IGI PUBLISHING **ITJ 4443**

701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.igi-global.com

This paper appears in the publication, International Journal of Distance Education Technologies , Volume 6, Issue 4
edited by **Shi-Kuo Chang &Timothy K. Shih © 2008, IGI Global**

A Chinese Interactive Feedback System for a Virtual Campus

Jui-Fa Chen, Tamkang University, Taiwan

Wei-Chuan Lin, Tak Ming College, Taiwan

Chih-Yu Jian, Tamkang University, Taiwan

Ching-Chung Hung, Tamkang University, Taiwan

ABSTRACT

Considering the popularity of the Internet, an automatic interactive feedback system for E-learning websites is becoming increasingly desirable. However, computers still have problems understanding natural languages, especially the Chinese language, firstly because the Chinese language has no space to segment lexical entries (its segmentation method is more difficult than that of English) and secondly because of the lack of a complete grammar in the Chinese language, making parsing more difficult and complicated. Building an automated Chinese feedback system for special application domains could solve these problems. This paper proposes an interactive feedback mechanism in a virtual campus that can parse, understand and respond to Chinese sentences. This mechanism utilizes a specific lexical database according to the particular application. In this way, a virtual campus website can implement a special application domain that chooses the proper response in a user friendly, accurate and timely manner.

Keywords: *grammar; interactive feedback; lexical database; natural language; segmentation method*

INTRODUCTION

The easiest way to communicate to users is to talk to them in their natural language. Considering the popularity of the Internet, an automated interactive feedback system for e-learning Web sites is becoming increasingly desirable. However, it still is difficult for a computer to understand the meaning of some natural languages. At present a three-year old child can understand and respond to languages better than a computer can. To understand the natural language, a computer must be trained to understand a single sentence. Then, it would need to be trained to analyze longer sentences or paragraphs. In principle, there are at least

two skills that a computer should be able to apply to a single sentence**:**

1. Defining the meaning of each word in the sentence.
2. Transforming the linear structure of a sentence into another structure that represents the meaning of that sentence.

The first step of processing a Chinese sentence is seeking the meaning of each lexicon in a dictionary. However, there can be many meanings for each lexicon, and the computer must 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 step of understanding a Chinese sentence is transforming the segmented lexicons into a structure that 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 is transformed into a specific structure that represents 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 gives the appropriate response to users.

The remainder of this paper is laid out as followed. The next section discusses the related works on syntax and semantic analysis, followed by a description of the proposed four subsystems of segmentation, syntactic analysis, semantic analysis, and the response subsystems. The next section provides some examples to show the implementation of the proposed method. Finally, there is conclusion and some future works.

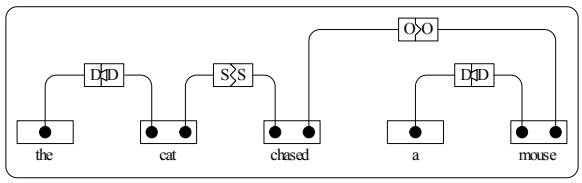
REVIEW OF RELATED WORKS

Link Grammar Technology

Most sentences in a natural language are structured so that arcs that connect words may not cross each other. This phenomenon is called planarity in the link grammar system (Sleator & Temperley, 1991). A link grammar consists of a set of words and has a linking requirement. The linking requirements of each word are contained in a dictionary. To illustrate the linking requirements, Figure 1 shows a simple dictionary for the words "a," "the," "cat," "mouse," and "chased." The linking requirement of each word is represented by the Figure 1 above the word.

Each of the lettered boxes is a connector which is satisfied when it is "plugged 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, the "cat" requires a D connector to its left

Figure 1. Words and connectors in the dictionary



and either an O connector to its left or an S connector to its right. Plugging a pair of connectors together corresponds to drawing a link between that pair of words.

Figure 2 is the simplified form of Figure 1 and shows that "the cat chased a mouse" is part of the language. Table 1 encodes the linking requirements of the example in Figure 2.

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 a formula of connectors combined by the binary associative operators & and or. Precedence is specified by parentheses. A connector is simply a character string ending in + or -.

Memory-Based Parsing System

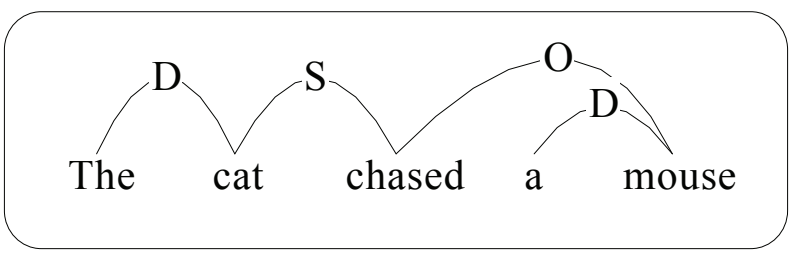
Most methods of semantic analysis first recognize the verb of a sentence and then determine the correctness on the semantics of lexical entries around the verb. Memory-based parsing (Chung, & Moldovan, 1993, 1994a, 1994b; Kim & Moldovan, 1993) also begins with the restrictions of a verb to determine the correctness of subject and object. The memory-based parsing system consists of four modules:

- Concept sequence layer: Keeps the restrictions of the subject and object of each verb for both syntax and semantics.
- Syntactic layer: Keeps all parts of speech for comparing the syntactic restrictions.
- Semantic concept hierarchy: Defines the relationship of all nouns, and is

Table 1. The words and linking requirements in a dictionary

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

Figure 2. The simplified form of Figure 1



used for verifying the semantic restrictions.

- Instance layer: Contains the lexical entries of a sentence typed by the user.

Figure 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, which consists 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.

THE PROPOSED SYSTEM

Overview

There are many learners in a virtual campus, and each learner has his or her own preference. Although the search goal can be found by a belief network, as considered in customization, using only the default category to analyze is insufficient. When a learner logs onto a virtual campus, if he or she is an existing learner, the system could load his learning profile to achieve the customization. If the learner is new, the system could administer a quiz to determine an initial learning profile. The flowchart is shown in Figure 4

The proposed Chinese interactive feedback system (Chen, Lin, & Jian, 2003a, 2003b, 2003c; Chen, Lin, Jian, & Hung, 2005) is divided into four sub-systems: the segmentation system, syntactic analysis system, semantic analysis system,

Figure 3. Part of knowledge base used for processing: "The Shining Path"

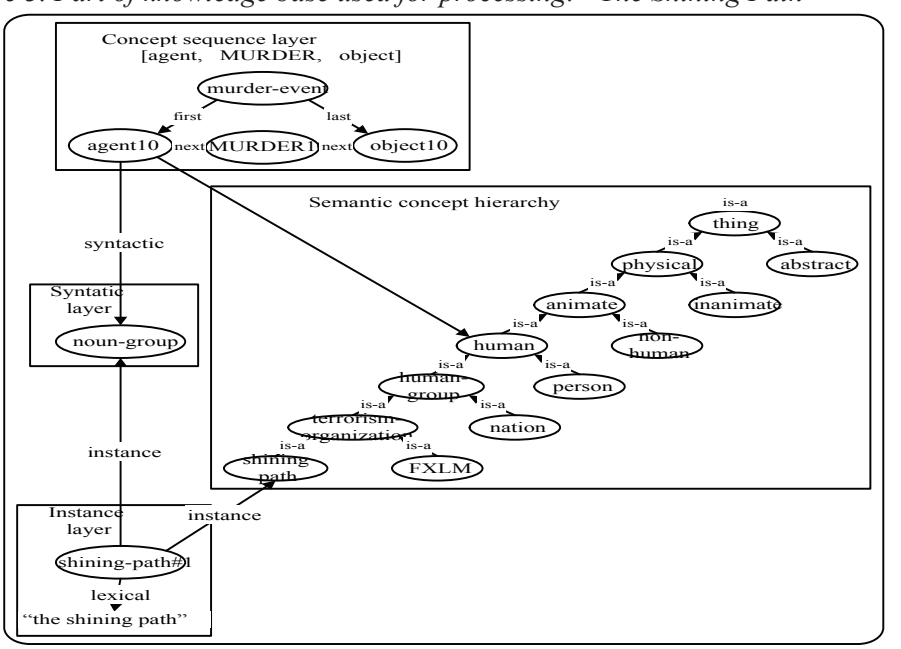
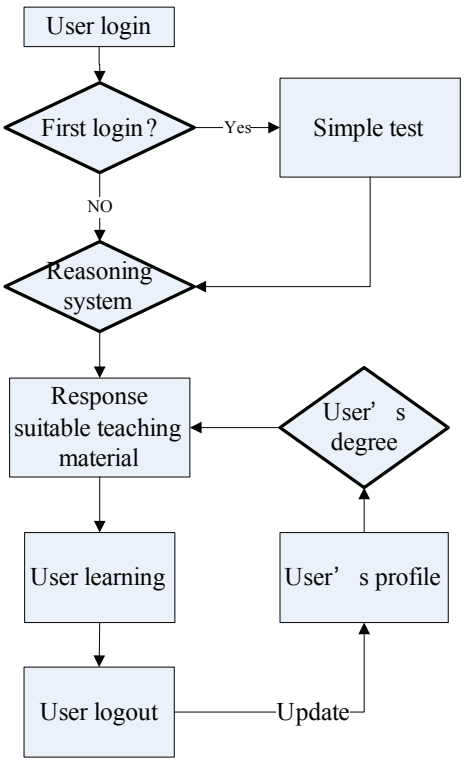


Figure 4. Flowchart of feedback system



and response system. Thus, learners can use Chinese sentences to interact with the virtual campus. When the learners input Chinese sentences, the segmentation system separates the learner's input sentences and gives the appropriate part of speech for each segmented lexical entry. The syntactic analysis system parses these segmented lexical entries to judge whether the sentence is legal and gives the syntactic part of each lexical entry. The semantic analysis system judges the correctness of the semantics and provides a semantic learning method based on the learner's habits. Finally, a response system gives the learner the response result according to the encoding of the input sentence.

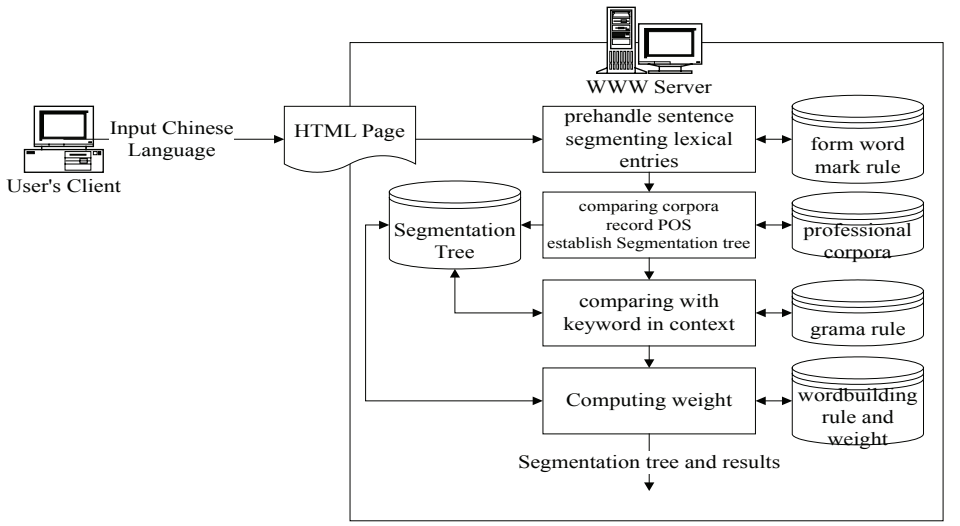
Segmentation System

One difference between the Chinese and 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. Figure 5 shows the architecture of the Segmentation System.

The segmentation system structure is divided into four sub-systems: the segmentation system, corpora-comparing system, keyword in context comparing system, and weighted calculation method (Chen, Lin, & Jian, 2003a). These subsystems are explained as follows:

1. Segmentation: Segmentation separates the user's input sentences and compares the separated units with those obtained from 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 into the Unknown Word Judgment System and Context-proofreading System.
4. Weighted calculation system: Because there may be more than one kind of segmentation result, each 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.

Figure 5. The architecture of the segmentation system



Segmentation

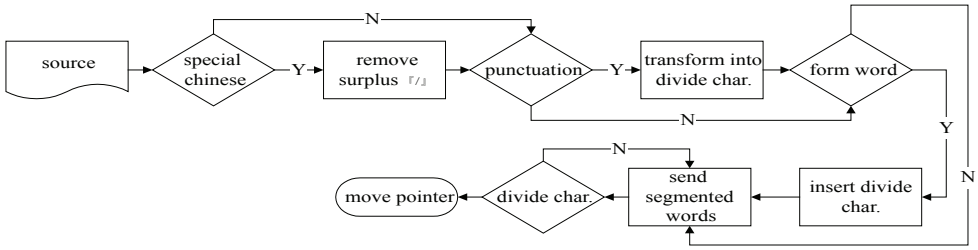
There is no space between lexical entries in the Chinese language to help segmentation. Chinese characters are composed of two continuous bytes in representation. The system judges whether this word is Chinese code when segmenting sentences to put the pointer's displacement in the best place. However, in transmitting, some special Chinese words have a special 『\』 inserted after transmitting through the network browser. The system would remove the 『\』 prior to segmenting the sentences.

The main functions of the segmentation system are:

1. The consideration of special Chinese words in the user's input sentences serves to avoid punctuation-transferring mistakes
2. The transfer of punctuation in the user's input sentences serves to obtain the same dividing code. The system splits the continuous Chinese words and numbers them into strings and 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 also splits the user's input sentence and compares the segmented lexical entries with the corpora-comparing system. Figure 6 shows the segmentation system process.

Most Chinese lexicons possess 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 (Chen & Liu, 1992) 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

Figure 6. Segmentation system's flow chart



method would choose the longest word and 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.

Corpora-Comparing System

After the segmentation system separates the sentences, it compares string length and context. The system records all matching POSs and adds the information of the lexical entries to a segmentation tree. If no POSs match, the system feeds back a false value and recalls the unknown word judgment sub-system to determine whether this word is an unknown word. If the lexical entry is determined by the system to be an unknown word, then the entry is saved into the segmentation tree. Because most new unknown words are proper names, the system often sets the POSs 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 or cannot find the proper unknown word after processing by the unknown word judgment system, it views these six continuous words as an unknown word and adds the unknown word into the segmentation tree.

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

A. **Numbers of words:** To speed up the comparing of the corpora, the information of the numbers of lexical entries are recorded so that the system does not have to search the entire database, greatly improving the efficiency of the system.
B. **Context:** Refers to the recorded context of the lexical entry.
C. **POS:** Records the POS of the lexical entry. If the number of the POS is larger than one, the system separates the sentence with "," as a divided symbol.
D. **Types:** This paper is focused on mutual conversation segmentations in the basic

Table 2. Corpora data structure

Numbers of words	Context	POS	Types	Word frequency

computer concept domain. Therefore, the type is used to mark the kind of special domain database that is used for the lexical entry.

E. **Word frequency:** Shows how often the lexical entry has appeared in the equilibrium corpus database. The information is used primarily for weight calculation.

Unknown Word Judgment System

This system searches the segmented lexical entries for unknown words. After the system receives strings, it splits N continuous words continuously and compares them with the corpus database. If the proofreading is successful, the system 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 the system fails when compared, it feeds back 0 to show that this word is not an unknown word. Figure 7 shows the flow of the unknown word judgment system.

The Data Structure of the Segmentation Tree Node

The system adds the segmented lexical entry into the segmentation tree to speed up node searching. The segmentation tree structure can make data saving more flexible by increasing or decreasing segmentation nodes. The segmentation tree is a six node tree. The tree structure is shown in Figure 8. Every node in Figure 8 follows from zero 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 space dynamically to save and display the segmentation results.

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

The fields in Figure 9 are explained as follows:

Figure 7. Unknown word judgment system process

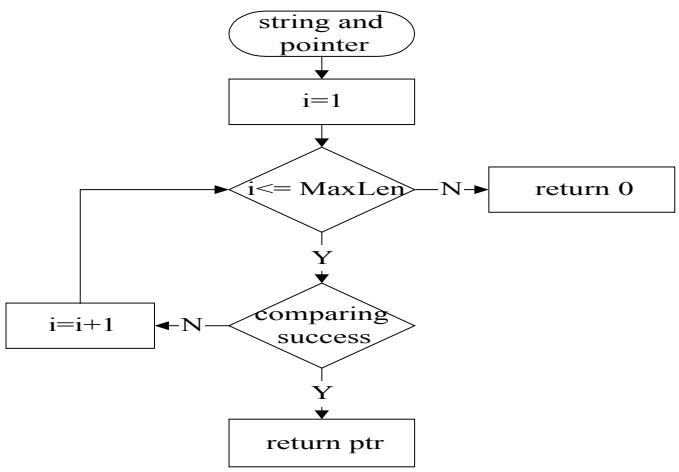
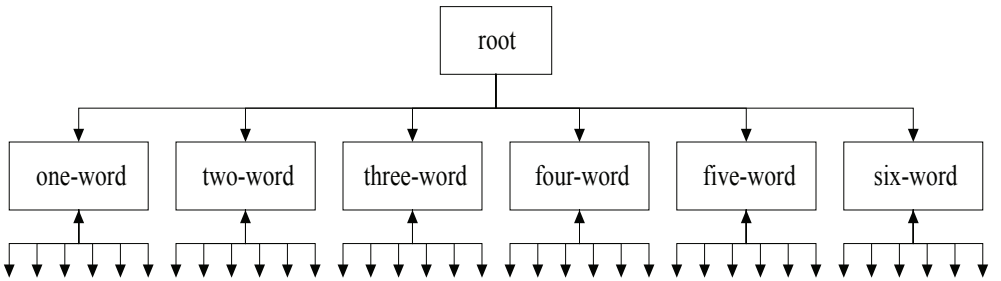
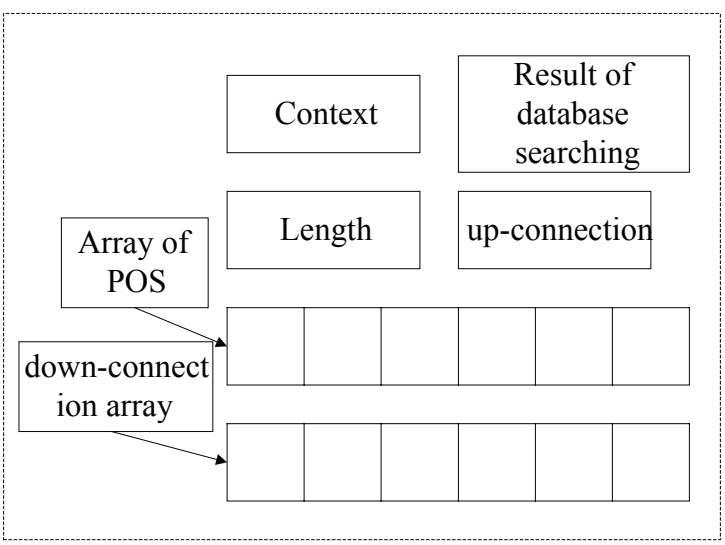


Figure 8. Segmentation tree structure



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 (up to six) with a pre-set value of an empty string.
D. **Result of database searching:** Records the searching result of this lexical entry. If the searching result 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 up or down connection, it records 0. Because the system deals at most with six continuous words when segmenting sentences, the array size of the down-connecting is six and that of the up-connecting array is one.

Figure 9. Node data structure



Context-Proofreading System

This system uses the segmentation tree built in the corpora-comparing system to evaluate the context according to the grammar recorded in the grammar principle database. This system deletes segmentation sub-trees which are not matched with the grammar and decides the POS 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 so that the system can determine the POS of the lexical entry in the oral language conversation more correctly while increasing 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 is that grammar structure is usually not perfect in an oral language conversation. If the system uses only grammar rules it would find errors in determining the POS. In contrast, if the system checks only the relationship between the front and rear words, it would correctly determine the POS. The detailed procedure is shown in Figure 10.

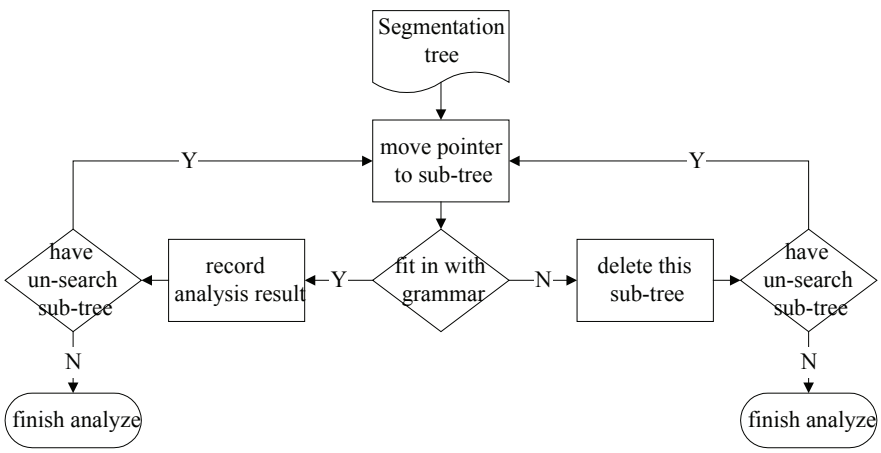
Weighted-Calculation System

The weighted-calculation system is used to judge the correctness of segmentation results when there is more than one result after grammar analyzation, as a segmented Chinese sentence may have more than one suitable way for splitting. This system computes the weights according to the lexical entry building principle, with the segmentation result having the largest weight being the correct one. The process of the weighted calculation is listed as follows:

Weight = Weights of length * Weights of searching result * 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 the searching result changes based on whether this word is an unknown

Figure 10. The flow chart of the keyword in context comparing system



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. Therefore, the weight of an unknown word is only slightly larger than a one-continuous known word.

C. **Word frequency:** Shows how often the lexical entry appears 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 the most suitable for the lexical entry-building principle.

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 and thereby increase the level of fluency in the conversation. Figure 11 shows the flowchart of the syntactic analysis system which utilizes the "Word-based Link Grammar" (Sleator & Temperley, 1991) as the parsing method of the syntax.

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.

When the syntactic analysis system starts analyzing, it obtains the 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.

Figure 11. Flowchart of the syntactic analysis system

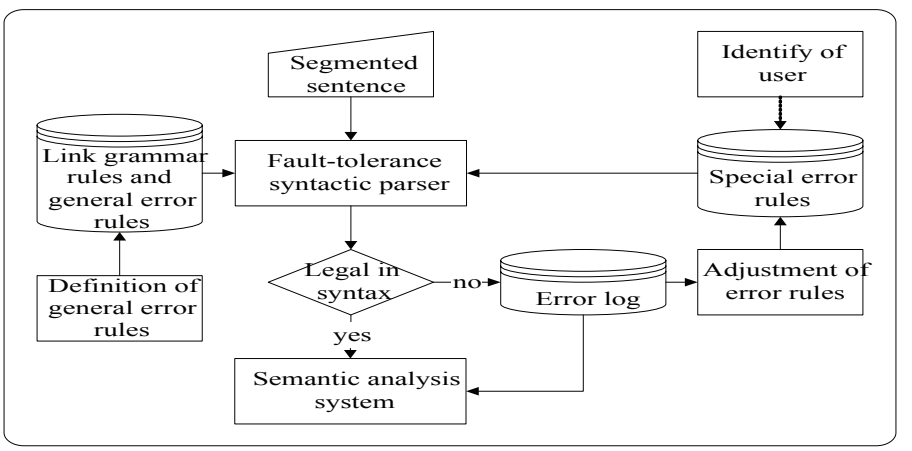


Table 3. The linking rules of each part of speech

part of speech	linking rules
Noun(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 ())

Fault-Tolerance Mechanism

The sentences that have a syntax error usually appear in oral conversations and those sentences that 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 modifying the linking rules of interrelated lexical entries. Figure 12 shows an example of fault-tolerance processing by omitting the preposition "的".

In the first block of Figure 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 Figure 12, 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 Figure 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 processing.

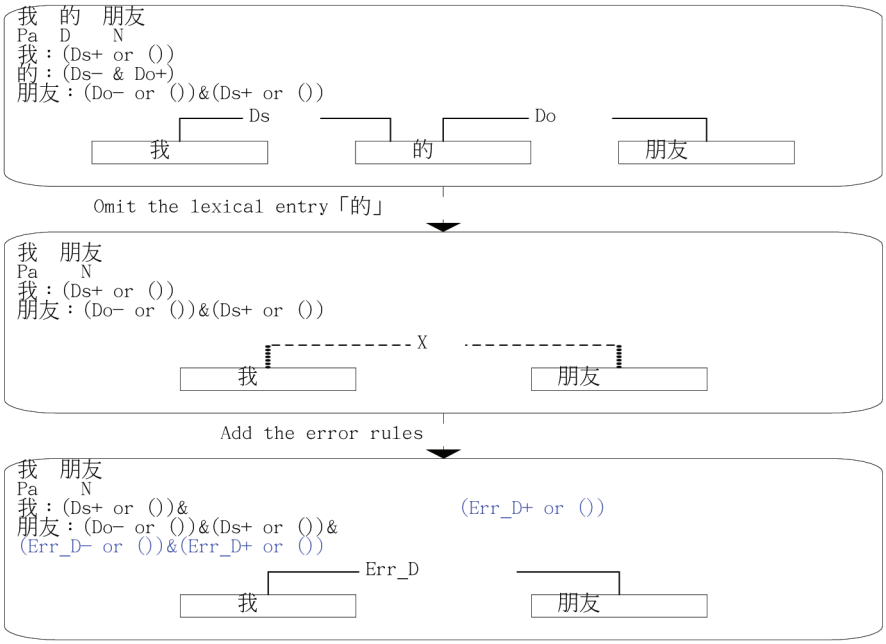
Semantic Analysis System

The semantic analysis system, as shown in Figure 13, transforms the structure of the sentence, as constructed by the syntactic analysis system, into the semantic meaning.

Algorithm 1. Syntactic analysis system

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

Figure 12. Fault-tolerance processing with omitting of the preposition



The system judges the correctness of the semantics and provides a semantic learning method based on the user's oral habits.

Because the judgment of semantics only determines the correctness of the subject and the object around the verb, the proposed system searches for the verb of a sentence in advance. If there is no verb in the sentence, the system will continue to the next sub-system after retaining the semantic meaning in the semantic network. The parsing algorithm of semantics is shown as Algorithm 2.

Memory-Based Parsing System

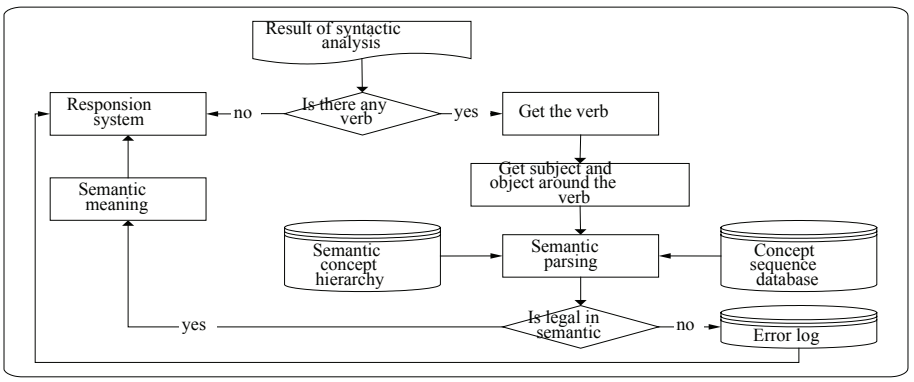
The proposed system utilizes the "Memory-based parsing system" (Chung & Moldovan, 1994b) as the parsing method of the semantics. There are three parts in the memory-based parsing system: the concept sequence layer, the semantic concept hierarchy, and the instance layer.

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 Figure 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. The detailed contents are explained as follows:

- 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.

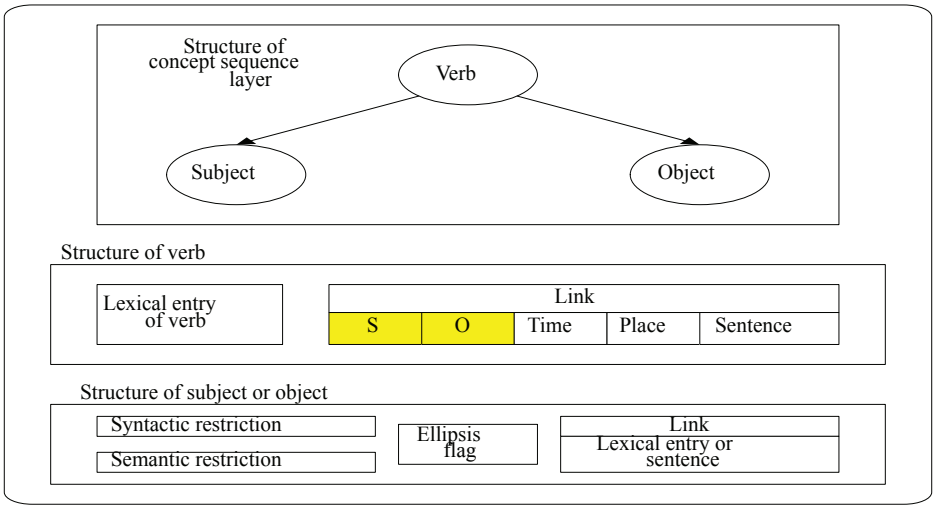
Figure 13. Flowchart of the semantic analysis system



Algorithm 2. Semantic analysis system

```
Comment:
Token: lexical entry
First_Token: first lexical entry of asentence
Last_Token: last lexical entry of a sentence
Syntactic_Part: syntactic part
Verb: the verb of a sentence
Subjective_Token: the lexical entry of a subject
Objective_Token: the lexical entry of a object
Verb_Token: the lexical entry of a verb
Semantic_Error_Procedure: the procedure when an error exist in the semantics
Semantic_Network: semantic network
BEGIN
    FOR(i=First_Token to Last_Token)
    BEGIN
        IF(exist Token which Syntatic_Part is a Verb) THEN
        BEGIN
            search Subjective_Token and Objective_Token that is 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
```

Figure 14. Structure of the concept sequence layer



- 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 also a basis of encoding of the input sentence.

Semantic Concept Hierarchy

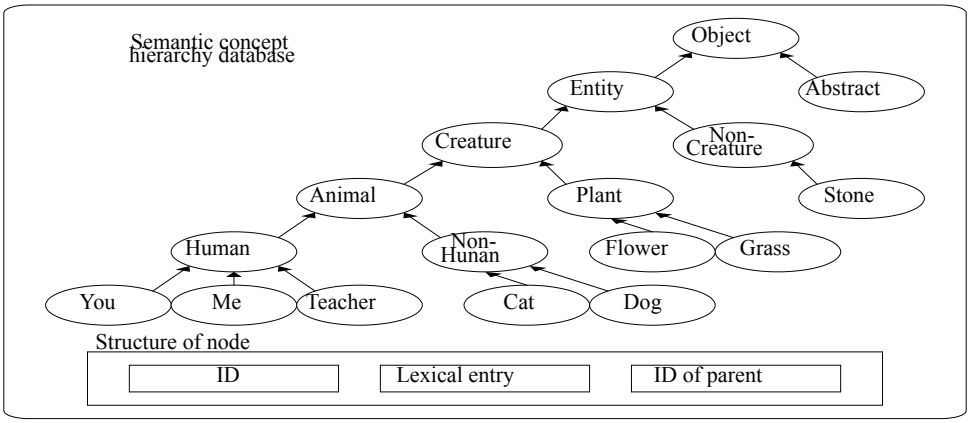
The semantic concept hierarchy defines the relation of the nouns according to the meaning of the nouns. The semantic restrictions of subject and object in the concept sequence layer directs them to get their restrictions via the pointers. The structure of the semantic concept hierarchy is shown in Figure 15. The detailed contents are explained as follows:

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

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. Figure 16 takes

Figure 15. Structure of semantic concept hierarchy



"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 "「」➔「」➔「」" and the path of the object is "「」➔「」➔「」".

Learning Mechanism of Semantics

In the procedure of semantics processing there could be some inconsistency between the system and the user. Because of this problem, the system should provide a learning mechanism (Kim & Moldovan, 1995) for reducing the differences between the system and the user. The proposed learning mechanism is divided into two parts: generalization and specialization.

Generalization

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

There are two conditions of generalization when differences appear between the system and the user:

- The restriction layer of the system is greater and equal to the restriction layer considered by the user, as shown on the left side of Figure 17. Should the user decide that one of the oblique nodes should be corrected, the system would find the lowest common parent node (meshed node) of the oblique node and the restriction node (black node) as the new restriction.
- The restriction layer of the system is less than the restriction layer considered by the user as shown on the right side of Figure 17. If the user decides that the meshed node on top of the restriction node (black node) should be corrected, this meshed node would become the new restriction node.

Specialization

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

If a user decides that one of the nodes (oblique node) under the restriction node is illegal in semantics, the system would change the restriction by eliminating all the illegal nodes under the restriction node

Figure 16. Example of semantic verification

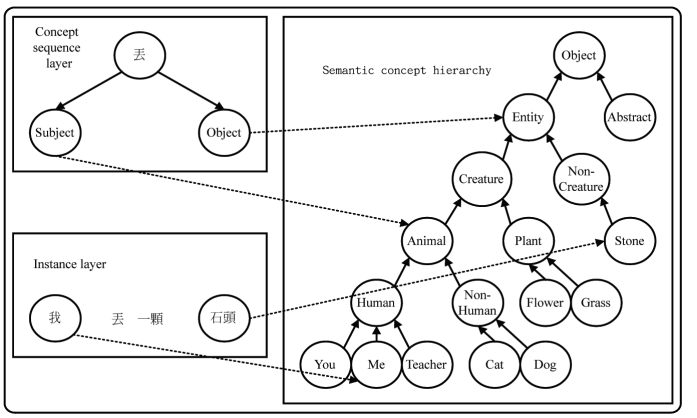


Figure 17. Generalization

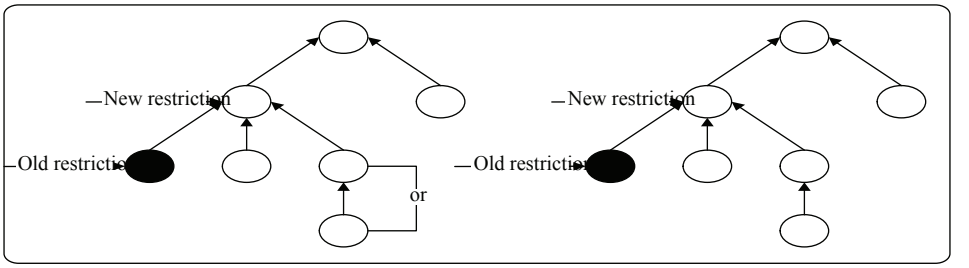
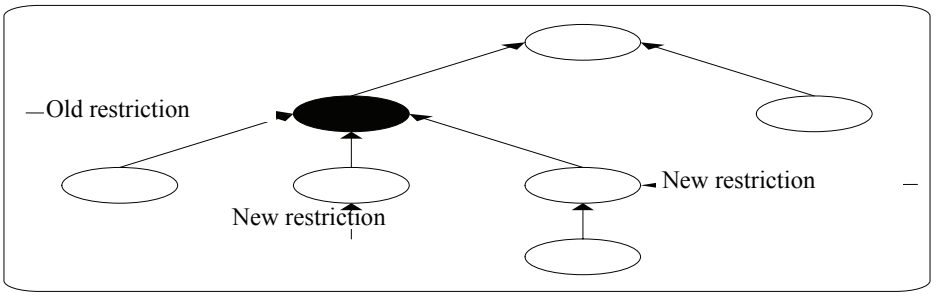


Figure 18. Specialization



and the meshed nodes would become the new restriction nodes, as shown in Figure 18. If the illegal node considered by the user is above or the equal to the restriction node, the system would ask the user what the restriction should be.

Semantic Network

The purpose of the semantic network (Quillian, 1968) is to store and represent the meaning of semantics so as to apply it in the inference mechanism. The semantic network describes the relationship between an object and an event. There are three ele-

ments in a semantic unit: 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.

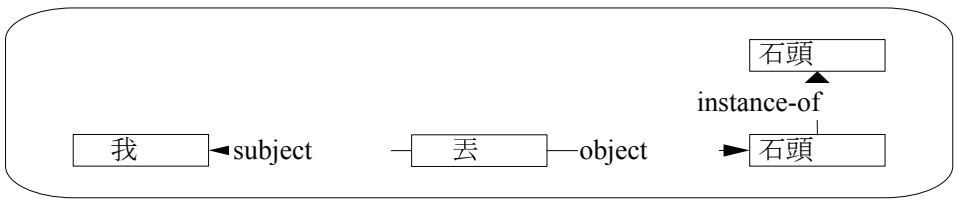
Figure 19 uses the sentence "我丟一顆石頭" as an example. The meshed node is one of the nodes in the concept hierarchy layer, and the actual lexical entry '石頭' in the instance layer uses the arc of the attribute 'instance-of' to form a semantic unit of semantic network with the concept node '石頭'. With the link of the attribute 'instance-of' the actual lexical entry '石頭' inherits 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 is established through linkage by the syntactic parser.

Response System

In the past, learners have had to know exact keywords when using a search engine. However, two identical keywords in different contexts produce different meanings and therefore return different, sometimes undesirable, results. For example, "tell me about the specifications of Bluetooth" and "tell me about the applications of Bluetooth," although having the same keyword, have different meanings. The first sentence concerns hardware specifications of Bluetooth, and the second, software applications of Bluetooth. Learners then would have to filter the data from the results by themselves. The Bayesian Network(BN) and Natural Language Understanding(NLU) (Arai, Wright, Riccardi, & Gorin, 1998; Carpenter & Chu-Carroll, 1998; Kuhn & De Mori, 1995; Miller & Bobrow, 1994; Pieraccini & Levin, 1992) are used to decipher ambiguous sentences and evaluate the searching preferences of different learner.

Before attempting to develop a system, the application domain should be defined as it is very difficult to solve problems of uncertain domain. The application domain of the proposed system is set to the basic computer concept. As shown in Figure 20, understanding natural language queries for a specific application domain involves parsing the input query into a series of domain-specific keywords and searching for the goal of the learner's query. A searching goal assumes that within a restricted application domain, there is a finite set of semantic keywords (M) as well as a finite set of searching goals (N). The searching goals SG_i and keywords K_i are all binary decisions, and the keyword K_i is true if it appears in the speech. In this way, the

Figure 19. An example of semantic network



proposed system can formulate the NLU problem by making M binary decisions with N BNs. The BN for the searching goal SG_i takes the input as a set of keywords K extracted from the learner's query. The BN then gives a posterior probability P(SG | K) for the binary decision. The connection of the BN assumes conditional independence among the set of keywords K, meaning that there are direct links between the goal and the concept nodes and no linkages among the concepts nodes. This is equivalent to a naïve Bayes formulation.

When applying the Baysian rule the proposed system assumes that the searching goal SG_i is present if P(SG | K) is greater than a threshold θ or absent otherwise. θ may be set to 0.5 for simplicity because $P(SG_i = 1 \mid K) + P(SG_i = 0 \mid K)$ is equal to 1. This formula provides a method to reject out-of-domain queries(OODQ). A query is classified as OODQ when all BNs vote negative for their corresponding goals. Assuming that the searching sentence contains i keywords denoted by KW_i, every keyword affects the searching goal. The searching sentence can be represented as Figure 20 and Formula 1.

According to the Minimum Description Length (MDL) principle, every node in the BN provides the complexity of the network by a magnitude of $L_{network}$. Lower values for the $L_{network}$ reflect lower network complexities. Each node also provides the accuracy in modeling the data by a magnitude of L_{data}. Lower values for L_{data} reflect higher accuracy. In this way, the total description length L_{total} provided by the given node is defined by $L_{total} = L_{network} + L_{data}$. The total description length of a network is the sum of all the concept nodes in the network. The trained BN topologies is shown in Figure 21. There are two keyword groups (maximal sets of nodes that are all pairwise linked)—(SG, K_1, K_2) and (SG, K_3) which show that the keyword groups can communicate through the separator node SG. Each keyword group K_i relates to a joint probability P(SG, K_i). The keyword group (SG, K_1, K_2) relates to the joint probability P(SG, K_1, K_2), and the keyword group (SG, K_3) relates to the joint probability P(SG, K3). Given a learner's query, the proposed system derives the presence and absence of the various keywords K and updates the joint probability according to Formula 2. The updated joint probability is eventually marginalized to produce a probability for the searching goal $P^*(SG)$.

Figure 20. Belief network of keywords and searching goal

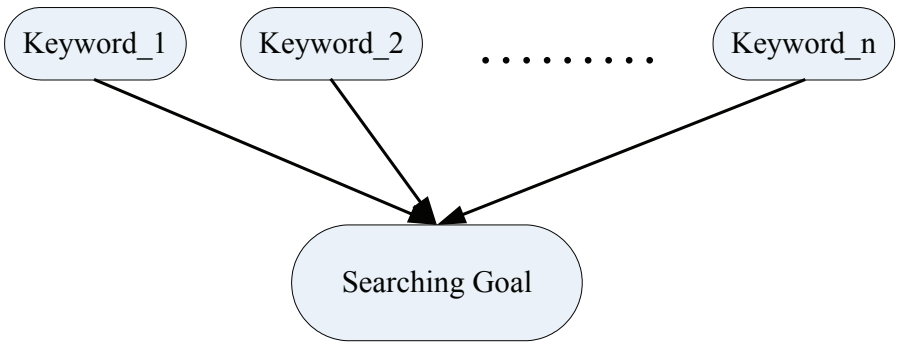
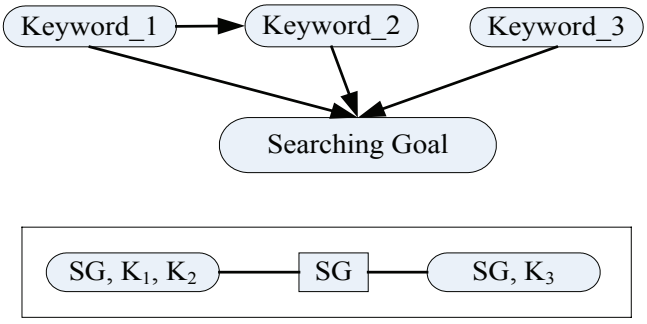


Figure 21. Trained topology of BN and keyword groups



Formula 1.

$$P(SG \mid KW_1, KW_2, ..., KW_n) = \frac{P(KW_1, KW_2, ..., KW_n, SG)}{P(KW_1, KW_2, ..., KW_n)} = \sum P(KW_1, KW_2, ..., KW_i) P(KW_2 \mid KW_1) P(KW_3 \mid KW_2) \cdots P(KW_n \mid KW_{n-1}) \quad (1)$$

KW_1, KW_2, …., KW_n : search keywords
SG : Searching Goal

The BN framework from NLU is extended to mixed-initiative dialog modeling. The idea is to enable BNs to automatically detect missing or spurious keywords according to domain-specific constraints captured by their probabilities. If a missing keyword is detected, the BN prompts the dialog to display the necessary information to the learner. If a spurious keyword is detected, the BN prompts the dialog to notify the learner regarding the unnecessary information. Automatic detection of missing and spurious keywords is achieved by the technique of backward inference which involves probability propagation within the BN. Considering the inferred searching goal SG_i for a given learner's query, the goal node of the corresponding BN is instantiated (to either 1 or 0) to test the network's reliability in each of the input keywords. If the BN topology is assumes conditional independence among the keywords, the updated probability of the concepts would be $P(K_j \mid SG)$. However, in the proposed BN in which the keywords depend on each other, the updated searching goal probability $P^*(K_i)$ would propagate to update the joint probabilities of each keyword group $P^*(K_i, SG_i)$. In this way, each $P^*(K_j)$ can be obtained by marginalization. This procedure is described by Formula 3 and is similar to the procedure described by Formula 2 for updating concept probabilities.

Based on the value of $P^*(K_j)$, the system makes a binary decision (by thresholdθ) regarding whether K_j should be present

Formula 2.

$$P^*(SG_i, K) = P(SG_i \mid K)P^*(K) = P(SG_i, K)\frac{P^*(K)}{P(K)} \quad (2)$$

$P^*(K)$: Initialized by the presence or absence of the concepts in the learner's query.
$P(SG_i, K)$: Joint probability obtained from the training set.
$P^*(SG_i, K)$: Updated joint probability.
* : Denotes an updated probability with knowledge about the presence or absence of the various concepts in the learner's query.

Formula 3.

$$P^*(K, SG_i) = P(K \mid SG_i)P^*(SG_i) = P(K, SG_i)\frac{P^*(SG_i)}{P(SG_i)} \quad (3)$$

$P^*(SG_i)$: Updated from instantiating the searching goal node.
$P(K, SG_i)$: Joint probability of the keyword group obtained from the training set.
$P^*(K, SG_i)$: Updated joint probability of the keyword group.

or absent. This decision is compared with the actual occurrence of K_j in the learner's query. Should the binary decision indicate that K_j is absent and it appears in the input query, the keyword is labeled spurious and the dialog would invoke an explanation. If the binary decision indicates that K_j should be present but it is absent from the query, the keyword is labeled missing and the dialog would invoke the prompting act.

Using keywords and searching goals to categorize knowledge domains is not always desirable as some learners just browse casually in which case a wider variety of search results is preferred. Therefore, the system records browsing habits as another factor that affects the knowledge domain. The corresponding BN is shown as Figure 22 and the probability is described by Formula 4.

The last part is to determine the content and degree of the response to learner. The learning results could be judged by some type of test where a high score indicates a good learning effect and a low score indicates a poor learning effect. The modified BN is shown as Figure 23 and the probability is described by Formula 5 and 6.

EXPERIMENTAL RESULTS

The implementation uses the following example sentence "[illegible]" to describe the process of each step.

Figure 22. Bayesian network of knowledge domain

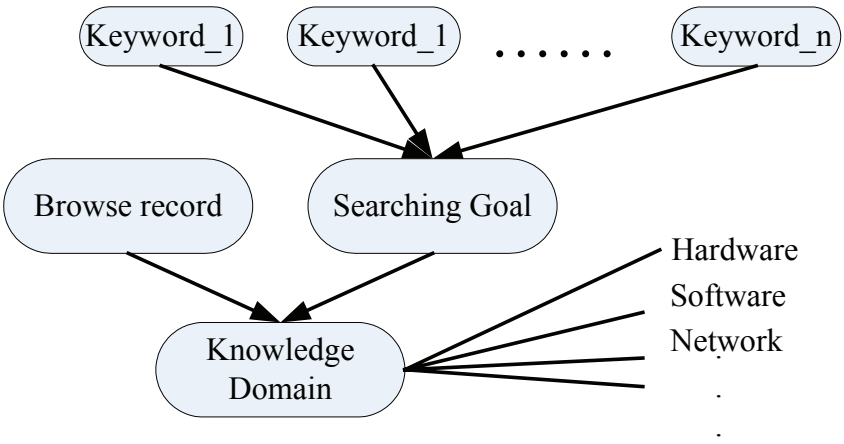
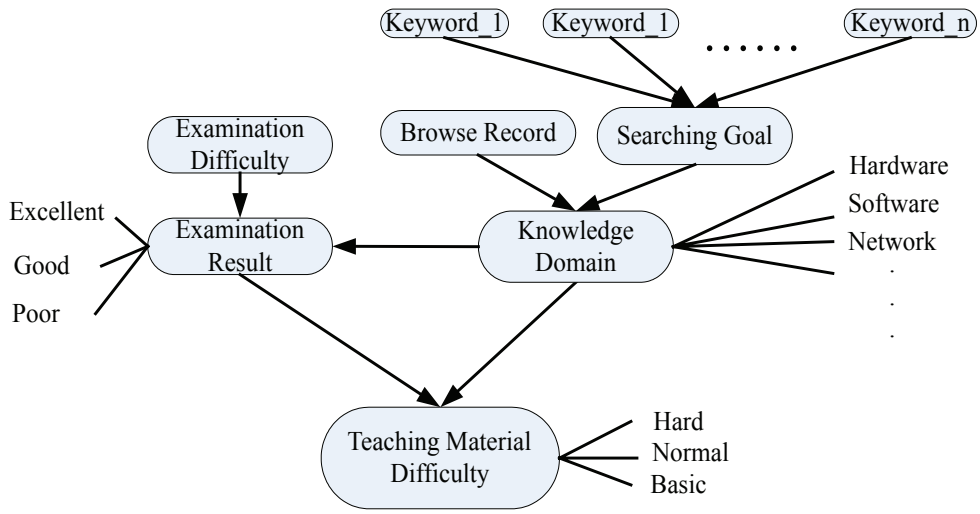


Figure 23. Bayesian network of teaching material difficulty



Formula 4.

$$P(KW \mid BR, SG) = \frac{P(BR \mid KW, SG)P(KW \mid SG)}{P(BR \mid SG)} \quad (4)$$

P(KD|BR,SG): Probability of knowledge domain
P(BR): Probability of learner's browse record.
P(SG): Probability of searching goal.

Formula 5.

$$P(ER \mid ED, KD) = \frac{P(ED \mid ER, KD)P(ER \mid KD)}{P(ED \mid KD)} \quad (5)$$

P(ER|ED,KD): Probability of examination result.
P(ED): Probability of examination difficulty.
P(KD): Probability of knowledge domain.

Formula 6.

$$P(TMD \mid ER, KD) = \frac{P(ER \mid TMD, KD)P(TMD \mid KD)}{P(ER \mid KD)} \quad (6)$$

P(TMD|ER,KD): Probability of teaching material difficulty.
P(ER): Probability of examination result.
P(KD): Probability of knowledge domain.

Segmentation System

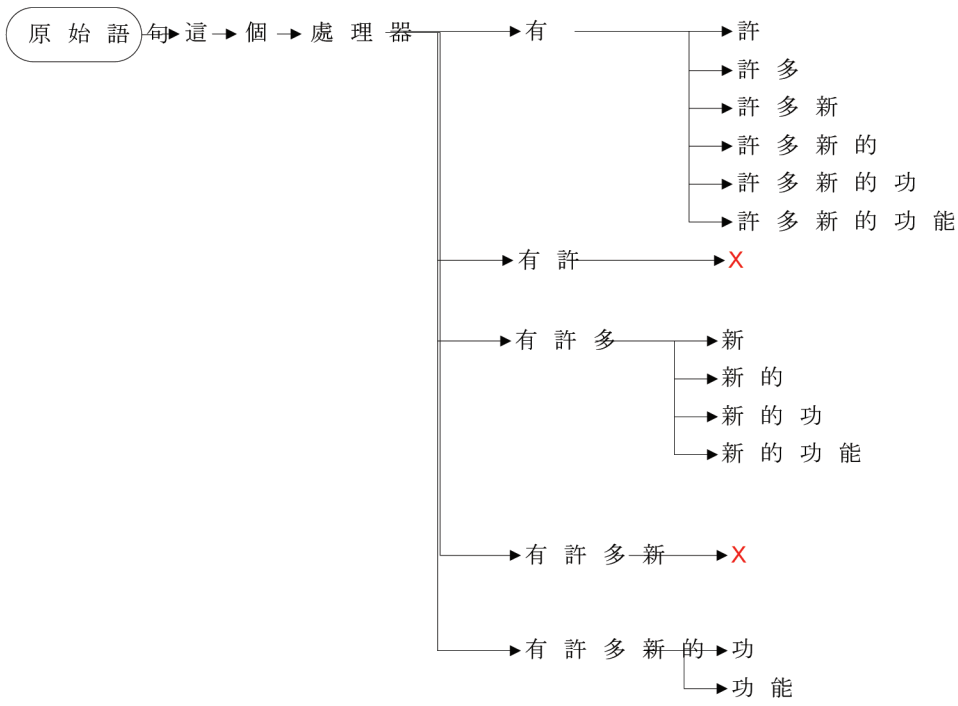
The segmentation system divides the example sentence into lexical entries and gives each lexical entry a suitable part of speech as follows. Figure 24 shows the list branching from the first to the fourth layer. Table 4 shows the segmentation table of sentence in Figure 24

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))
個((()(Pq))
2. : (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))

Figure 24. List of separating from the first layer to the fourth layer



((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 process of the first and second lexical entries are shown in Figure 25.

Because the first lexical entry contains only its 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

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
	⋮	⋮	
	⋮	⋮	

Figure 25. Process of syntactic analysis

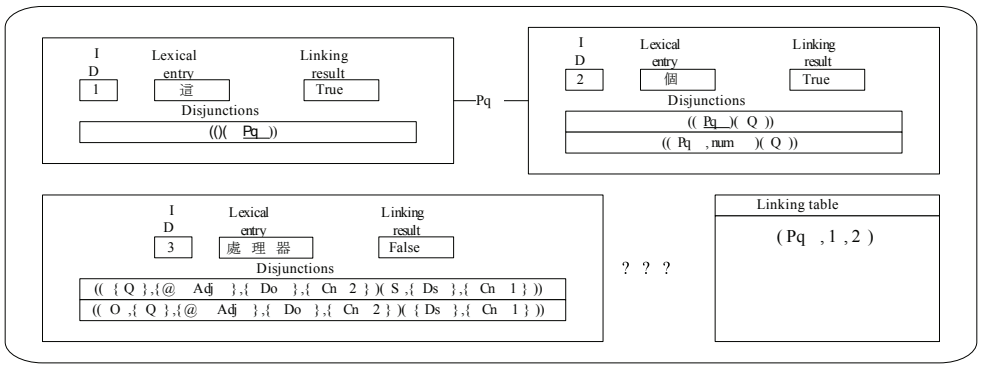
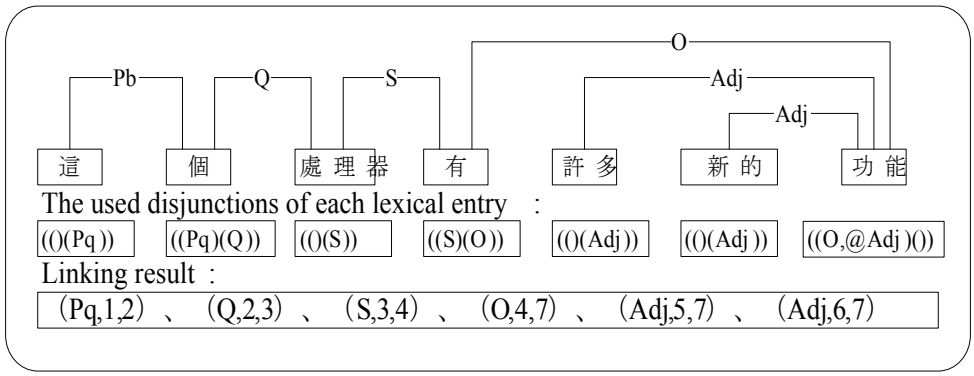


Figure 26. Result of the syntactic analysis



result of the syntactic analysis is shown in Figure 26.

Semantic Analysis System

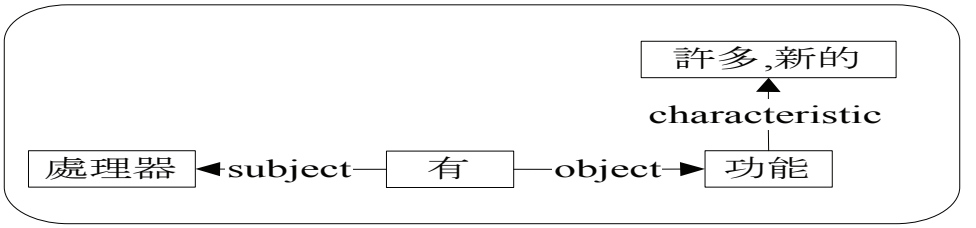
After the syntactic analysis procedure, the system can determine that the fourth lexical entry is a verb and that the third and seventh lexical entries are the subject and object, respectively, according to linkages (S,3,4) and (O,4,7). The system determines the correctness of the semantics, and the sentence is classified as legal if it conforms to semantic restrictions. Finally, it transforms these linkages into semantic meanings as shown in Figure 27:

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

CONCLUSION AND FUTURE WORK

This paper applies the linking of grammar to describe the syntactical construction of sentences and proceeds to verify and record the semantics according to the construction. At the end, the proposed system replies to the user by finding the suitable response derived from the response database. According to the implementation results, the proposed system could correctly describe

Figure 27. Semantic network of example sentence



the relationship between lexicons. Furthermore, with the use of memory-based parsing, the proposed system can check the correctness of the semantics and provide a learning mechanism by changing the semantic restrictions of the concept sequence layer. Finally, by classifying the response databases and using the verb of the sentence as the search key, the response system greatly reduces the amount of response results and increases speed and accuracy. Of the numerous possible applications in the future, one could further develop the interaction between computer and user by utilizing user characteristics and habits, allowing the proposed method to generate a variety of improved responses. With the continual progression of the computing age and the increasing need for improved methods of automated communication, the future of the proposed system remains both worthwhile and practical.

REFERENCES

Arai, J., Wright, G., Riccardi, & Gorin, A. (1998). Grammar fragment acquisition using syntactic and semantic clustering. *The 4th International Conference on Spoken Language Processing*.

Carpenter, B. & Chu-Carroll, J. (1998). Natural language call routing: A robust, self-organizing approach. *The 4th International Conference on Spoken Language Processing*.

Chen, J.-F., Lin, W.-C., Jian, C.-Y. (2003a). Using the keyword in context segmentation method for collaborative design in a Chinese Web site. *The 10th ISPE International Conference on Concurrent Engineering: Research and Applications*, pp. 967–975.

Chen, J.-F., Lin, W.-C., Jian, C.-Y., Ho, T-.Y., & Dai, S.-Y. (2003b). Using the keyword in context segmentation method for a Chinese Web site. *2003 International Conference on Computer-Assisted Instruction*, National Taiwan Normal University, Taipei, Taiwan, pp. 74-80.

Chen, J.-F., Lin, W.-C., Jian, C.-Y., & Hung, C-.C. (2003c). A Chinese automatic interactive feedback system for applying in a Web site. *The Second International Human.Society@Internet Conference*, pp.238-248.

Chen, J.-F., Lin, W.-C., Jian, C.-Y., & Hung, C-.C. (2005). A Chinese interactive feedback system for an e-learning Web site. *Journal of Information Science and Engineering, 21*(5), 929-957.

Chen, K. J., & Liu, S. H. (1992). Word identification for mandarin Chinese sentences. *Proceedings of the Fifteenth International Conference on Computational Linguistics*, Nantes, pp.101-107.

Chung, M. & Moldovan, D. (1993). Parallel memory-based parsing on SNAP. *Parallel Processing Symposium, Proceedings of Seventh International Conference*, pp. 680-684.

Chung, M. & Moldovan, D. (1994a). Applying parallel processing to natural-language process-

ing**.** *IEEE Expert [see also IEEE Intelligent Systems], 9*(1), 36-44.

Chung, M. & Moldovan, D. (1994b). Memory-based parsing with parallel marker-passing. *Proceedings of the Tenth Conference on Artificial Intelligence for Applications*, pp. 202-207.

Kim, J-.T. & Moldovan, D.I. (1993). Acquisition of semantic patterns for information extraction from corpora. *Proceedings of Ninth Conference on Artificial Intelligence for Applications*, pp.171-176.

Kim, J-.T. & Moldovan, D.I. (1995). Acquisition of linguistic patterns for knowledge-based information extraction. *IEEE Transactions on Knowledge and Data Engineering, 7*(5), 713-724.

Kuhn, R. & De Mori, R. (1995). The application of semantic classification trees for natural language understanding. *IEEE Trans. Pattern Anal. Machine Intell., 17*, 449-460.

Miller, S. & Bobrow, R. (1994). Statistical language processing using hidden understanding models. *The Human Language Technology Workshop*, 278-282.

Pieraccini, R. & Levin, E. (1992). Stochastic representation of semantic structure for speech understanding. *Speech Communication, 11*, 283-288.

Quillian, M.R. (1968). Semantic Memory. In *Semantic information processing*, pp. 216-270. Cambridge, MA: MIT Press.

Sleator, D. & Temperley, D. (1991). *Parsing English with a link grammar.* Carnegie Mellon University Computer Science technical report CMU-CS-91-196.

Jui-Fa Chen (陳瑞發) received his PhD, MS, and BS degrees in the department of computer science and information engineering from TamKang University (TKU), Danshui, 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 software engineering.

Wei-Chuan Lin (林 偉 川) received his PhD, MS, and BS degrees in the department of computer science and information engineering from TamKang University (TKU), Danshui, 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 assocated professor in the Department of Information Technology in Takming College, Nei-Hoo District, Taipei, Taiwan, since 1993. His research interests include intelligent avatar, peer-to-peer communication, and software engineering.

Chih-Yu Jian (簡志宇) eceived his PhD, MS, and BS degrees in the Department of Computer Science and Information Engineering from TamKang University (TKU), Danshui, Taipei, Taiwan, in 2007, 2001, and 1999, respectively. His research interests include intelligent avatar, peer-to-peer communication, and software engineering.

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