

A Chinese Interactive Feedback System for An e-Learning Website

JUI-FA CHEN[†], WEI-CHUAN LIN^{*}, CHIH-YU JIAN AND CHING-CHUNG HUNG

*Department of Information Engineering
Tamkang University
Tamsui, 251 Taiwan*

[†]E-mail: alpha@mail.tku.edu.tw

^{}Department of Information Technology
Takming College
Taipei, 114 Taiwan*

E-mail: wayne@takming.edu.tw

Considering the popularity of the Internet, an automatic interactive feedback system for e-learning websites is becoming increasingly desirable. However, there are still some problems for computers to understand the natural language, especially the Chinese language. First, because the Chinese language has no space to segment the lexical entry, its segmentation method is more difficult than that of English. Second, the lack of complete grammar for Chinese language makes parsing more difficult and complicated. Building an automatic Chinese feedback system for special application domains could solve these problems. In this paper, an interactive mechanism is proposed to parse, understand and response to a Chinese sentence. This mechanism utilizes a specific lexical database according to the particular application. In this way, a Chinese interactive feedback e-learning website is built for a special application domain that will choose the proper response in a user friendly, accurate and timely manner.

Keywords: natural language, segmentation method, grammar, interactive feedback, lexical database

1. INTRODUCTION

The easiest way to communicate between users is to talk their natural language. Considering the popularity of the Internet, an automatic interactive feedback system for e-learning websites is becoming increasingly desirable. However, it is still difficult for a computer to understand the meaning of natural language. A computer should be capable of a dialog based on some topics of background knowledge. At present a three-year old child can understand and respond better than a computer. To understand the natural language, a computer must be trained to understand a single sentence. The next step would be train it to analyze longer sentences or paragraphs. In principle, there are at least two things a computer must comprehend from a single sentence [11]:

1. Recognize the meaning of each word in the sentence.
2. Transform the linear structure of a sentence into another structure, which represents the meaning of that sentence.

Received July 1, 2004; revised March 14, 2005; accepted May 10, 2005.
Communicated by Robert Lewis.

The first task is to seek the meaning of each lexicon in a dictionary. However, there can be many meanings of each lexicon, and the computer should have the ability to choose the right one. Even if that is accomplished, it is still difficult for the computer to process the Chinese sentence because there are no spaces used to segment the lexicon. Therefore, a segmentation method is needed before parsing the Chinese sentences.

The second task in understanding a Chinese sentence is to transform the segmented lexicons into the structure, which can be understood by a computer. In general, the transformation procedure can be divided into three parts:

- A. Syntactic analysis procedure: In this procedure, the input lexicon should be transformed into a specific structure that can represent the relationship between lexicons. However, not all the combinations of lexicons of a sentence are legal. The computer must eliminate the illegal combinations to ensure a correct performance.
- B. Semantic analysis procedure: This procedure obtains the meaning of the sentence from the established structure. The obtained meaning is a unit of knowledge representation, which can be mapped to the corresponding object or event in the actual world.
- C. Pragmatics analysis procedure: This procedure determines the real purpose of the sentences and makes appropriate response to users.

The remainder of this paper is laid out as follows. Section 2 discusses the related works on syntax and semantic analysis. Section 3 describes the proposed four subsystems of segmentation, syntactic analysis, semantic analysis, and the response subsystems. In section 4 some examples are provided to show the implementation of the proposed method. Finally, in section 5 we draw our conclusion and mention some future works.

2. REVIEW OF RELATED LITERATURE

2.1 Link Grammar Technology

Most sentences in a natural language have the structure that if each word is connected by arcs, the arcs may not cross. This phenomenon is called planarity in the link grammar system [4]. A link grammar consists of a set of words, which has a linking requirement. The linking requirements of each word are contained in a dictionary. To illustrate the linking requirements, Fig. 1 shows a simple dictionary for the words a, the, cat, mouse, and chased. The linking requirement of each word is represented by the Fig. 1 above the word.

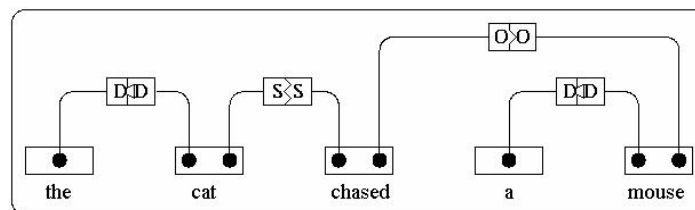


Fig. 1. Words and connectors in the dictionary.

Each of the intricately shaped labeled boxes is a connector. A connector is satisfied by “plugging it into” a compatible connector (as indicated by its shape). If the mating end of a connector is drawn facing to the right, then its mate must be to its right facing to the left. Exactly one of the connectors attached to a given black dot must be satisfied. Thus, cat requires a D connector to its left, and either an O connector to its left or an S connector to its right. Plugging a pair of connectors together corresponds to draw a link between that pair of words.

Fig. 2 is the simplified form of Fig. 1 and it shows that “the cat chased a mouse” is part of this language. Table 1 encodes the linking requirements of the example in Fig. 3.

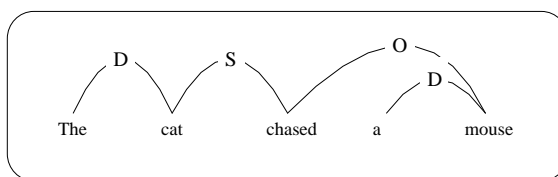


Fig. 2. The simplified form of Fig. 1.

Table 1. The words and linking requirements in a dictionary.

Words	Formula
a the	D+
cat mouse	D- & (O- or S+)
John	O- or S+
ran	S-
Chased	S- & O+

The link grammar dictionary consists of a collection of entries, each of which defines the linking requirements of one or more words. These requirements are specified by means of a formula of connectors combined by the binary associative operators & and or. Precedence is specified by means of parentheses. A connector is simply a character string ending in + or -.

2.2 Memory-based Parsing System

Most methods of semantic analysis give the first place to the verb of a sentence and then determine the correctness on the semantics of lexical entries around the verb. The memory-based parsing [1, 7-9] also begins with the restrictions of a verb to determine the correctness of subject and object. The memory-based parsing system consists of four modules as follows.

- Concept sequence layer: This layer keeps the restrictions of the subject and object of each verb for both the syntax and semantics.
- Syntactic layer: This layer keeps all kinds of parts in speech for comparing the syntactic restrictions.

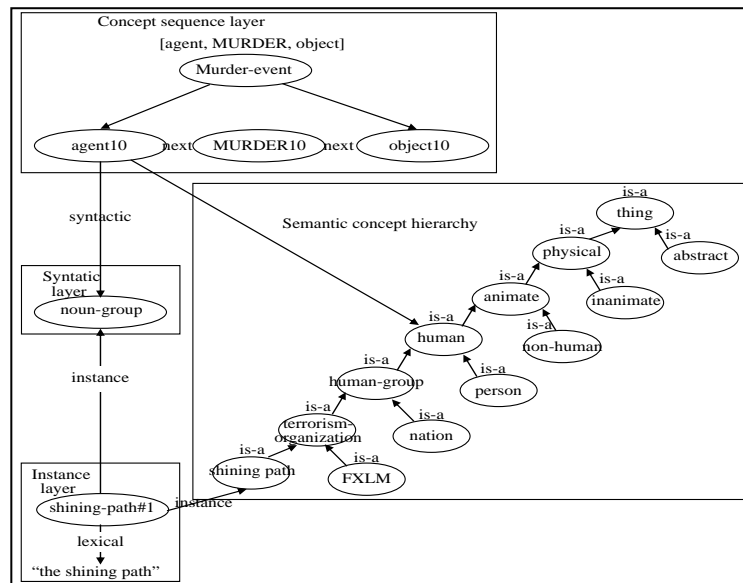


Fig. 3. Part of knowledge base used for processing "the Shining Path".

- Semantic concept hierarchy: This structure defines the relationship of all nouns, and is used for verifying the semantic restrictions.
- Instance layer: This layer contains the lexical entries of a sentence typed by the user.

Fig. 3 shows an example of a memory-based parsing with a concept sequence [agent, MURDER, object] for murder-event. At the top of the knowledge base is the concept sequence layer consisting of concept sequence roots and elements. The semantic concept hierarchy and syntactic layer connect concept sequence elements with concept instances in the instance layer. Concept instances are produced from phrasal inputs and are connected to the corresponding syntactic category and semantic concept nodes. The result of parsing is represented by connecting instances of concept sequence roots and corresponding concepts in the instance layer.

3. THE PROPOSED SYSTEM

The proposed system structure is divided into four sub-systems: segmentation system, syntactic analysis system, semantic analysis system, and response system. As Fig. 4 shows, users can input the Chinese sentences to the system via the interface provided by the system. Then the segmentation system begins the segmenting action of the user's input sentences, and gives the appropriate part of speech for each segmented lexical entry. In the syntactic analysis the system parses these segmented lexical entries to judge whether the sentence is legal, and then gives the syntactic part of each lexical entry. In the semantic analysis the system judges the correctness of the semantics and provides a semantic learning method bases on the user's oral habit. Finally, a response system gives the user the response result according to the encoding of the input sentence.

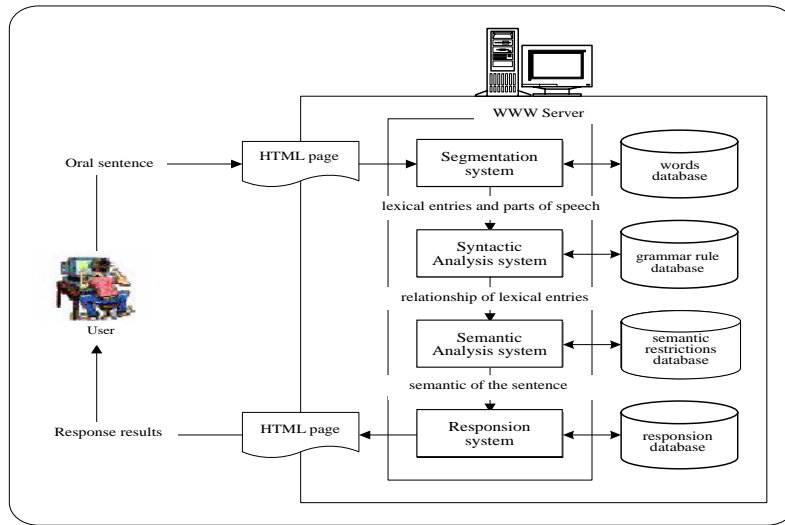


Fig. 4. The proposed system architecture.

3.1 Segmentation System

The difference between the Chinese and the English language is that the Chinese language has no obvious separation to segment the lexical entry. Therefore a segmentation method to parse the Chinese language is necessary. Fig. 5 shows the architecture of the Segmentation System.

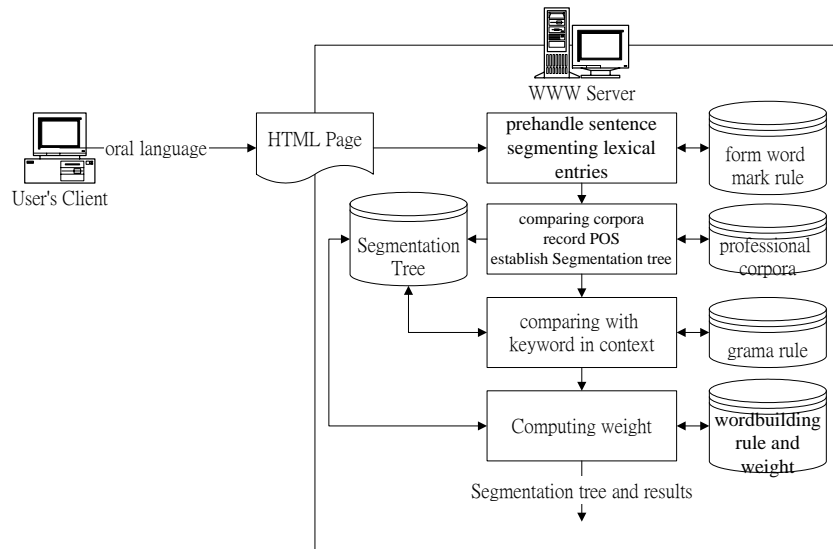


Fig. 5. The architecture of segmentation system.

The segmentation system structure is divided into four sub-systems: segmentation system, corpora-comparing system, keyword in context comparing system, and weighted calculation method [2]. These subsystems are explained as follows:

1. Segmentation: Segmentation separates the user's input sentences, and compares these separated units with the corpora-comparing system.
2. Corpora-comparing system: This system includes two steps, corpora-comparing and part-of-speech (POS) saving. It compares the receiving strings with those in the corpora, and saves the results to build a segmentation tree.
3. Keyword in context comparing system: After building a segmentation tree, the system compares the POS with the keyword in context according to the grammar rules and deletes the improper segmentation tree. This mechanism is divided as Unknown Word Judgment System, and Context-proof-reading System.
4. Weighted calculation system: Because there may be more than one kind of segmentation results, every result's weighted value is computed to find the most proper one. The segmentation result with the largest weighted value is the most suitable result.

3.1.1 Segmentation

There is no space between lexical entries in the Chinese language to help segmentation. Chinese is composed of two continuous bits in representation. The system must judge whether this word is Chinese code when segmenting sentences in order to put the pointer's displacement in the best place. However in transmitting, some special Chinese word will gain a "\ " after being transmitted through the network browser. Thus the system should remove the rest of the "\ " prior to segmenting the sentences. As a result, the segmentation system must offer a mechanism to deal with this problem.

The main functions of the segmentation system are:

1. Dealing with the special Chinese word in the user's input sentences in order to avoid punctuation-transferring mistakes.
2. Transferring the punctuation of user's input sentences so as to be the same dividing code. The system splits the continuous English words and numbers them into strings, then adds a dividing code both in front of and behind the strings. The system also adds a dividing code behind the empty word of the user's input sentence. In addition the system splits the user's input sentence and compares the segmented lexical entries with the corpora-comparing system. Fig. 6 shows the segmentation system process.

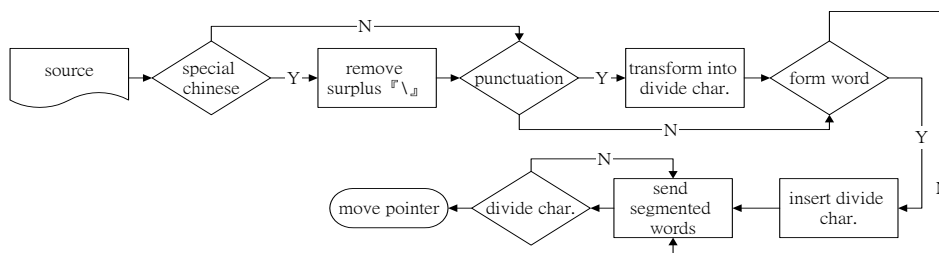


Fig. 6. Segmentation system's flow chart.

Most of the Chinese lexicon possesses at most six characters. The segmentation length of a Chinese sentence should be limited to avoid segmenting a sentence into many impossible ways. For example, a sentence composed of n words should have $2^{(n-1)}$ possible segmentations. A maximum matching [3] mechanism is used to segment a sentence. Basically, the maximum matching method compares a string started at the k th character with a lexical database and finds out all possible segmentations. If $C(k)$, $C(k)_C(k+1)$, $C(k)_C(k+1)_C(k+2)$ are stored in the lexical database, the maximum matching method would choose the longest word and then continue with $C(k+3)$. Because the length of most of the Chinese lexicons do not exceed six characters, the maximum length of a word in a sentence is set to six characters.

3.1.2 Corpora-comparing system

After the segmentation system separates the sentences, it compares string length and context. If there are fitting POS, it records all represented POS and adds the information of the lexical entries to a segment sub-system to determine whether this word is an unknown word. If it is determined to be an unknown word, it will be saved into the segmentation tree. Because most of the new unknown words are proper names, the system often sets the POS of these unknown words to be temporary nouns, and continues processing the following set of strings. If the system still cannot find the corresponding POS in the corpus database of one to six continuous words, it views these six continuous words as an unknown word, and gives it the property of a noun which will be added into the segmentation tree.

The corpora structure used in the system is shown in Table 2. The saved data format is the numbers of words, the Context, POS, types, and the word probability. They are explained as follows:

Table 2. Corpora data structure.

Numbers of words	Context	POS	Types	Word frequency
------------------	---------	-----	-------	----------------

Numbers of words: In order to speed up the comparing of the corpora, the information of the numbers of lexical entries is recorded. In this way, the system does not have to search the entire database, and as a result greatly improves the efficiency of the system.

Context: Refers to the recorded context of the lexical entry.

POS: Recording the POS of the lexical entry. If the number of the POS is larger than one, the system separates the sentence with “,” as a dividing symbol.

Types: As mentioned before, this paper is focused on mutual conversation segmentation in the network domain. Therefore the type is used to mark what kind of database the lexical entry belongs to.

Word frequency: It shows how often which lexical entry has appeared in the equilibrium corpus database. The information is primary used for weight calculation.

3.1.3 Unknown word judgment system

This system searches the segmented lexical entries for the unknown word. It is used to determine if this word is an unknown word so as to avoid mistakes. After the system receiving strings, it splits N continuous words continuously and compares them with the corpus database. If the proofreading is successful it feeds the first words of the lexical entry back to the position where the string engages. The system sets the string which is beyond the position of being an unknown word and saves it into a segmentation tree. If it fails when compared, then the system feeds back 0 to show that this word is not an unknown word. Fig. 7 shows the flow of the unknown word judgment system.

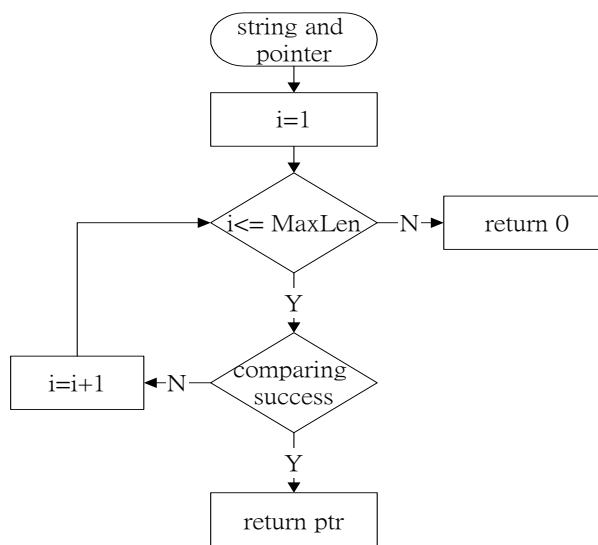


Fig. 7. Unknown word judgment system process.

3.1.4 The data structure of the segmentation tree node

The system adds the segmented lexical entry into the segmentation tree in order to speed up the node searching. The segmentation tree structure can make data saving more flexible whether increasing or decreasing the segmentation nodes. The segmentation tree is a six node tree. Its structure is shown in Fig. 8. Every node follows from none to at most six sub-nodes, which are added dynamically when compared with the corpora. The original input sentence connects the first node of the root to the following branches. In this way, the system can dispose of the space dynamically in order to save the segmentation result, and it also shows clearly every different segmentation result.

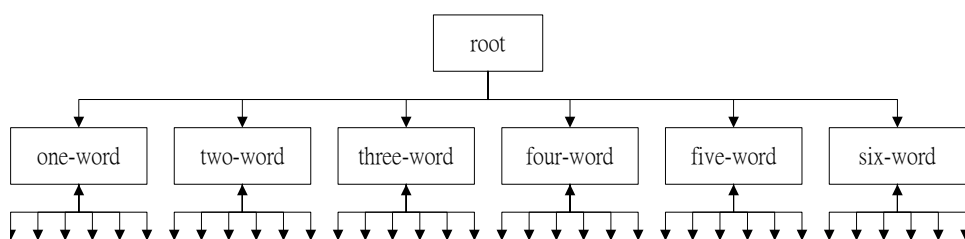


Fig. 8. segmentation tree structure.

Every node of the segmentation tree is composed of the following node structure as shown in Fig. 9. Each node clearly records the information after the system searches the database, which is convenient to use for the context-proofreading system and the weighted-calculating system.

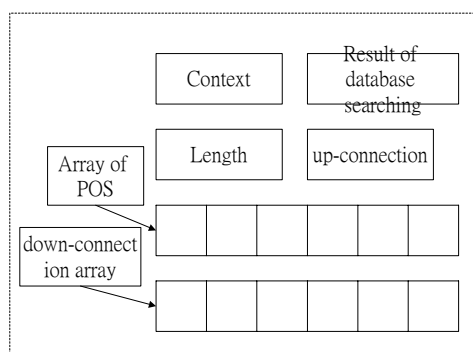


Fig. 9. Node data structure.

The fields in Fig. 9 are explained as follows:

- (A) Context: The context of the lexical entry.
- (B) Length: Records the length of the strings.
- (C) Array of POS: Records the POS of a lexical entry. It records at most six POS and the pre-setting value is an empty string.
- (D) Result of database searching: Records the searching result of this lexical entry. If it is found from the corpus database, it is recorded as True. However, if it is an unknown word, it is recorded as False.
- (E) Down-connection and up-connection: Records the number of the upper or lower nodes, referring to the upward or downward lexical entry.

If there is no pointing direction, it records 0. Because the system deals at most with six continuous words when segmenting sentences, the amount of the down-connecting array is six and that of the up-connecting array is one.

3.1.5 Context-proofreading system

This system uses the segmentation tree built-in in the corpora-comparing system for comparing with the context according to the grammar recorded in the grammar principle database. This system deletes the segmentation sub-tree that does not fit the grammar, and decides the POS to which the lexical entry belongs to. When proceeding with the grammar proofreading, the system only compares the POS of the front lexical entry, rather than proofreading the whole article. This way the system can not only determine the POS of the lexical entry in the oral language conversation more correctly, but it also increases the speed and flexibility of the judgment.

The main reason for adopting the method of judging the relationship of the grammar between the front and rear words only is because the grammar structure is usually not perfect in an oral language conversation. If the system would use it, then this would result in judging mistakes in determining the POS. In contrast, if the system uses the method of the relationship between the front and rear words, it would correctly determine the POS by fitting some grammar principle.

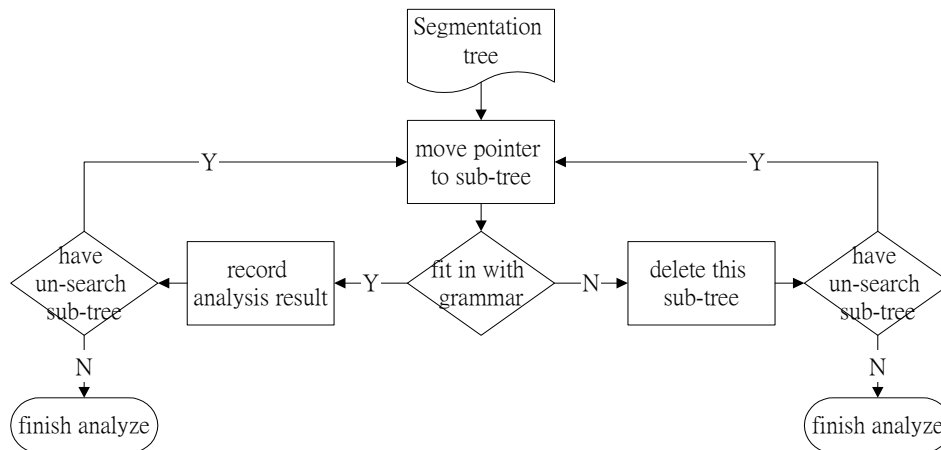


Fig. 10. The keyword in context comparing system's flow chart.

3.1.6 Weighted-calculation system

The weighted-calculation system is used to judge the correct segmentation result when there is more than one result after grammar analyzing, since a segmented Chinese sentence may not always have only one suitable way for being split. To solve this problem, this system computes weights according to the lexical entry-building principle, and the segmentation result having the larger weight is the suitable one. The function of the weighted calculation is listed as follows:

$$\text{Weight} = \text{Weights of length} * \text{Weights of searching result} * \text{Word frequency.}$$

- (A) Weights of length: The longer lexical entry has a higher priority according to the lexical entry-building principle. Therefore, the longer the length the larger the weight.
- (B) Weight of searching result: The weight of searching result changes based on whether this word is an unknown word or not. In principal, the weight of a known word is larger than that of an unknown word. However, according to the principle of long lexical entry privacy, the system sets the weight of an unknown word to be the same as N-continuous words, and so its weight is only slightly larger than a one-continuous known word.
- (C) Word frequency: It shows how often which lexical entry has appeared in the equilibrium corpora. The more often the lexical entry appears, the higher frequency it has.

After calculating the weighted sum of all nodes on every branch, the system can find the segmented result that is most suitable for the lexical entry-building principle.

3.2 Syntactic Analysis System

The main function of the syntactic analysis system is to transform the lexical entries of the input sentence into a structure that can represent the relationship of these lexical entries. However, not all the input sentences are legal in syntax, and the system should provide a fault-tolerance mechanism. With a fault-tolerance mechanism, the system can tolerate common mistakes in general oral conversation thereby raising the level of fluency in the conversation. Fig. 11 shows the flowchart of the syntactic analysis system which utilizes the “Word-based Link Grammar” [4] as the parsing method of the syntax.

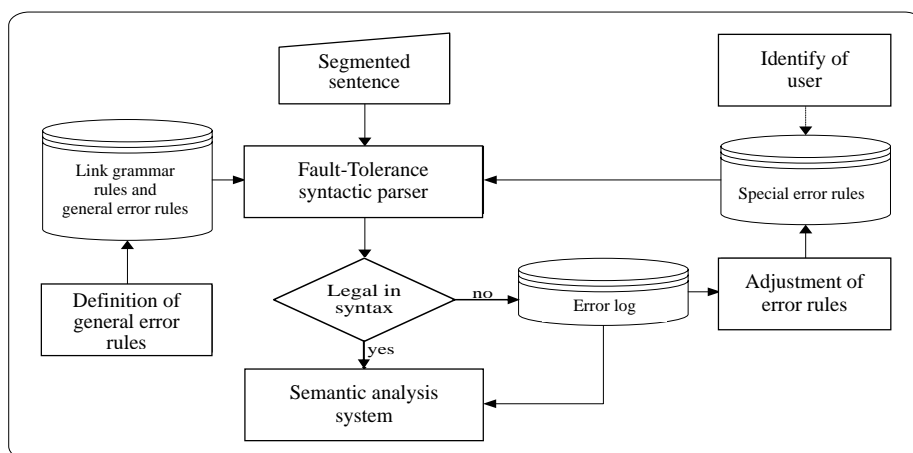


Fig. 11. Flowchart of syntactic analysis system.

3.2.1 Word-based link grammar

The method of the Word-based Link Grammar defines the linking rules on each

lexical entry for making the link relations. The syntactic analysis system obtains the relations as the syntactic parts of each lexical entry. Table 3 shows the linking rules of each part of speech.

Table 3. The linking rules of each part of speech.

Part of speech	Linking rules
Nuon(N)	(S+ or O-) & (Q- or ()) & (@Adj- or ()) & (Do- or ()) & (Ds+ or ()) & (Cn1+ or ()) & (Cn2- or ())
Personal pronoun(Pa)	(S+ or O-) & (@Adj- or ()) & (Ds+ or ()) & (Cn1+ or ()) & (Cn2- or ())
Demonstrative(Pb)	(Bs+ or Pq+)
Doubt pronoun(Pc)	(S+ or O-)
Quantifier (Q)	(num- or Pq- or (Pq- & num-)) & (Q+)
Adjective(Adj)	(Adj+ or Bj-) & (Adva- or ()) & (Noj- or ()) & (Ca1+ or ()) & (Ca2- or ())
Adverb-decorate adjective(Adva)	(Adva+)
Adverb-decorate verb(Advb)	(Advb+)
Negation(No)	(Noj+ or Nov+)
Auxiliary verb(Hv)	(Hv+)
Transitive verb(Vt)	(Hv- or ()) & (S-) & (O+) & (Advb- or ())
Intransitive verb(Vi)	(Hv- or ()) & (S-) & (Advb- or ())
Preposition(D)	(Ds- & Do+)
Conjunction(C)	(Ca1- & Ca2+) or (Cn1- & Cn2+)
Indicative(Bv)	(Bs- or S-) & (O+ or ()) & (Bj+ or ())

When the syntactic analysis system starts analyzing, it obtains linking rules of each lexical entry from a dictionary and makes a link according to these linking rules. The parsing algorithm is shown as:

Algorithm 1. Syntactic Analysis System

<p>Comment: Sentence: sentence inputted by user Token: segmented lexical entry First-Token: first lexical entry of sentence Last-Token: last lexical entry of sentence Token_Link: flag of whether the lexical entry be linked or not Link_Grammar: linking rules of lexical entry Disjuncts: linking rules in disjunctive form Syntactic_Error: syntactic error flag Right_Links: right connectors of linking rules</p>

```

Left_Links: left connectors of linking rules
Syntactic_Part: syntactic part
Syntactic_Error_Procedure: procedure when errors exist on syntax
BEGIN
  get Sentence's Tokens which segmented by segmentation system
END
BEGIN
  FOR (i = First-Token to Last-Token)
  BEGIN
    set Token_Link off
    get Link_Grammar of ith Token from Dictionary
    make Disjuncts of ith Token
  END
  set Syntactic_Error off
  FOR (i = First-Token to Last-Token)
  BEGIN
    FOR (j = next Token of ith Token to Last-Token and exist Right_Links)
    BEGIN
      IF (one of jth Token's Left_Links match one of ith Token's Right_Links) THEN
      BEGIN
        1. make a link between ith and jth Token and assign Syntactic_Part
        2. set both ith and jth Token's Token_Link on
        3. remove the Disjuncts of ith Token and jth Token that without the link
        4. remove this link from the Disjuncts of ith Token
      END
    END
    IF (ith Token's Token_Link = off) THEN
    BEGIN
      set Syntactic_Error on
    END
  END
  IF (Syntactic_Error = on) THEN
  BEGIN
    call Syntactic_Error_Procedure()
  END
END

```

3.2.2 Fault-tolerance mechanism

The sentences that have a syntax error usually appear in the oral conversation and those sentences are difficult to parse. Therefore it is necessary for the syntactic analysis system to provide a fault-tolerance mechanism. The proposed syntactic analysis system provides the fault-tolerance mechanism by means of modifying the linking rules of interrelated lexical entries. Fig. 12 shows an example of fault-tolerance processing by omitting the preposition “的”.

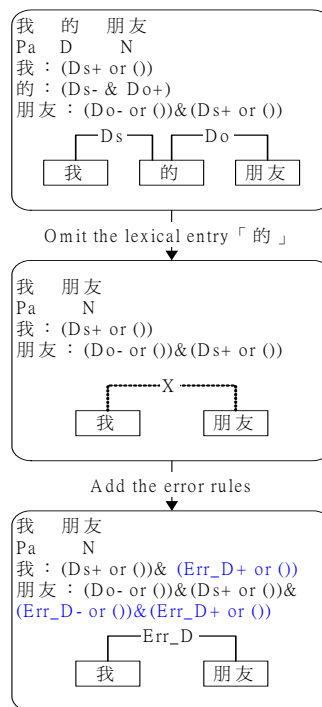


Fig. 12. Fault-tolerance processing with omitting the preposition.

In the first block of Fig. 12, after the segmentation system process, the correct Chinese sentence “我的朋友” is segmented into “我”, “的”, and “朋友” and their parts of speech are “Pa”, “D”, and “N” respectively. The system obtains the linking rules of each lexical entry from the dictionary and checks if the linkage of each lexical entry is correct. However, in the second block of Fig. 6, because of the omission of preposition “的”, the sentence can not make a connection between lexical entry “我” and “朋友” by means of the linking rules. Therefore, in the last block of Fig. 12, with the defining of error linking rules “Err_D”, the lexical entry “我” and “朋友” can make a connection by linking rules “Err_D” so as to provide the fault-tolerance.

3.3 Semantic Analysis System

The semantic analysis system, as shown in Fig. 13, transforms the structure of the sentence, as constructed by the syntactic analysis system, into the semantic meaning. The system judges the correctness of the semantics and provides a semantic learning method based on the user’s oral habit.

Because the judgment of semantics only determines the correctness of the subject and the object around the verb, the proposed system searches the verb of a sentence in advance. If there is no verb in the sentence, then it will continue to the next sub-system after retaining the semantic meaning in the semantic network. The parsing algorithm of semantics is shown as follows.

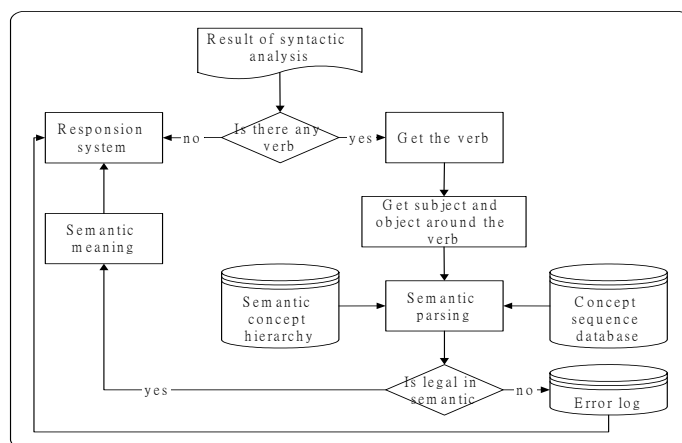


Fig. 13. Flowchart of semantic analysis system.

Algorithm 2. Semantic Analysis System

```

Comment:
Token: lexical entry
First-Token: first lexical entry of sentence
Last-Token: last lexical entry of sentence
Syntactic_Part: syntactic part
Verb: the verb of a sentence
Subjective-Token: the lexical entry of subject
Objective-Token: the lexical entry of object
Verb-Token: the lexical entry of verb
Semantic_Error_Procedure: procedure when errors exist on semantics
Semantic_Network: semantic network
BEGIN
  FOR (i = First-Token to Last-Token)
  BEGIN
    IF (exist Token which Syntactic_Part is a Verb) THEN
    BEGIN
      search Subjective-Token and Objective-Token that related to this Verb-Token
      check semantics between Subjective-Token and Verb-Token
      check semantics between Verb-Token and Objective-Token
      IF(semantics is not illegal) THEN
      BEGIN
        call Semantic_Error_Procedure()
        return
      END
      according to Subjective-Token, Verb-Token and Objective-Token create Semantic_Network
    END
  END
  FOR (i = First-Token to Last-Token)
  BEGIN
  
```

```

IF (Token isn't in Semantic_Network) THEN
BEGIN
  insert the Token into the Semantic_Network according to the link
END
END
END

```

3.3.1 Memory-based parsing system

The proposed system utilizes the “Memory-based parsing system” [9] as the parsing method of the semantics. There are three parts in the memory-based parsing system: concept sequence layer, semantic concept hierarchy, and instance layer.

3.3.1.1 Concept sequence layer

The concept sequence layer keeps both the syntactic and the semantic restrictions of the subject and the object around the verb. As shown in Fig. 14, the concept sequence layer takes the verb as the principal element. The verb element links to both the subject and the object elements via the pointers to obtain their restrictions.

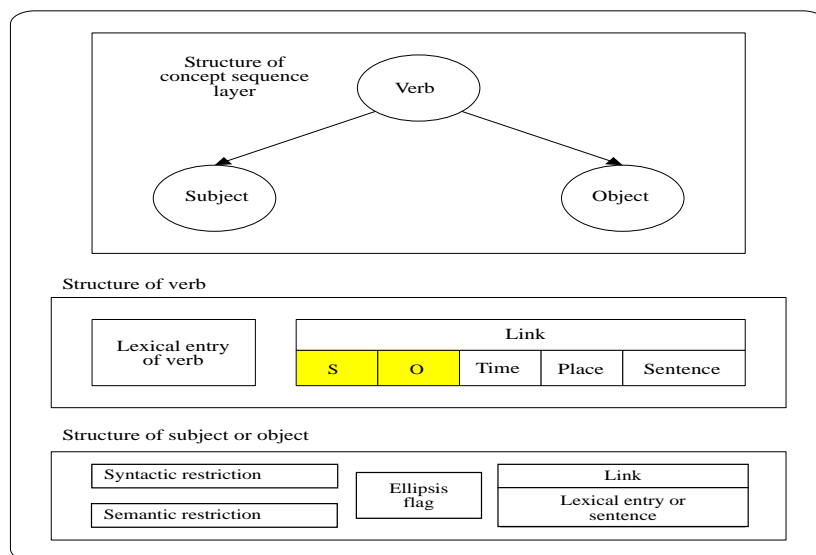


Fig. 14. Structure of concept sequence layer.

- Structure of the verb:
 - ◆ Lexical entry of the verb: Save the context of the verb.
 - ◆ S: Link to the restriction of the subject.
 - ◆ O: Link to the restriction of the object.
 - ◆ Time: Link to the parts of the sentence regarding time.

- ◆ Place: Link to the parts of sentence regarding place.
- ◆ Sentence: Link to the sub-sentence.
- Structure of the subject or the object:
 - ◆ Syntactic restriction: Record the restriction of the syntax.
 - ◆ Semantic restriction: Record the restriction of the semantics.
 - ◆ Ellipsis flag: For judging whether the subject or the object element can be omitted.
 - ◆ Lexical entry or sentence: Link to the actual lexical entry or sub-sentence of the subject or the object.

Based on the structure of the concept sequence layer, the system provides not only verification of semantic restrictions but is also a basis of encoding of the input sentence.

3.3.1.2 Semantic concept hierarchy

The semantic concept hierarchy defines the relation of the nouns according to the meaning of the nouns. The semantic restrictions of subject and object in the concept sequence layer direct them to get their restrictions via the pointers. The structure of the semantic concept hierarchy is shown in Fig. 15.

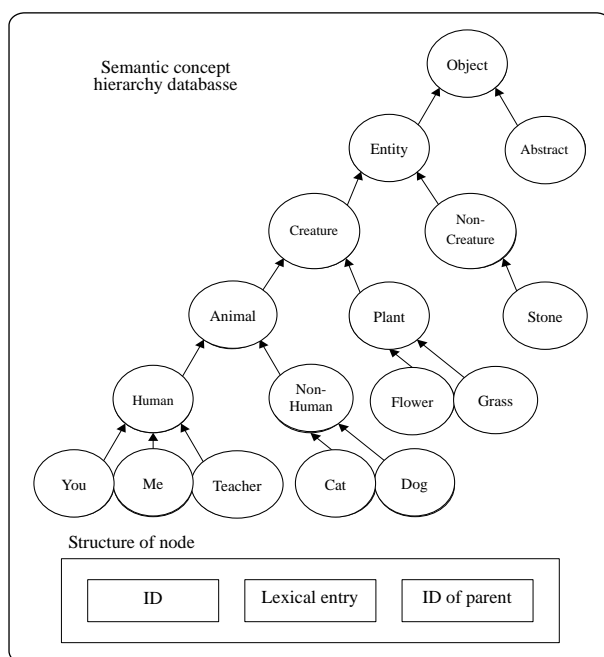


Fig. 15. Structure of semantic concept hierarchy.

- ID: Store the identity number of noun.
- Lexical entry: Store the context of noun.
- ID of parent: Keep up-link of parent's ID.

3.3.1.3 Instance layer

In the instance layer, the proposed system records the lexical entries and obtains the semantic restrictions from the concept sequence layer by comparing the lexical entry of the verb with the verb elements in the concept sequence layer. Fig. 16 takes “我丟一顆石頭” as an example for parsing by the above three layers. Because each lexical entry of subject and object could find a path to reach their semantic restrictions, the example sentence is legal. The parsing path of the subject is “「我」→「人類」→「動物」” and the path of the object is “「石頭」→「非生物」→「實體」”.

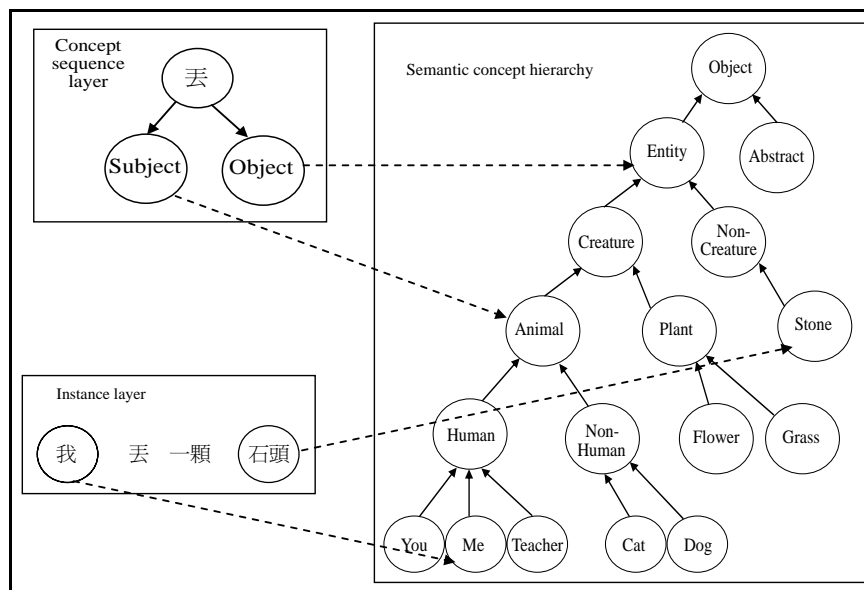


Fig. 16. Example of semantic verification.

3.3.2 Learning mechanism of semantics

In the procedure of semantics processing there could be some inconsistency between the system and the users. Because of this problem, the system should provide a learning mechanism [6] for reducing the differences between the system and the users. The proposed learning mechanism is divided into two parts: generalization and specialization.

3.3.2.1 Generalization

Generalization of the learning mechanism loosens the semantic restrictions. Fig. 17 shows examples of generalization.

There are two conditions of generalization when differences appear between the system and the users.

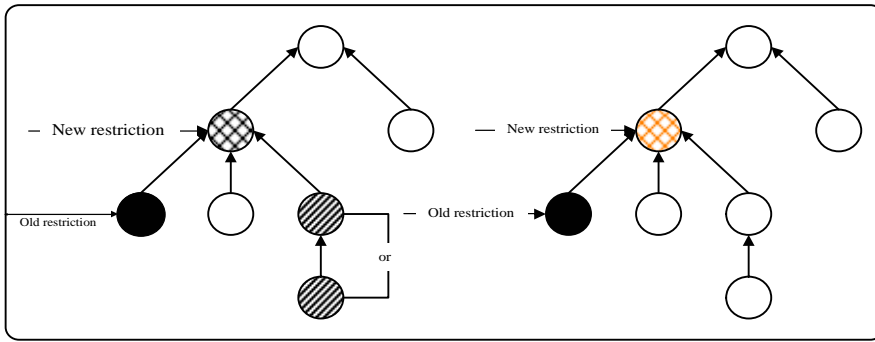


Fig. 17. Generalization.

- Restriction layer of the system \geq restriction layer considered by the users: As shown in the left side of Fig. 17. If users consider that one of the oblique nodes should be corrected then the system will find the lowest common parent node (meshed node) of the oblique node and the restriction node (black node) as the new restriction.
- Restriction layer of the system $<$ restriction layer considered by users: As shown in the right side of Fig. 17. If users consider that the meshed node on top of the restriction node (black node) should be corrected, then this meshed node becomes the new restriction node.

3.3.2.2 Specialization

Specialization of the learning mechanism shrinks the semantic restrictions. Fig. 18 shows an example of specialization.

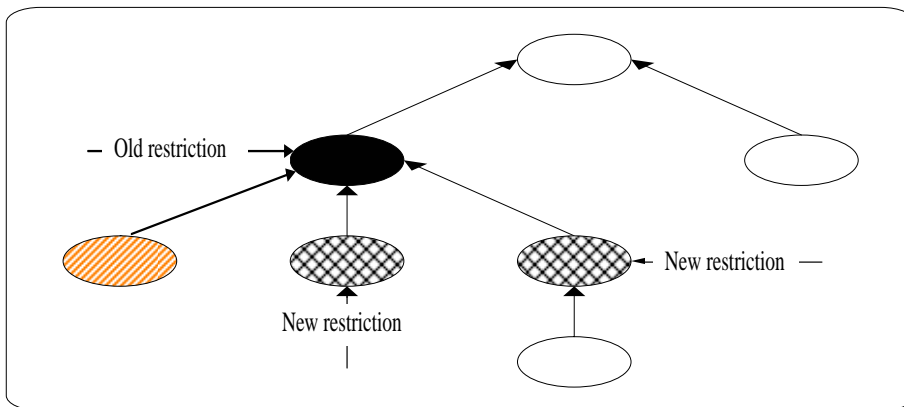


Fig. 18. Specialization.

If a user considers one of the nodes (oblique node) that under the restriction node should be illegal in semantics, then the system will change the restriction by eliminating

all the illegal nodes under the restriction node and the meshed nodes then become the new restriction nodes, see Fig. 18. If the illegal node considered by users is above or the same as the restriction node, then the system will ask users what the restriction should be.

3.3.3 Semantic network

The purpose of the semantic network [10] is to store and represent the meaning of semantics so as to apply it in the inference mechanism. The semantic network describes the relation between the object and the event. There are three elements in a semantic unit which are entity, attribute, and value.

- Entity: The principal part of a semantic unit that represents an object or event.
- Attribute: An arc that describes the attribute of the entity.
- Value: The result of the attribute that describes the entity.

Fig. 19 takes the sentence “我丟一顆石頭” as an example. The meshed node is one of the nodes in the concept hierarchy layer, and the actual lexical entry (石頭) in the instance layer can use the arc of the attribute ‘instance-of’ to form a semantic unit of semantic network with the concept node ‘石頭’. With the link of the attribute ‘instance-of’ the actual lexical entry ‘石頭’ can inherit the property or the capability from the concept node. The system transforms the linkage (S, 我, 丟) into the semantic unit subject (丟, 我). Consequently the unit of semantic network can be established through linkage by the syntactic parser.

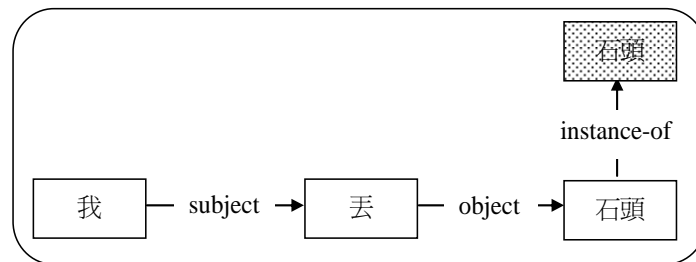


Fig. 19. An example of semantic network.

3.4 Response System

The response system obtains the structure of the semantic meaning and determines the category of the sentences. Finally it encodes the sentence and searches for the result of the response from the database, as shown in Fig. 20.

3.4.1 Category of sentence

In order to raise the accuracy and speed of the response, the system classifies the sentence as follows:

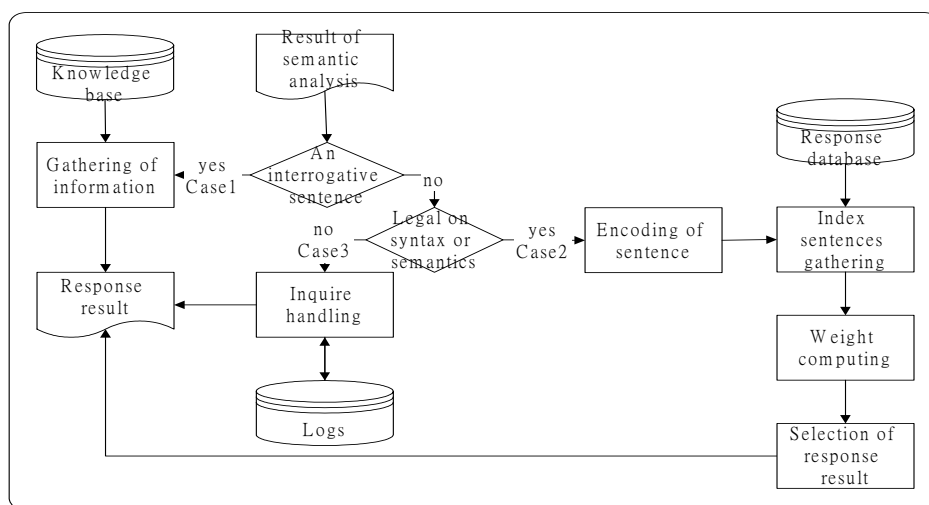


Fig. 20. Flowchart of response system.

- Interrogative sentence: If the response system determines that the inputted sentence is an interrogative sentence according to the interrogatives, it obtains the information concerning the key lexical entries such as nouns from the knowledge base.
- General sentence: If the input sentence is not an interrogative sentence, and is illegal regarding syntax and semantics, then the response system will encode the sentence and search for the response result from the response database.
- Illegal sentence: If the system considers that the input sentence is illegal regarding syntax or semantics, it will record the errors of the sentence and start a syntactic fault-tolerance or semantic learning mechanism.

3.4.2 Encoding mechanism

It is necessary to transform the abstract lexical entries into a specific code format in the response system in order to obtain the response basis. Because the verb is the heart of a sentence, “verb” is set as a high propriety when encoding the sentence. Fig. 21 takes the sentence “我丟一顆石頭” as an example and describes the process.

- Get the construction via the linkage in the syntactic analysis system.
- Get the semantic tree and parse in the semantic analysis system.
- According to the construction of (B) replace each lexical entry with the corresponding syntactic part, part of speech, and lexical number.
- Under the rule of preorder traversal of the binary-tree, the response system edits the contexts of each node into specific code according to the order of traversal. The final result of encoding is “Sen(V-vt002, S-pa001, O-n002, N(1), q001)”.

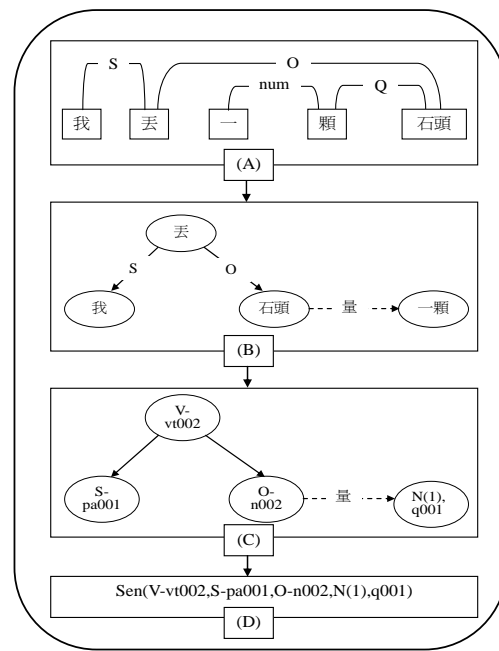


Fig. 21. Diagram of encoding.

3.4.3 Weight-Calculation base on syntactic part

The response system needs a weight-calculation method to judge the best response result when there is more than one result. The proposed system sets weights (\mathbf{W}_w) according to the syntactic part of each lexical entry as follows:

1. Weight of the verb: 3
2. Weight of the subject: 2
3. Weight of the object: 2
4. Weight of the adjective or the adverb: 1
5. Others: 0.5

There is still a standard of evaluation (\mathbf{E}) as:

1. Corresponding lexical entry: 1
2. Without lexical entry or clause: 0
3. Non-corresponding lexical entry: - 1

The function of the weight-calculation (\mathbf{W}_s) is shown as:

$$\mathbf{W}_s = \sum (\mathbf{W}_w * \mathbf{E}).$$

Fig. 22 shows an example of weight-calculation. The first sentence is the response result due to the higher weight.

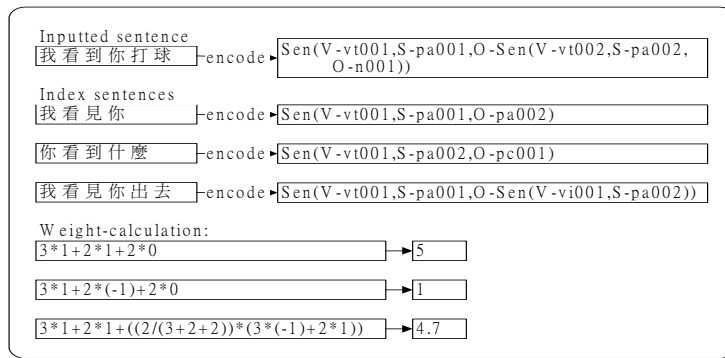


Fig. 22. Example of weight-calculation.

4. EXPERIMENTAL RESULTS

The implementation is to take an example sentence “這個處理器有許多新的功能” to describe the process of each step.

4.1 Segmentation System

The segmentation system divides the example sentence into lexical entries and gives each lexical entry a suitable part of speech as follows. Fig. 23 shows the list of separating from the first layer to the fourth layer.

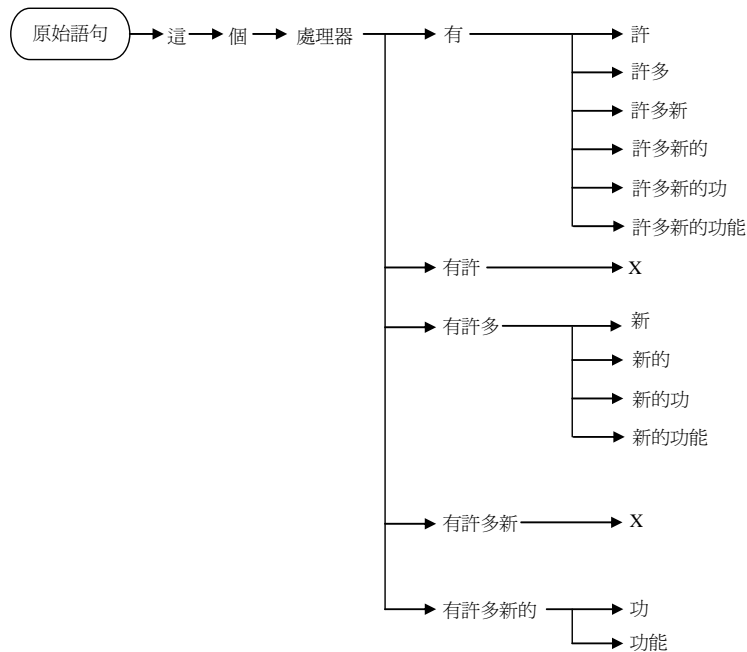


Fig. 23. List of separating from the first layer to the fourth layer.

Table 4. The segmentation table of sentence.

START	STRING	WORD	RESULT
Start at C(1)			
	C(1)	這	O
	C(1)_C(2)	這個	X
	C(1)_C(2)_C(3)	這個處	X
	C(1)_C(2)_C(3)_C(4)	這個處理	X
	C(1)_C(2)_C(3)_C(4)_C(5)	這個處理器	X
	C(1)_C(2)_C(3)_C(4)_C(5)_C(6)	這個處理器有	X
Start at C(2)			
	C(2)	個	O
	C(2)_C(3)	個處	X
	C(2)_C(3)_C(4)	個處理	X
	C(2)_C(3)_C(4)_C(5)	個處理器	X
	C(2)_C(3)_C(4)_C(5)_C(6)	個處理器有	X
	C(2)_C(3)_C(4)_C(5)_C(6)_C(7)	個處理器有許	X
Start at C(3)			
	C(3)	處	X
	C(3)_C(4)	處理	X
	C(3)_C(4)_C(5)	處理器	O
	C(3)_C(4)_C(5)_C(6)	處理器有	X
	C(3)_C(4)_C(5)_C(6)_C(7)	處理器有許	X
	C(3)_C(4)_C(5)_C(6)_C(7)_C(8)	處理器有許多	X
	:	:	
	:	:	

The segmentation system divides the example sentence into lexical entries and gives each lexical entry a suitable part of speech as follows.

這: Demonstrative pronoun [Pb]

個: Quantifier [Q]

處理器: Noun [N]

有: Transitive verb [Vt]

許多: Adjective [Adj]

新的: Adjective [Adj]

功能: Noun [N]

1. 這: (Bs+ or Pq+) →

(()(Bs))

(()(Pq))

2. 個: (num- or Pq- or (Pq- & num-)) & (Q+) →

((num)(Q))

- ((Pq)(Q))
 - ((Pq, num)(Q))
- 3. 處理器: (S+ or O-) & (Q- or ()) & (@Adj- or ()) & (Do- or ()) & (Ds+ or ()) & (Cn1+ or ()) & (Cn2- or ()) → (1)
 - (({Q}, {@Adj}, {Do}, {Cn2})(S, {Ds}, {Cn1}))
 - ((O, {Q}, {@Adj}, {Do}, {Cn2})({Ds}, {Cn1}))
- 4. 有: (Hv- or ()) & (S-) & (O+) & (Advb- or()) →
 - ((S)(O))
 - ((Hv, S)(O))
 - ((S, Advb)(O))
 - ((Hv, S, Advb)(O))
- 5. 許多: (Adj+ or Bj-) & (Adva- or ()) & (Noj- or ()) & (Ca1+ or ()) & (Ca2- or ()) →
 - (({Adva}, {Noj}, {Ca2})(Adj, {Ca1}))
 - ((Bj, {Adva}, {Noj}, {Ca2})({Ca1}))
- 6. 新的: (Adj+ or Bj-) & (Adva- or ()) & (Noj- or ()) & (Ca1+ or ()) & (Ca2- or ()) →
 - (({Adva}, {Noj}, {Ca2})(Adj, {Ca1}))
 - ((Bj, {Adva}, {Noj}, {Ca2})({Ca1}))
- 7. 功能: (S+ or O-) & (Q- or ()) & (@Adj- or ()) & (Do- or ()) & (Ds+ or ()) & (Cn1+ or ()) & (Cn2- or ()) →
 - (({Q}, {@Adj}, {Do}, {Cn2})(S, {Ds}, {Cn1}))
 - ((O, {Q}, {@Adj}, {Do}, {Cn2})({Ds}, {Cn1}))

After obtaining the above linking grammars, the system begins to parse the sentence according to the above algorithm. The linking processes of the first and second lexical entries are shown in Fig. 24.

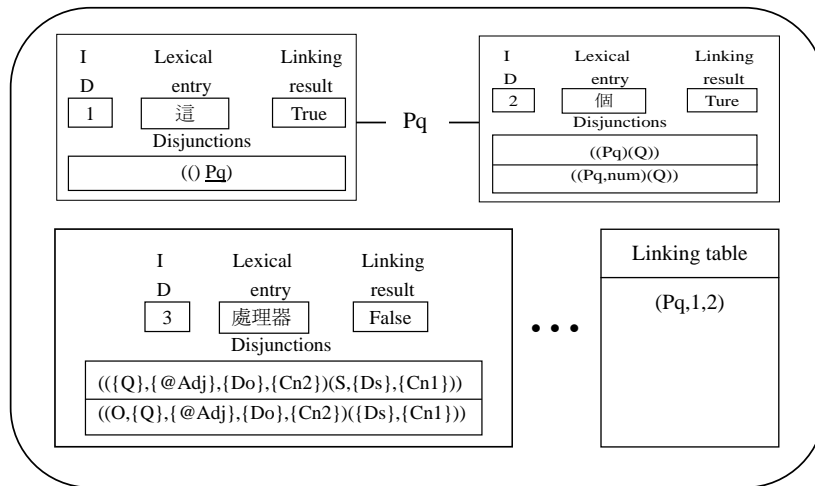


Fig. 24. Process of syntactic analysis.

Because the first lexical entry only has the relation to the second lexical entry with the linking requirement 'Pq', the linking results of the first and second lexical entries are set to true and it records the linkage '(Pq, 1, 2)' in the linking table. The linkage '(Pq, 1, 2)' denotes that the first lexical entry connects leftward to the second lexical entry via the connector 'Pq'. The linking result of the third lexical entry is still false as a result of having no relationship with the first lexical entry. The final result of the syntactic analysis is shown in Fig. 25.

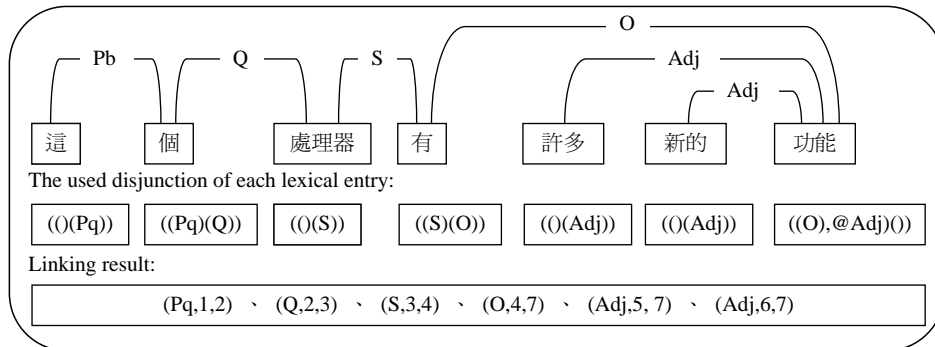


Fig. 25. Result of syntactic analysis.

4.3 Semantic Analysis System

After the syntactic analysis procedure, the system can determine that the fourth lexical entry is a verb, and that the third and seventh lexical entries are subject and object, respectively, according to the linkages (S, 3, 4) and (O, 4, 7). The system determines the correctness of the semantics, and the sentence is legal in semantics by conforming to the semantic restrictions. Finally, it transforms these linkages into semantic meanings as shown in Fig. 26.

- (S, 3, 4) → subject (有, 處理器)
- (O, 4, 7) → object (有, 功能)
- (Adj, 5, 7) → characteristic (功能, 許多)
- (Adj, 6, 7) → characteristic (功能, 新的)

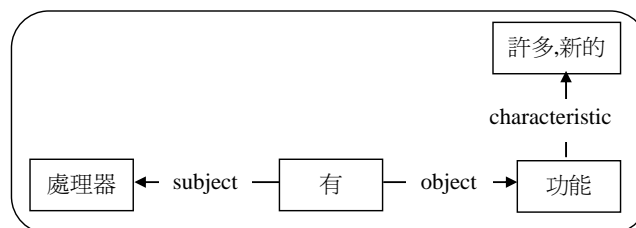


Fig. 26. Semantic network of example sentence.

4.4 Response System

The response system encodes the example sentence according to the syntactic parts, part of speech, and lexical numbers of each lexical entry. The result of encoding is ‘Sen(V-vt001, S-n001, pb001, q001, O-n002, adj001, adj002)’. With the result of the encoding the system searches the response results for those index sentences that resemble the example sentence from the response database. Finally, as shown below in Fig. 27, it chooses the best response result for the user according to these computed weights.

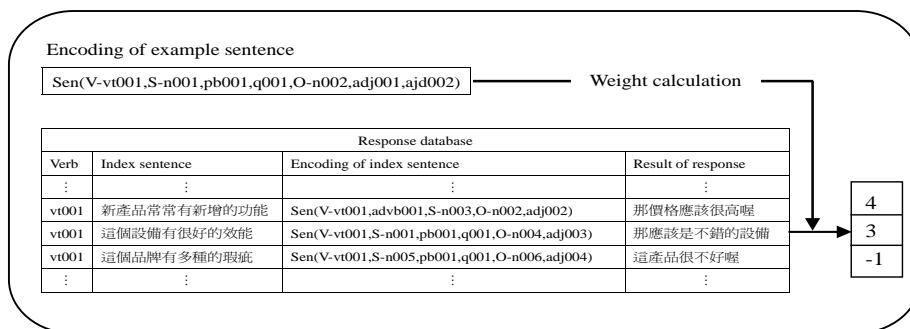


Fig. 27. Contexts of response database.

5. CONCLUSIONS AND FEATURE WORKS

This paper applied the linking grammar to describe the syntactic construction of the sentence and proceeded to verify and record the semantics according to the construction. In the end it replied to the user by finding the response result from the response database. According to the implementation results, the proposed method can quickly and correctly describe the relation between the lexicons. Furthermore, with the use of memory-based parsing, the proposed method can check the correctness of the semantics, and provide a learning mechanism by changing the semantic restrictions of the concept sequence layer. Finally, with the classification of the response databases and by taking the verb of a sentence as the search key, the response system greatly reduces the amount of response results and promotes the speed and accuracy of the response.

To make the interaction of the computer with the users closer, some of the characteristics of the users should be defined in the future. In this way, the proposed method can come up with better and different responses based on these characteristics.

REFERENCES

1. M. Chung and D. Moldovan, "Parallel memory-based parsing on SNAP," in *Proceedings of Seventh International Parallel Processing Symposium*, 1993, pp. 680-684.
2. J. F. Chen, W. C. Lin, and C. Y. Jian, "Using the keyword in context segmentation

- method for collaborative design in a Chinese website,” *10th ISPE International Conference on Concurrent Engineering: Research and Applications*, 2003, pp. 967-975.
3. K. J. Chen and S. H. Liu, “Word identification for mandarin Chinese sentences,” in *Proceedings of 15th International Conference on Computational Linguistics*, 1992, pp. 101-107.
 4. D. Sleator and D. Temperley, “Parsing English with a link grammar,” Carnegie Mellon University Computer Science, Technical Report CMU-CS-91-196, 1991.
 5. T. Y. Ho, “A segmentation method based on keyword in context for the lexical entry of the Chinese language,” Master Degree Thesis, Department of Information Engineering of Tamkang University, 2001.
 6. J. T. Kim and D. I. Moldovan, “Acquisition of linguistic patterns for knowledge-based information extraction,” *IEEE Transactions on Knowledge and Data Engineering*, Vol. 7, 1995, pp. 713-724.
 7. J. T. Kim and D. I. Moldovan, “Acquisition of semantic patterns for information extraction from corpora,” in *Proceedings of 9th Conference on Artificial Intelligence for Applications*, 1993, pp. 171-176.
 8. M. H. Chung and D. Moldovan, “Applying parallel processing to natural-language processing,” *IEEE Expert* [see also *IEEE Intelligent Systems*], Vol. 9, 1994, pp. 36-44.
 9. M. H. Chung and D. Moldovan, “Memory-based parsing with parallel marker-passing,” in *Proceedings of 10th Conference on Artificial Intelligence for Applications*, 1994, pp. 202-207.
 10. M. R. Quillian, “Semantic memory,” *Semantic Information Processing*, MIT Press, Cambridge, MA, 1968, pp. 216-270.
 11. 陳正佳, “一套中文語法分析系統的研究與設計,” Master Degree Thesis, Department of Information Engineering, National Taiwan University, 1985.



Jui-Fa Chen (陳瑞發) received his Ph.D., M.S., and B.S. degrees in the Department of Computer Science and Information Engineering from Tamkang University (TKU), Tamsui, Taipei, Taiwan, in 1998, 1992, and 1990, respectively. He is an Assistant Professor in the Department of Information Technology in Tamkang University (TKU). His research interests include intelligent avatar, peer-to-peer communication, and network application.



Wei-Chuan Lin (林偉川) received his Ph.D., M.S., and B.S. degrees in the Department of Computer Science and Information Engineering from Tamkang University (TKU), Tamsui, Taipei, Taiwan, in 1998, 1986, and 1984, respectively. After graduated from TKU, he worked in the Institute of Information Industry until 1993. He is an Associate Professor in the Department of Information Technology in Takming College, Neihu, Taipei, Taiwan, since 1993. His research interests include intelligent avatar, peer-to-peer communication, and software engineering.



Chih-Yu Jian (簡志宇) received his M.S. and B.S. degrees in the Department of Computer Science and Information Engineering from Tamkang University (TKU), Tamsui, Taipei, Taiwan, in 2001, and 1999. He is studying for Ph.D. degree in Tamkang University (TKU) now. His research interests include intelligent avatar, peer-to-peer communication, and network application.



Ching-Chung Hung (洪慶全) received his M.S. and B.S. degrees in the Department of Computer Science and Information Engineering from Tamkang University (TKU), Tamsui, Taipei, Taiwan, in 2002, 2000. After graduated from TKU, he worked in Internet Information Corporation until now.