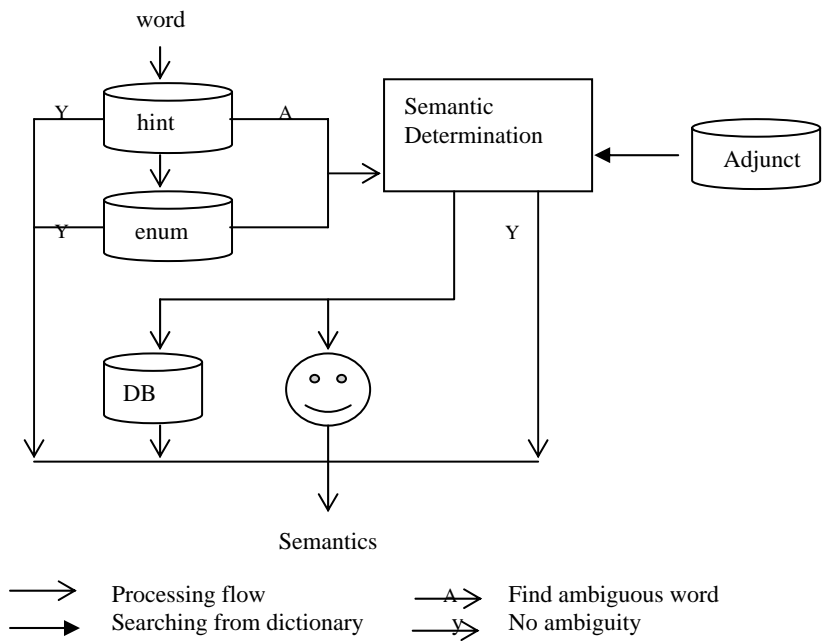


Fig.4 Determining the Semantics of UNKNOWN word

3 . 5 The Disambiguating rules in NChiq1

We give some general rules to eliminate the syntactic ambiguity, not the semantic ambiguity, in segmenting words. Compared with the semantic ambiguity, these syntactic ambiguities have certain regularity. These rules are good complementary to the above methods. Following are some typical rules in NChiq1.

- **VALUEHINT_RULE**: dealing with adjacent values and hints. If the single value and hint appears adjacently, and the data type of the value is same with data type of the hint, then they are combined.

- **UNKNOWNHINT_RULE**: dealing with adjacent UNKNOWN words and hints. If a hint is adjacent with an UNKNOWN word and a VALUE at the same time, but there is no way to combine the hint and VALUE, then the hint and the UNKNOWN word are combined.

- **SIMILARITY-MERGE_RULE**: dealing with adjacent words with same type and semantics. If the adjacent words have same tokens, they should be combined according to their original sequence first. Then it looks up the dictionary to get the semantic set of this new word. If

the new semantic set includes their original semantics, then the combination is right.

4. System Structure and Implementation

Figure 5 illustrates the runtime architecture of the Word Segmentation based on SCM in NChiq1 (called DBSWS, DB Semantic Word Segmentation). DBSWS is made up of segmentation procedure and lexicons. There are three different kinds of dictionary in the system:(1)general lexicon (2) specific lexicon (3) SCM lexicon (refer to section 2 for the content of the lexicons).

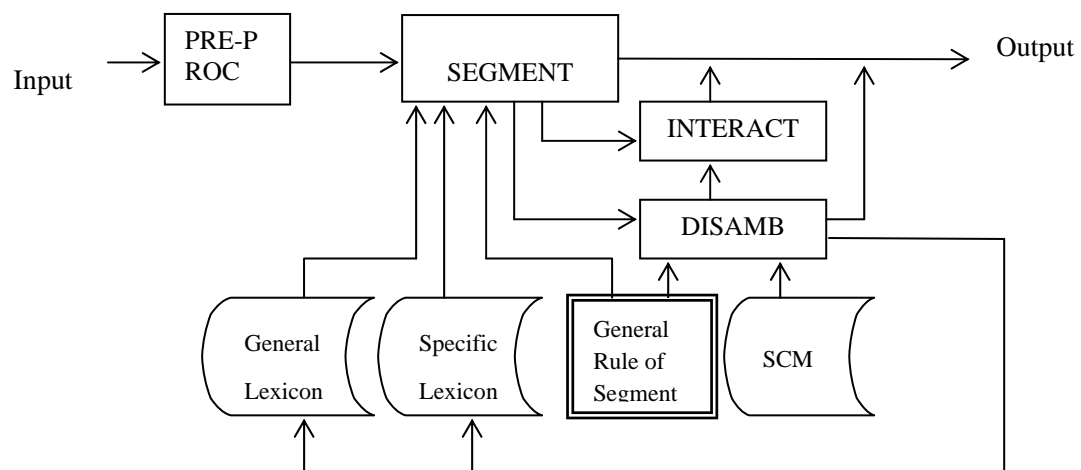


Fig. 5 The System Structure of DB Semantic Segmentation

There are four modules in DBSWS:

1. Pre-processing procedure: PRE-PROC

This procedure divides the query statement into several independent, disposable strings according to the morphology in the statement (such as: punch, digital, etc.). The independent strings are linked in a link structure.

2. Segment procedure: SEGMENT

In this procedure the sub-strings generated by PRE-PROC are segmented according to the word segmentation algorithm described in section 3.

3. Disambiguating procedure: DISAMB

The function of this procedure is to use the information provided by segment procedure, and to call the correspondent rule to disambiguate or to identify those words that can not be handled by SEGMENT. Through searching the semantic concept lexicon it can determine the “problematic” words (refer to Section 3 for more details).

4. Interactive procedure: INTERACT

When DISAMB procedure fails to disambiguate word, the interaction between system and user is helpful. There are following forms of the interaction between system and user:

- a . User gives the type of UNKNOW word;

b . User selects one word token from a list provided by the system.

5 Conclusions

In NChiq1 system, we present the Semantic Conceptual Model (SCM) for a specific domain knowledge. Based on SCM, we depict a novel word-segmentation algorithm to delimit words in sentences. Compared with general algorithms, the new method has following advantages:

- 1) It is convenient to perform database query translation based on the segmented words with database semantics;
- 2) It provides a good ground to disambiguate words;
- 3) It can easily handle those attribute values (e.g. name, address) based on database semantics;
- 4) Do not need very large dictionaries as general systems done.

In sum, the new approach is simple and efficient to our special requests for the natural language query processing in NChiq1.

References

- [1] Copestake A., Jones K S. Natural Language Interfaces to Databases, *The Knowledge Engineering Review*, 1990, 5(4):225-249.
- [2] Yu S W. The ambiguity in Natural language and the Strategy in Machine Language. *J. Of Chinese Information*, 1989, 3(2).
- [3] Feng Z W. Computer Processing to Natural Languages. *Shanghai Foreign Education Press*, 1996.
- [4] Cercone N, McCalla G. Accessing Knowledge through Natural Language. *Advances in Computers*, 1986, 25:1-99.
- [5] Sproat R, et al. A Stochastic Finite-State Word-Segmentation Algorithm for Chinese. Available at URL: <http://xxx.lanl.gov/abs/cmp-lg>
- [6] Meng X F, Zhou Y. Wang S, Domain Knowledge Extracting in a Chinese Natural Language Interface to Databases: Nchiql. *In Proc. of PAKDD'99, Spinger-Verlag, Beijing, April 1999.*
- [7] Meng X F., Wang S., Researches on the Chinese Restricted Natural Language Interface to databases. *In Proc. of the Fifth International Conference for Young Computer Scientists, ICYCS'99, Nanjing, August, 1999.*
- [8] Meng X F, et al. Investigation and Evaluation of Chinese Natural Language Queries. *Technical Report, Renmin University of China, 1998.*

Meng Xiaofeng, Associate professor of School of Information, Renmin University of China. He obtained his M.S. degree from Remin University of China in 1993 and Ph.D. degree from the Institute of Computing Technology, The Chinese Academy of Sciences in 1999. His research interests include database systems, natural language interface, mobile and embedded software, and Web application.

Liu Shuang, Ph.D candidate at Institute of Computing Technology, The Chinese Academy of Sciences. She obtained her M.S. degree from Renmin University of China in 1999. Her research interests include database systems.

Wang Shan, Professor and Dean of School of information, Renmin University of China. She obtained her M.S. degree from Remin University of China in 1982. Her research interests include database systems, datawarehouse & data mining, and information systems.