IGI PUBLISHING
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

ITJ 4440

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**

Applying Semantic Agents to Message Communication in e-Learning Environment

Ying-Hong Wang, Tamkang University, Taiwan

Chih-Hao Lin, Asia University, Taiwan

ABSTRACT

A traditional distance learning system requires supervisors or teachers always available on online to facilitate and monitor a learner's progress by answering questions and guiding users. We presents an English chat room system in which students discuss course contents and ask questions to and receive from teachers and other students. The mechanism contains an agent that detects syntax errors in sentences written by the online the user and also checks the semantics of a sentence. The agent can thus offer recommendations to the user and, then, analyze the data of the learner corpus. When users query the system, this system will attempt to find the answers from the knowledge ontology that is stored in the records of previous user comments. With the availability of automatic supervisors, messages can be monitored and syntax or semantic mistakes can be corrected to resolve learner-related problems.

Keywords: distance learning; link grammar; semantic check; syntax parser

INTRODUCTION

Distance Learning has become a hot topic in the disciplines of computer science and education in the recent years (Tsang, Hung, Ng, 1999). Furthermore, online learning technologies operating through the Web interface have been developed during the past decade. Because of its ability to incorporate multimedia, the World Wide Web has become an ideal platform for distance learning (Adhvaryu, & Balbin, 1998). Through the Internet, distance learning allows students to enroll in courses and acquire new knowledge. It is a good solution for anyone who does not have enough time to attend traditional classes. Therefore, distance learning now plays a very important role in education (Harris, Cordero, & Hsieh, 1996; Willis, n.d.; Goldberg, 1996; Goldberg & Salari, 1997; Goldberg, Salari, & Swoboda, 1996).

The advantage of the Internet is information sharing. Many applications on the Internet support information interchange, including Telnet, FTP, e-mail, BBS, and chat-rooms. Each participant can communicate with other participants through text-, voice-, and even video-based messages.

However, it is difficult for instructors trace the activities and behaviors of learners in distance learning environments. For example, instructors may need answers to the following questions:

- Do the learners understand the teaching context?
- Are learners talking about the issues indicated by the instructor?
- Do the learners really understand the issues being studied in the course?

Therefore, it is quite useful if there are some automatic supervising mechanisms. These mechanisms can monitor discussions and detect mistakes in grammar. This helps students obtain educational training without the need to go to a classroom. Thus, people can teach or learn anywhere any time.

However, there are many problems with distance learning systems. For example, instructors cannot control learners' activities, instructors cannot stay online forever—the Instructor-off problem—and instructors cannot track of frequent answers and questions (FAQs); thus, learners cannot learn from previous learners and other learners.

To solve the problems mentioned above, this study built up an ontology-based Semantic Agent system that provides supervision and learning-assistance for textual chat rooms. This system was built based on Agent, Link grammar, XML, a learner corpus, and other supporting functions to solve the Instructor-off problems. The system provides a Learning_Angel agent and a Semantic agent. Also, the QA sub-system can collect/analyze frequent mistakes and problems. The Learning_Angel agent is designed to provide monitoring and syntax checking functions online. While discussing in the class, if learners fall behind the topic of discussing courses, Semantic agent can make some comments and/or suggestions. The statistical analyzer then records, classifies, and analyzes the learners' discussion. Furthermore, this discussion can be used to generate QA pairs and update the learner corpus. By means of these resources, instructors can revise or enhance their teaching materials. Learners can also learn from the experience of the previous learners and other learners.

This article is organized as follows: we first describe related works and introduce link grammar and ontologies. The next section presents the architecture of proposed system. The chief processes in the proposed system and evaluations of several related systems then are given. The last part of this article gives conclusions and discusses future researches.

THEORETICAL BACKGROUND

Link Grammar

Link grammar is an English grammar parser system that was proposed by researchers at the School of Computer Science of Carnegie Mellon University (CMU). Link grammar is a scheme for describing natural language (Sleator & Temperley, 1991). Link grammar defines a set of ***words***, which are the terminals of grammar, and each has some ***linking requirements***. The linking requirements of each word are gathered in a ***dictionary***. Figure 1 illustrates the linking requirements defined in a simple dictionary for the following words: *a/the, cat/mouse, John, ran, and chased*.

Each intricately shaped labeled box is defined as a ***connector***. A pair of compatible connectors will join, given that they correspond to the same type. For each black dot, only one connector can be selected. Figure 2 shows that the linking requirements are satisfied in the sentence, *"The cat chased a mouse."*

The linkage can be perceived as a graph, and the words can be treated as vertices, which are connected by labeled arcs. Thus, the graph is connected and planar. The labeled arcs that connect the words to other words on either their left or right sides are ***links.*** A set of links that proves that a sequence of words is in the language of a link grammar is called a ***linkage.*** Thus, Figure 3 shows a simplified form of the diagram, indicating that *the cat chased a mouse* is part of this language.

Table 1 presents an abridged dictionary, which encodes the linking requirements of the above example.

The linking requirement for each word is expressed as a formula that includes the operators *"&"* and *"or," parentheses*, and *connector names*. The "+" or "–" suffix after a connector indicates the direction in which the matching connector must be laid. Thus,

Figure 1. Words and connectors in a dictionary

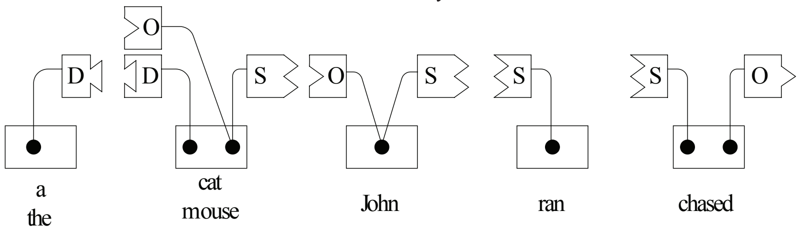


Figure 2. All linking requirements are satisfied

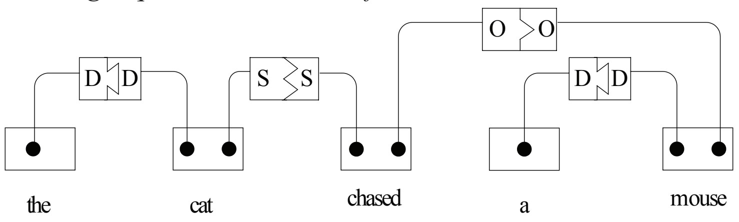


Figure 3. A simplified form of Figure 2

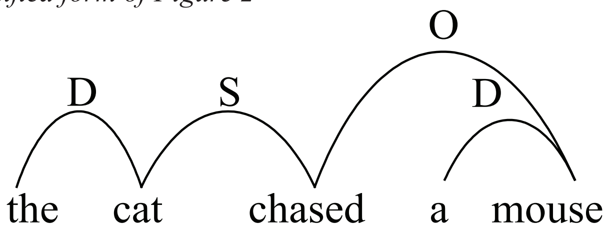


Table 1. The words and linking requirements in a dictionary

words	formula
a / the	D+
cat / mouse	D- & (O- or S+)
John	O- or S+
ran	S-
chased	S- & O+

the farther to the left a connector is in an expression, the nearer the word to which it connects must be.

A sequence of words is a *sentence* form a language defined by the grammar. If links can be established among the words so as to satisfy the formula of each word, then they must satisfy the following ***meta-rules:***

1. **Planarity:** The links do not cross when draw above the words.
2. **Connectivity:** The links suffice to connect all the words of the sequence together.
3. **Ordering:** When the connectors of a formula are traversed from left to right, the words to which they connect proceed from near to far. To understand this, consider a word, and consider two links connecting that word to the word on its left. Compared with the other word, the link connecting the closest word (the shorter link) must satisfy a connector that appears to the left (in the formula) of that connector in the other word. Similarly, a link to the right must satisfy a connector to the left (in the formula) of a longer link to the right.
4. **Exclusion:** No two links may connect the same pair of words.

Using a formula to specify a link grammar dictionary is convenient for creating natural language grammars. However, it is cumbersome for mathematical analysis of link grammars and for describing algorithms for parsing link grammars. Therefore, an alternate method of expressing link grammar is known as ***disjunctive form***, in which each word has an associated set of disjuncts. In disjunctive form, each word of the grammar has a set of disjuncts associated with it. Each disjunct corresponds to one particular way of satisfying the requirements of a word. A disjunct consists of two ordered lists of connector names: the ***left list*** and ***right list***. The left list contains connectors that connect to the left of the current word, and the right list contains connectors that connect to the right of the current word. A disjunct is denoted as

$$((L_1, L_2, \ldots, L_m)(R_n, R_{n-1}, \ldots, R_1)),$$

where $L_1, L_2, \ldots, L_m$ and $R_n, R_{n-1}, \ldots, R_1$ are the connectors that must connect to the left and right, respectively.

Thus, it is easy to see how to translate a link grammar in disjunctive form to one in standard form. Translating a link grammar from disjunctive to standard form can be accomplished as follows:

$$(L_1\&L_2\&\ldots\&L_m\&R_1\& R_2\&\ldots\&R_n).$$

By enumerating all the ways in which the formula cab be satisfied, we can translate a formula into a set of disjuncts. For example, the formula

(A- or ( )) & D- & (B+ or ( )) & (O- or S+)

corresponds to the following eight disjuncts, which may be used in some linkages:

```
( (A,D)    (S,B) )
( (A,D,O)  (B) )
( (A,D)              (S) )
( (A,D,O)  ( ) )
( (D)     (S,B) )
( (D,O)              (B) )
( (D)     (S) )
( (D,O)    ( ) ).
```

To streamline the difficult process of writing the dictionary, we incorporate several other features into the **Dictionary Language**. It is useful to consider connector-matching rules that are more powerful than those which simply require that the strings of the connectors be identical. The most general matching rule is simply a table—part of the link grammar—that specifies all the pairs of connectors that match. The resulting link grammar is still context-free. In the dictionary, a matching rule is used that is slightly more sophisticated than simple string matching. This rule is described below.

A connector name begins with one or more upper case letters, followed by a sequence of lower case letters or *s. Each lower case letter (or *) is a ***subscript***. To determine if two connectors match, we delete the trailing + or - and append an infinite sequence of *s to both connectors. The connectors match if and only if these two strings match under the proviso that * matches a lower case letter (or *).

For example, S matches both Sp and Ss, but Sp does not match Ss. The formula "((A- & B+) or ( ))" is satisfied either by using both A+ and B-, or by using neither of them. Conceptually, then, the expression "(A+ & B-)" is optional. Since this situation occurs frequently, we denote it with curly braces, as follows: {A+ & B-}.

It is useful to give certain connectors to be ability connect to one or more links. This makes it easy, for example, to allow any number of adjectives to attach to a noun. We denote this by putting a "@" before the connector name, and we call the result a ***multi-connector***. A dictionary consists of a sequence of ***entries***, each of which is a list of words separated by spaces, followed by a colon, followed by the formula defining the words, followed by a semicolon.

If a word (such as **move** or **can**) has more than one distinct meaning, then it is useful to be able to give it two definitions. This is accomplished by defining several versions of the word with differing suffixes. The suffix always begins with a "." followed by one or more characters. We use the convention that ".v" indicates a verb and ".n" indicates a noun (among others). When the user types the word "move," the program uses an expression that is equivalent to that obtained by oring the expressions for the two versions of the word. When it prints out the linkage, it uses whichever version is appropriate for that particular linkage. As of this writing, there is no macro facility in the dictionary language. There is reason to believe that using macros would significantly reduce the size of the dictionary while making it easier to understand.

Ontology

Ontologies are important in various fields, such as knowledge engineering, natural language processing, intelligent information integration, and knowledge management. An ontology provides a shared, common representation of a domain that can be communicated between heterogeneous and widespread application systems. Ontologies have been developed in AI to facilitate knowledge sharing and reuse. An ontology provides an explicit conceptualization that describes the semantics of data. (Ide, 2003; Ide, Reppen, & Suderman, 2002).

Current computer systems are changing from single isolated devices to entry points into a worldwide network of information exchange. Therefore, support for data exchange, information, and knowledge is becoming a key issue in communication technology. It can facilitate communication between people and application systems. The provision of shared, common domain structures is becoming essential for describing the structure and semantics of information exchange. Now, Internet technology and the World Wide Web comprise the main technology infrastructure for online information exchange. It is not surprising to see that a number of initiatives are providing notations for data structures and semantics. These include:

- the resource description framework (RDF);
- the Extendible Markup Language (XML);
- XML schemes providing standards for describing the structure and semantics of data;
- the transformation language of XSL (XSL-T); and
- various querying languages for XML (XQL, XML-QL).

With the large number of online documents, several document management systems have entered the market. However, these systems have several weaknesses, which are explained below:

- Searching information: Existing keyword-based search schemes retrieve irrelevant information about the use of certain word in a different context, or they may miss information when different words about the desired content are used.
- Extracting information: Human browsing and reading is currently required to extract relevant information from information sources, as automatic agents lack the common sense knowledge required to extract such information from textual representations and fail to integrate information spread over different sources.
- Maintaining weakly structured text resources is difficult and time-consuming when the amount of resources becomes hug. Keeping these resources consistent, correct, and up-to-date requires a mechanized representation of semantics and constraints that help to detect anomalies.
- Automatic document generation takes advantage of the usefulness of adaptive Web sites, which enable dynamic reconfiguration according to user profiles or other relevant aspects. The generation of semi-structured information presentations from semi-structured data requires a machine-accessible representation in semantic information sources.

In the near future, semantic annotations will make structural and semantic definitions of documents possible, thus opening up new possibilities:

- intelligent search instead of keyword matching;
- query answering instead of information retrieval;
- document exchange between departments via XSL translations; and
- definitions of views on documents.

Depending on their level of generality, different types of ontologies may be identified and play different roles. The following are some ontology types:

- Domain ontologies: these capture knowledge that is valid for a particular type of domain.
- Metadata ontologies: these, such as Dublin Core (Weibel, Gridby, & Miler, 1995), provide a vocabulary for describing the content of on-line information sources.
- Generic or common sense ontologies: these aim to capture general knowledge about the world.
- Representational ontologies: these do not commit themselves to any particular domain.

Ontological engineering is concerned with the principled design, modification, application, and evaluation of ontologies. Ontologies can be adopted in situations where the capability for representing semantics is important to overcome XML's disadvantages in terms of maturity.

One well-known ontology language—the OWL Web Ontology Language (McGuinness & Harmelen, 2004)—is designed to process information instead of just presenting information to humans. OWL facilitates greater machine interpretability of Web content by providing additional vocabulary along with formal semantics.

OWL has three increasingly-expressive sublanguages: OWL Lite, OWL DL, and OWL Full. OWL is used when the information is contained in documents and needs to be processed by applications. It also opposes situations in which the content only needs to be presented to humans. OWL can be used to explicitly represent the meanings of terms in vocabularies and the relationships between those terms. This representation of terms and their interrelationships is called an ontology. OWL has more facilities for expressing meaning and semantics than XML, RDF, and RDF-S do; thus, OWL goes beyond these languages in its ability to represent machine interpretable content on the Web. OWL is a revision of the DAML+OIL Web ontology language and incorporates lessons learned from the design and application of DAML+OIL.

SYSTEM ARCHITECTURE

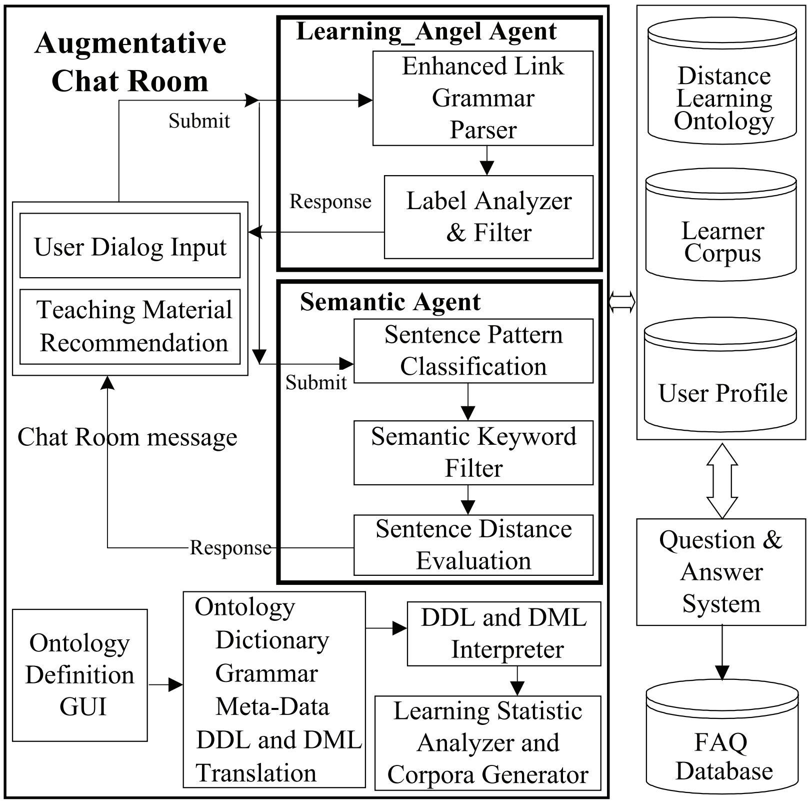
This section introduces the proposed Chat Room System, which is shown in Figure 4. The left part of the figure shows the components of the Augmentative Chat Room, the flow of Chat Room supervisors, and the Ontology Definition process. This system has two kinds of online supervisors: (1) Learning_Angel Agent and (2) Semantic Agent. The right part of the figure shows the database, which includes the Distance Learner Ontology, Learner Corpus Database, and User Profile Database. The Question and Answer System analyzes the Corpus and user profile to collect questions that are frequently asked by learners. Finally, the data is sent to the FAQ system, which generates new OA-pairs.

The following sections describe the chief components of the proposed system.

Domain Specific Sentences

Before introducing each chief sub-system of the proposed chat room, we first will explain why this reason is restricted the research to specified domain. Domain specific sentences refer to those sentences that frequently appear in certain application domain texts but rarely in others. The following are some characteristics

Figure 4. The system architecture and operation flow



domain specific sentences (Li, Zhang, & Yu, 2001):

1. the vocabulary set is limited;
2. word usage is based on patterns;
3. semantic ambiguities are rare; and
4. term and jargon appear frequently in the domain.

It is fairly hard to apply semantic-level analysis to common language conversation. Take the following two sentences as examples. The syntax of the two sentences "The car is drinking water" and "The data is pushed in this heap" is correct. But the meaning of these sentences is incorrect. In the real world, a car cannot drink water. In a data structure course, a heap cannot be pushed. In fairy tales, cars perhaps can drink water or maybe even cola. Therefore, in different situation, the meaning of such a sentence might be different. For this reason, the domain must be restricted. Thus, the proposed system deals with only the "Data Structure" domain. The same scheme can be extended to other domains.

For the above reasons, the class topic and user messages are all restricted in a domain. Thus, the terms in the data structure are limited and can be pre-defined in the system ontology to support the functions of syntax and semantic analysis.

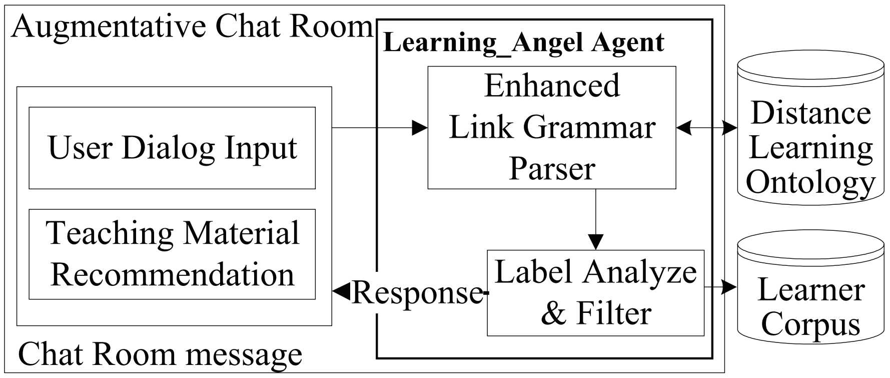
Furthermore, the system manager can load pre-defined terms about the Data Structure through the Ontology Definition GUI during system initialization. This Ontology Definition GUI interface is designed to provide the ability to generate another scaffolding teaching material. The ontology built for this system includes the Dictionary, Grammar, and Meta-Data. This process of ontology creation is designed to transform the pre-defined ontology into DDL and DML form. Finally, the DDL and DML Interpreter can interpret the ontology and then send the data to the Corpora Generator, which records the data to the Distance Learning Ontology and Learner Corpus databases.

Learning_Angel Agent

The Learning_Angel Agent is designed to be a supervisor. It can constantly detect syntax errors online as online users submit messages to the system. It can then correct the learners' errors.

The Learning_Angel Agent workflow is shown in Figure 5. Strictly speaking, when learners in the Augmentative Chat Room submit sentences to the Learning_Angel Agent, it will forward them to the Link Grammar Parser. Then, the Link Grammar Parser will query the ontology to get the tags for the input sentences (Wible, Kuo, Chien, & Taso, 2001). Meanwhile, the Link Grammar Parser will send the tags and sentences to the Label Analyzer & Filter, which can find out if there are any incorrect linkages. In addition, if the input messages have grammar errors, the Label Analyzer & Filter can detect them, search for suitable sentences from the Learner Corpus, and convey them to the online learners. In addition, the Label Analyzer & Filter analyzes the links of input words' sequences sent by the Link Grammar Parser to check whether the links between words satisfy the meta-rules in terms of planarity, connectivity, ordering, and exclusion. If the input words' sequences have received particular tags from the Learning_Angel Agent, the Label Analyzer & Filter will record them in the Learning Corpus and efficiently send the correct information to the online learners.

Figure 5. Workflow of the Learning_Angel agent



Semantic Agent

This section describes the Semantic Agent. The Semantic Agent is also designed to be an online supervisor. It can check the semantics of each sentence. In the distance-learning environment, learners sometimes may fall behind the in courses discussions. They may not understand the course topic clearly and, thus, may make some semantic level mistakes. For example, learners may submit sentences that do not make sense of the course topic. The Semantic Agent can analyze the data in the Learner Corpus and make some comments or give suggestions to the users.

Two proposed methodologies for constructing the Semantic Agent are proposed in this article. One is the Semantic Link Grammar, which is based on Link Grammar, and the other is the Semantic Relation of Knowledge Ontology, which is based on ontology technology. The Semantic Link Grammar can use the algorithm from the Link Grammar to parse sentences. However, it is quite difficult to modify the dictionary, which consists of correct semantic meanings. It will take a lot of money and time for linguistic classification and the performance is not very good.

In this article, we will employ the second method: the **Semantic Relation of Knowledge Ontology**. This method can be used to evaluate the distance between specified keywords. The Semantic Agent subsystem shown in Figure 6 contains three processes.

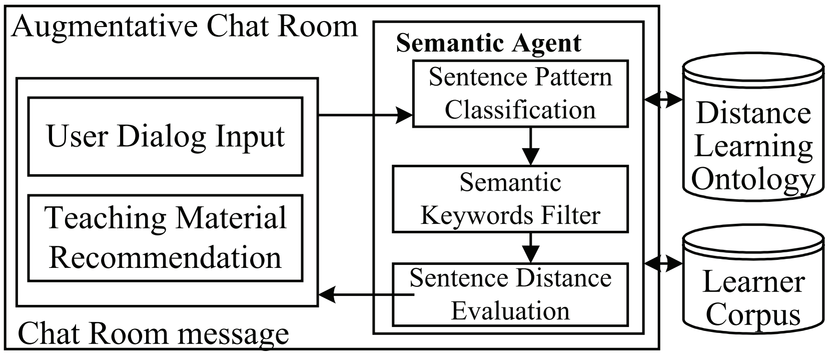
1. Sentence Pattern Classification

Firstly, the sentence pattern classification process classifies input sentence patterns. Currently, this process can only identify Simple and Negative Sentence Patterns. Other types of sentence patterns are ignored. After classification is completed, each sentence will be tagged with its sentence pattern information and passed to the Semantic Keyword Filter to be processed.

2. Semantic Keywords Filter

According to the sentence pattern information, the Semantic Keyword Filter will extract the sentence's keywords and query the ontology (in this article, the Data Structure Ontology) to get the keywords' IDs.

Figure 6. Workflow of the Semantic Agent


Augmentative Chat Room
Semantic Agent
Sentence Pattern Classification
User Dialog Input
Semantic Keywords Filter
Teaching Material Recommendation
Sentence Distance Evaluation
Chat Room message
Distance Learning Ontology
Learner Corpus

3. Sentence Distance Evaluation

We have designed a tree structure that can be used to encode the Data Structure Ontology shown in Figure 7. Each node (i.e., keyword) in the ontology has its unique ID number. The Sentence Distance Evaluation component uses the ID information to calculate the distance between two keywords.

For example, consider: **The tree doesn't have pop method.** After the sentence is processed by the Semantic Keyword Filter, the keywords will be "tree" and "pop." Table 2 shows some of the predefined IDs and keywords in the ontology. Here, we know that the ID number of the keyword "tree" is 4 and that of "pop" is 33.

And the following Figure 8 shows the tree-view of the above keyword ontology structure shown in Table2. In the Schema of the Data Structure Course, the depth value of the ID indicates the keyword's distance from the root node. For example, the depth value between "Stack" the "Course" is 2 and that of "pop" and "Course" is 3. Thus, we transfer the depth value calculation to string comparison with the keywords' ID values. We examine the ID string from left to right. The left-most digit represents the top category in the domain of the Data Structure. The following digit represents the subcategory related to its parent and so on. According to the design of the ontology structure, the problem of evaluating the distance between two nodes can be simplified as the string comparison problem. If the left most digits of two keywords' IDs are not equal, then we conclude that the two keywords are absolutely unrelated.

As Figure 8 shows, the ID of the keyword "tree" is 4 and that "pop" is 33. We discover that the left-most digits of the two keywords' IDs are not the same. So "tree" and "pop" in the example sentence is not related. This information is then combined with the pattern classification obtained from Sentence Pattern Classification component. Since the result for this example sentence is negative, the semantic checker will con-

Table 2. The ID of keywords in the knowledge ontology

Keywords	Stack	Tree	pop	Push
ID	3	4	33	34

Figure 7. Schema of the data structure course

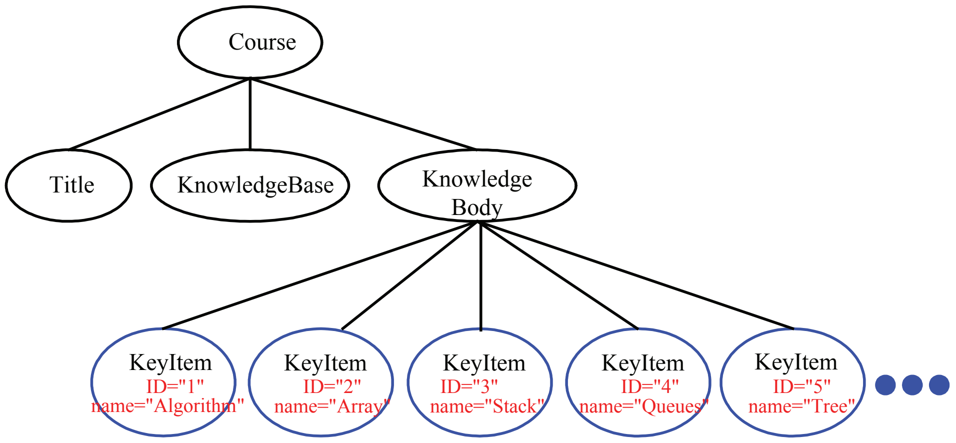
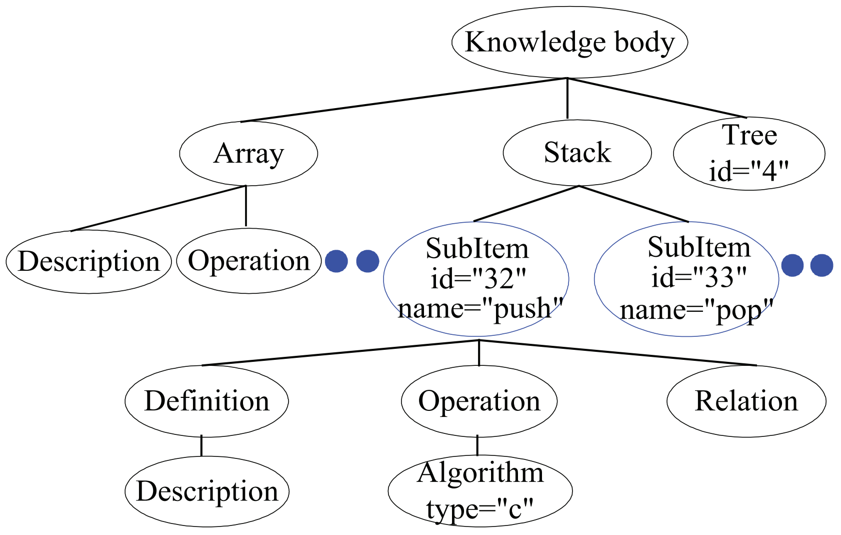

Course
Title
KnowledgeBase
Knowledge Body
KeyItem ID="1" name="Algorithm"
KeyItem ID="2" name="Array"
KeyItem ID="3" name="Stack"
KeyItem ID="4" name="Queues"
KeyItem ID="5" name="Tree"

Figure 8. Knowledge ontology representation of the Data Structure (taking "tree" and "push" as examples)



clude that the semantics of this sentence are still correct.

In our discussion of Sentence Pattern Classification, we will focus on the semantic checking of a simple sentence pattern and a negative sentence pattern. Based on the analysis performed by the Link Grammar, Tables 3 and 4 show the sentence pattern classifications of simple sentence pattern and negative sentence pattern. Based on these two tables, the keywords can be extracted from learners' sentences.

Other examples of such sentences are as follows:

- I push the data into a tree.

This is a simple sentence pattern. In this simple sentence pattern, we find the ID of the keyword "push" is 32 and that of "tree" is 4. This means that these two words are not in the same branch. Thus, we know that they are not mutually related, so there is a semantic mistake with this sentence:

- The tree doesn't have pop method.

This is a negative sentence, even though the IDs of "tree" and "pop" are not in the same branch. Thus, the semantics meaning of this sentence is still correct.

However, in the Sentence Pattern Classification process, if learners submit the following W/H sentence pattern or yes/no sentence pattern sentences, it will be hard to determine the relationships between the subjects and objects in the sentence patterns:

- How can I push data in to Stack?
- Is Tree has a method pop?
- What method can be used in link-list?

In this section, we have ignored this kind of question pattern checking. But in the following section, we will use the Question and Answer System to answer learners'

Table 3. Simple sentence pattern and link grammar tags

		Active voice	Passive voice
Simple pattern	present past future	Ss or Sp Ss or Sp Ss or Sp + I	Ss or Sp + Pv Ss or Sp + Pv Ss or Sp + Ix + Pv
Continuous pattern	present past future	Ss or Sp + PP Ss or Sp + PP Ss + If + PP	Ss or Sp + ppf + Pv Ss or Sp + ppf + Pv Ss or Sp + If + ppf + Pv
Proceed pattern	present past future	Ss or Sp + Pg Ss or Sp + Pg Ss or Sp + Ix + Pg	Ss or Sp + Pg + Pv Ss or Sp + Pg + Pv none
Perfective continuous pattern	present past	Ss or Sp + ppf + Pg Ss or Sp + ppf + Pg	
Comments: Ss and Sp: connects subject nouns to finite verbs. I: connects infinitive verb forms to certain words, such as modal verbs and "to." PP: connects forms of "have" with past participles. PV: connects forms of the verb "be" with past participles. Pg: connects forms of the verb "be" with present participles. ppf: connects forms of the verb "be" with the past participle "been."			

Table 4. Negate sentence pattern

		Original+**not**	"Not" condensation
Simple pattern	present past future	Ss or Sp + N Ss or Sp + N Ss or Sp + N + I	Ss or Sp + I*d Ss or Sp + I*d Ss or Sp + I
Continuous pattern	present past future	Ss or Sp + N + PP Ss or Sp + N + PP Ss or Sp + N + If + PP	Ss or Sp + PP Ss or Sp + PP Ss or Sp + If + PP
Proceed pattern	present past future	Ss or Sp + N + Pg Ss or Sp + N + Pg Ss or Sp + N + Ix + Pg	Ss or Sp + Pg Ss or Sp + Pg Ss or Sp + Ix + Pg
Perfective continuous pattern	present past	Ss or Sp + N + ppf + Pg Ss or Sp + N + ppg +Pg	Ss or Sp + ppf + Pg Ss or Sp + ppf + Pg
Comments: There must be a label "N" in the negative sentence pattern. N: connects the word "not" to preceding auxiliaries.			

questions based on the learning ontology and learning corpus.

Question and Answer System

In chat room systems, learners can ask questions of each other or direct questions to the system. In this system, the domain knowledge that is in the Distance Learning Ontology and Learning Corpus can provide answers for users. When users query the system, the system will attempt to find answers from the Knowledge Ontology or Learner Corpus. Also, if a sufficient number of QA pairs have been accumulated, the FAQ can act as a powerful learning tool for learners. Based on these corpora, instructors can revise or enhance their teaching materials. Learners can also learn from the experience of previous learners and other learners.

As discussed in a previous section, we used the knowledge ontology based approach, the **Semantic Relation of Knowledge Ontology**, to construct the Semantic Agent. This methodology can detect sentence patterns and find the positions of the keywords in the Knowledge Ontology. This is a new way to design a Question and Answer system (QA system). The workflow of the QA system is shown in Figure 9.

When the QA system receives a question pattern sentence, it can find the IDs of keywords in the Data Structure Ontology and find their related information (for example, "description" and "algorithm") and then try to answer the learner's question.

The question sentence analysis process is illustrated in Tables 5 and 6.

Some examples of such sentences are as follows:

- What is Stack?
- Which data structure has the method push?
- Does stack have pop method?

According to the Yes/No question sentence pattern and WH question sentence pattern, when the QA system receives the question "***What is Stack***" from a learner, the sentence will first be recognized as a type of question sentence pattern. Then, the QA system will extract the keyword "stack" to find its ID in the Knowledge Ontology. With its ID and the question sentence pattern type "***What is,***" The system will understand the semantic meaning of this question is to ask the definition of stack. Then, it will try to find the definition or description of "stack" for the user. Then, the system will collect this question and answer into the FAQ database. Some examples of Knowledge Ontology are as follows:

```
- <KeyItem id="3" name="stack">
- <Definition>
  <Description>A stack is a Last In, First Out(LIFO) data structure in which all insertions and deletions are restricted to one end. There are three basic stack operations: push, pop, and stack top.</Description>
  <Symbol name="top">A stack is a linear
```

Figure 9. Workflow of questions and answers system

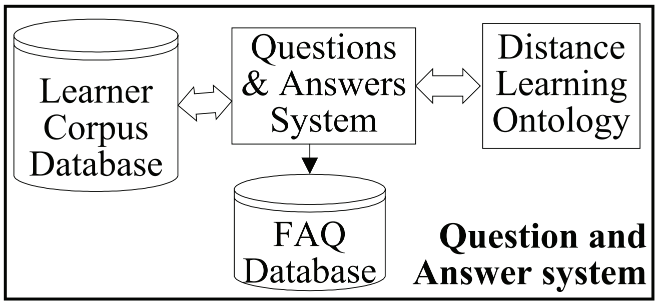

Learner Corpus Database
Questions & Answers System
Distance Learning Ontology
FAQ Database
Question and Answer system

Table 5. Yes/No question sentence

		Yes/No question sentence
Simple pattern	present	Qd + SIs + I*d
	past	Qd + SIs + I*d
	future	Qd + SIs + I
Continuous pattern	present	Qd + SIs + PP
	past	Qd + SIs + PP
	future	Qd + SIs + If + PP
Proceed pattern	present	Qd + SIs + Pg
	past	Qd + SIs + Pg
	future	Qd + SIs + Ix + Pg
Perfective continuous pattern	present	Qd + SIs + ppf + Pg
	past	Qd + SIs + ppf + Pg
Comments: A Yes/no question sentence pattern must be labeled as “Qd”.		

Table 6. WH question sentence

		What	Where and how
Simple pattern	present	Wq + Sid + I*d + Bsw	Wq + Q + SIs + I*d
	past	Wq + Sid + I*d + Bsw	Wq + Q + SIs + I*d
	future	Wq + SIs + I + Bsw	Wq + Q + SIs + I*d
Continuous pattern	present	Wq + SIs + I + Bsw	Wq + Q + SIs + PP
	past	Wq + SIs + I + Bsw	Wq + Q + SIs + PP
	future		
Proceed pattern	present	Wq + SIs + Pg + Bsw	Wq + Q + SIs + PP
	past	Wq + SIs + Pg + Bsw	Wq + Q + SIs + PP
	future		
Perfective continuous pattern	present past	None	none
Comments: A WH question sentence, such as where/when/why questions, must be labeled as “Wq.” Sid: connects subject nouns to finite verbs in cases of subject-verb inversion Bsw: connects auxiliary verb will□have/has/had to past participle.			

list in which all additions and deletions are restricted to one end, which is called the top.</Symbol>

<Symbol name=”overflow”>**There is one potential problem with the push operation: We must resure that there is room for the new item. If there is not enough room, then the stack is in an overflow state.**</Symbol>

<Symbol name=”underflow”>**When the last item in the stack is deleted, the stack must be set to its empty state. If pop is called when the stack is empty, then it is in an underflow state.**</Symbol>

</Definition>

There are some question templates for the question and answer system:

- What is
- The relations of
- Is … has …
- Which … has

Moreover, the FAQ database can also use the data mining technology to accumulate question and answer pairs received from learners while learners are engaged in discussions using this proposed system. If a sufficient number of QA pairs have been accumulated, the FAQ system will obtain the statistics of questions and answers and then get the most frequently Question and Answer pairs. The system can also show these QA pairs to learners. This can be a powerful learning assistant for online learners.

EVALUATION

Several related systems are surveyed in the following:

- IwiLL(Wible, Kuo, Chien, & Taso, 2001), which is a Web-based language-learning platform, consists of several tightly interwoven components. This is a learner corpus for students in Taiwan. Students can write article on the Web and then send then to teachers. The teachers also can check the homework online. Teachers cannot only check for spelling errors and grammar mistakes but also semantic problems. IwiLL also provides some multimedia data to help students learn English. And it can store articles written by students in the learner corpus. However, it is not an automatic system. Teachers need to go online frequently.
- The American National Corpus (Ide, 2003; Ide, et al.. 2002; American National Corpus, n.d.; British National Corpus, n.d.) includes, ideally, a balanced representation of texts and transcriptions of spoken data. Samples of other major languages of North America, especially Spanish and French Canadian, should also comprise a portion of the corpus and, ideally, be aligned with parallel translations in English. ANC comprises approximately 100 million words that include additional texts in a wide range of styles from various domains. It is designed to be a word corpus.
- The British National Corpus, which is a 100-million-word collection of samples of written and spoken language from a wide range of sources, is designed to represent a wide cross-section of current British English, both spoken and written. BNC is an online service that is different from ANC. It is also designed to be a word corpus only.
- CRITIQUE is a system provided by Yael Ravin (1988). The methodology employed in this system is grammar error detection. The system analyzes the grammar and style weaknesses, as the terms "error" and "weakness" suggest. CRITIQUE can detect five categories of grammar errors and eight categories of style weaknesses. This system uses rule-based methodology. But it is only an application; users cannot use this system on the Web. The system can record information from users, but its functions are not upgradeable.
- The Microsoft Office Word grammar checker is a well-known system. When one uses the Word system, there is a "paper clip" agent or something else to provides help. If one makes a spelling or grammar mistake, the agent

will show the error and can correct the mistake. The system uses several dictionaries like ActiveGrammarDictionary, ActiveHyphenationDictionary, ActiveSpellingDictionary, and ActiveThesaurusDictionary to return corresponding Dictionary objects. The system also uses the statistical methodology to detect and correct errors (Microsoft MSDN). But this system cannot collect user information when people use the system. It will also be difficult to upgrade its capability unless upgrade the Word system to next version.

The Table 7 shows the functionalities comparisons between our system and other systems.

IwiLL is also a learning system, whose goal and functionality are very similar to those of our proposed system. We first compare the IwiLL system with ours based on the functions listed in the table. Next, we compare the capability of our corpus with that of ANC and BNC. Lastly, we compare the about the grammar checking capabilities of our system with those of CRITIQUE and the MS Grammar checker. If there is no value in the comparison grid, then means that this function in here can't be compared between our system and the corresponding system.

CONCLUSION

In our proposed system, learners can send messages to each other in an English environment. They can discuss courses with each other and ask teachers questions. We have designed the Learning_Angel Agent and Semantic Agent to be online supervisors. These two agents automatically can help learners to practice English conversation and engage in discussions. The Learning_Angel Agent automatically can detect syntax errors. Then, the Semantic Agent can check the semantics of sentences in dialogues if learners fall behind in discussing courses. Thus, online teachers and tutors do not always have to wait for students to submit questions. In other words, this system can solve the Instructor-off problems.

The Link Grammar, a word-based parsing mechanism, is designed to be an accurate grammar checker. However, it fails to focus on fault tolerance. Different from the Link Grammar, our system is particularly useful for non-native English speakers. In addition it can parse sentences, make comments, and suggest corrections to users. The original Link Grammar does not have these functions, and it appears that the idea proposed herein can be applied in other domain specific applications.

With the system constantly serviced online, it is important for philologists to analyze sentences accumulated from students' dialogues. Then, the system can easily point out common or special mistakes. Subsequently, online teachers can refine their learning materials.

In conclusion, this system provides a better and more interactive environment for teachers and students. Words are the basic communication units in natural language texts and speech-processing activities. When teaching English, teachers always want to know the types of mistakes that their students make. The proposed system also can be extended to encompass more scalable domain. The concepts presented herein can aid in the development of other similar applications. In the future, we will focus on finding better approaches to semantic analysis by evaluating the accuracy of the proposed Semantic Agent and trying

Table 7. Evaluation of our system and other systems

Comparison	**Learning Assistance**		**Corpus Capability**		**Grammar**	
System	**Semantic Chat Room**	**IwiLL**	**ANC**	**BNC**	**CRITIQUE**	**MS Grammar checker**
Corpus	Lerner Corpus	Lerner Corpus	Standard Corpus	Standard Corpus		
Words Capability	Words are updatable	Words are updatable	1,00 Millions Words	1,00 Millions Words		
Grammar check	Automatic	Manual			Automatic	Automatic
Number Disagreement	Y	⊙			Y	Y
Wrong Pronoun	Y	⊙			Y	Y
Wrong Verb Form	Y	⊙			Y	Y
Wrong Article	Y	⊙			N	Y
Punctuation	Y	⊙			Y	Y
Web Application	Y	Y	N	Y		
Support Multimedia	Y	Y				
Semantic Analysis	Y	N				
FAQ collection	Automatic	N				
Online teachers supervising	Not always	Always				

⊙ *Manually checked by teacher*

to make our system adaptable with famous distance-learning standards.

REFERENCES

Adhvaryu, S., & Balbin, I. (1998). How useful is multimedia on the WWW for enhancing teaching and learning? *Proceedings of International Conference on Multimedia Computing and Systems (ICMCS'98)*, Austin, TX, June 28-July 31.

American National Corpus. (n.d.). http://americannationalcorpus.org/

British National Corpus (n.d.). http://www.natcorp.ox.ac.uk/

Goldberg, M. W. (1996). Student participation and progress tracking for Web-based courses using WebCT. *Proceedings of the Second International N.A. Web conference*, Fredericton, NB, Canada, October 5-8.

Goldberg, M. W., & Salari, S. (1997). An update on WebCT (World-Wide-Web Course Tools) – A tool for the creation of sophisticated Web-Based learning environments. *Proceedings of NAUWeb'97 – Current Practices in Web-Based Course Development*, Flagstaff, AZ, June 12-15.

Goldberg, M. W., Salari, S., & Swoboda, P. (1996). *World Wide Web course tool: An environment for building WWW-based courses.* Computer Networks and ISDN Systems.

Harris, D., Cordero, C., & Hsieh, J. (1996). High-speed network for delivery of education-on-demand. *Proceeding of Multimedia Computing and Networking Conference (MCN'96)*, San Jose, CA, January 29-31.

Ide, N. (2003). The American National Corpus: Everything you always wanted to know…and weren't afraid to ask. Invited keynote, Corpus Linguistics 2003, Lancaster, UK.

Ide, N., Reppen, R., & Suderman, K. The American National Corpus: More than Web can provide proceedings of the third Language Resources and Evaluation Conference (LREC). Las Palmas, Canary Islands, Spain, 839-844.

Labrou, Y. & Finin, T. (1999). Yahoo! as an ontology—Using Yahoo! categories to describe documents. *Proceedings of the 8th International Conference on Information and Knowledge Management*, Kansas City, MO, November, 180-187.

Li, J., Zhang, L., & Yu, Y. (2001). Learning to generate semantic annotation for domain specific sentences. *In the Workshop on Knowledge Markup and Semantic Annotation at the 1st International Conference on Knowledge Capture (K-CAP 2001)*, October, Victoria, B.C., Canada.

McGuinness, D.L. & Harmelen, F.V. (2004). OWL Web ontology language overview. W3C Recommendation, February 10.

Microsoft MSDN online Library. http://msdn.microsoft.com/vstudio/

Ravin, Y. (1988). Grammar errors and style weaknesses in a text-critiquing system. *IEEE Transactions on Professional Communication, 31*(3), 1988, 108-115.

Sleator, D. K. & Temperley, D. (1991). Parsing English with a link grammar. October, CMU-CS-91-196.

Tsang, H.W., Hung, L.M., & Ng, S.C. (1999). A multimedia distance learning system on the Internet. *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, *2*, 243-246.

Wang,Y.H., Wang, W.N., & Lin, C.H. An intelligent semantic agent for e-learning message communication. *Journal of Information Science and Engineering, 21*(5), 1031-1051.

Wang, Y.H. & Lin, C.H. (2004). A multimedia database supports English distance learning. *An International Journal Information Sciences, 158*, 189-208.

Wang,Y.H., Lin, C.H., & Wang, W.N. (2007). Semantic enhanced QA system architecture use link grammar parser for e-learning environment. *Proceeding of Conference Taiwan E-Learning Forum (TWELF) 2007*, May, 18-19.

Weibel, S., Gridby, J., & Miler, E. (1995). *OCLC/NCSA metadata workshop report,* Dublin, EUA. Retrieved from http://www.oclc.org:5046/oclc/research/conferences/metadata/dublin_core_report.html

Weible, D., Kuo, C.-H., Chien, F.-Y., & Taso, N.L. (2001). Automating repeated exposure to target vocabulary for second language learners, Advanced Learning Technologies. *Proceedings, of the IEEE International Conference on,* August 6-8, 127-128.

Willis, B. (n.d.). Distance education at a glance. Engineering Outreach at the University of Idaho, http://www.uidaho.edu/evo/distglan.html

Ying-Hong Wang is an associate professor in Departmant of Information Engineering of TamKang University since 1996. He received the BS degree in information engineering from Chung-Yuan University, and the MS and PhD degrees in information engineering from the TamKang University, in 1992 and 1996, respectively. From 1988 to 1990, he worked in the Product Development Division of Institute of Information Industry (III). From 1992 to 1996, he was a lecturer in the department of information engineering of TamKang University. Beginning fall 1996, he is an associate professor in the department of information engineering of TamKang University. His current research interests are wireless communication, mobile agent technologies, software engineering, and multimedia database system. He has nearly 100 technological papers published on international journals and International conference proceedings, he also join many International activities been program committee, workshop chair, session chair and so on. He had been invited as visiting researcher in The University of Aizu, Japan, from Jan to March 2002. He is the chair of Department of Computer Science and Information Engineering, Tamkang University.

Chih-Hao Lin is an assistant professor at the Department of Information Science and Application, Asia University, Taichung Taiwan. He received his MS and PhD degrees in computer science and information engineering of TamKang University in 2000 and 2004. Dr. Lin also has been on many program committee for international activities. His research interests focus on distance learning, natural language understanding and personal health record.