An AI Framework to Teach English as a Foreign Language: CSIEC

Jiyou Jia

■ *Computer Simulation in Educational Communication (CSIEC) is not only an intelligent web-based human-computer dialogue system with natural language for English instruction but also a learning assessment system for learners and teachers. Its multiple functions—including grammar-based gap-filling exercises, scenario show, free chatting, and chatting on a given topic—can satisfy the various requirements for students with different backgrounds and learning abilities. After a brief explanation of the conception of the dialogue system, as well as a survey of related works, I will illustrate the system structure and describe its pedagogical functions with the underlying AI techniques, such as natural language processing and rule-based reasoning, in detail. I will summarize the free Internet usage within a six-month period and its integration into English classes in universities and middle schools. The evaluation findings about the class integration show that the chatting function has been improved and frequently utilized by users, and the application of the CSIEC system on English instruction can motivate learners to practice English and enhance their learning process. Finally, I will conclude with potential improvements.*

As an international language, English is a key instrument in the development and cultivation of cross-cultural communication among students. English learning is gaining more and more attention in developing countries like China and has long been listed as obligatory in school and higher education. However, some problems exist in the English language education in these countries. First, as noted by situated learning (Brown, Collins, and Duguid 1989) and constructivist learning theory (Jonassen 1994, Von Glasersfeld 1996), one of the best ways to learn a foreign language is through frequent communication with a native speaker. Such communication, however, is not practical in the classroom because of the need for a one-to-one student-teacher ratio. A number of other factors, ranging from a lack of time to shyness or a limited opportunity for quality feedback, also hamper the use of the target language (Fryer and Carpenter 2006).

A potential solution to this problem is to apply computer spoken dialogue systems to role playing as conversational partners. If researchers could design an interactive web-based system that could chat with English learners anytime anywhere, the great demand for learning partners might be satisfied. In 2002, motivated by this demand, I began designing a system that could communicate with Internet users in English (Jia 2004a), which I called Computer Simulation in Educational Communication (CSIEC). The design principle was based on application and evaluation. As soon as was practical, I put the system into free use on the Internet[1] and obtained user feedback. I also cooperated with English teachers from universities and middle schools and integrated the system into English instruction. Through systematic application and evaluation, I continually receive suggestions and criticism that effectively direct the research.

Related Work

Brennan (2006) defined a chatbot as "an artificial construct that is designed to converse with human beings using natural language as input and output." A chatbot architecture integrates a language model and computational algorithms to emulate communication between a human user and a computer using natural language (Abu Shawar and Atwell 2007).

ELIZA (Weizenbaum 1966) was the first chatbot. It used key words to analyze input sentences and created its response based on reassembly rules associated with a decomposition of the input. But ELIZA held no memory of the conversation. Nevertheless, the syntactic way of natural language processing exemplified by ELIZA has been developed significantly and continually since ELIZA's inception, leading to the development of various chatbots such as ALICEBOT.[2] With the improvement of natural language processing since the 1990s, chatbots have become more practical and are now being used in education.

Graesser et al. (2005), for example, used AutoTutor, an intelligent tutoring system with mixed-initiative dialogue that can simulate a human tutor by holding a conversation with the learner in natural language to enhance the learner's engagement and the depth of the learning.

Kerfoot et al. (2006) described an experimental use of chatbots as a teaching adjuvant in training medical students. The experiment showed web-based teaching using chatbots significantly increased test scores in the four topics at each medical school and learning efficiency was increased threefold.

Seneff (2006) described several multilingual dialogue systems specifically designed to address the need for language learning and teaching. Several different domains were developed in which a student's conversational interaction was assisted by a software agent functioning as a tutor, providing them with translation assistance.

Abu Shawar and Atwell (2007) developed algorithms for adapting or retraining a chatbot to a training corpus. They stated that the evaluation feedback from language learners and teachers indicated that these adaptive chatbots offered a useful autonomous alternative to traditional classroom-based conversation practice.

Finally, Kerly, Hall, and Bull (2007) discussed the development and capabilities of both conversational agents and open learner modeling. They described an experiment that investigated the feasibility of using a chatbot to support negotiation. The experiment result showed most students liked the chatbot and that it helped them understand their learner model.

From these related works, I conclude that usage of chatbot systems in education is coming to the attention of more and more researchers in related fields. This trend confirms my determination to further the development of the CSIEC system and its application in English education.

Current System Architecture

Contrary to the partial parsing used in ELIZA, in CSIEC I attempted full syntactical and semantic analysis of user input, following the advice of logician G. Frege (1879) who pointed out that "the meaning of a sentence exists in the meanings of all words in the sentence and their conjunction method." After parsing user input text, I obtain user information in the form of extensible markup language (XML) (specifically, natural language markup language [NLML]), calling this information "user facts." These facts are retrieved from natural language expressions and also represented with the annotation of natural language in the sentence ontology. The facts function as the main contextual source of the robot dialogue reasoning. This thought originates from L. Wittgenstein's theory (1918–1921) about the world, facts, objects, and human language:

> The world consists of facts, the facts consist of objects. The facts are reflected in the language. A logical picture of facts is a thought. The boundary of language is the boundary of knowledge and cognition.

The current CSIEC system consists primarily of the components illustrated in figure 1, where a plain box represents a module, a box with four slots represents a variable, a cylinder represents a database, and an arrow represents the data flow. I introduce them in the sequence of data flow. (1) The HTTP request parser resolves the user request from the HTTP connection and obtains parameter values: input text, scenario topic, agent character, speech speed, spelling and grammar checker, and so on. Next, (2) the English parser parses the user input text into natural language markup language. NLML (Jia 2004b) is a dependency tree in XML form and structurally labels the grammar elements (phrases), their relations, and other linguistic information in English sentences (words, part of speech, entity type, and so on). For example the NLML of the sentence "*I come*" is shown in figure 2. (3) The NLML parser parses the NLML of the user input into a natural language object model in Java (NLOMJ), which represents the grammatical elements and their dependency with the sentence ontology in the working memory (Jia, Ye, and Mainzer 2004). Through NLOMJ, the declarative sentence can be retrieved and decomposed into atomic facts consisting of only one subject and one verb phrase. (4) The natural language database (NLDB) stores the historical discourse, the user atomic facts in the form NLML, the robot atomic

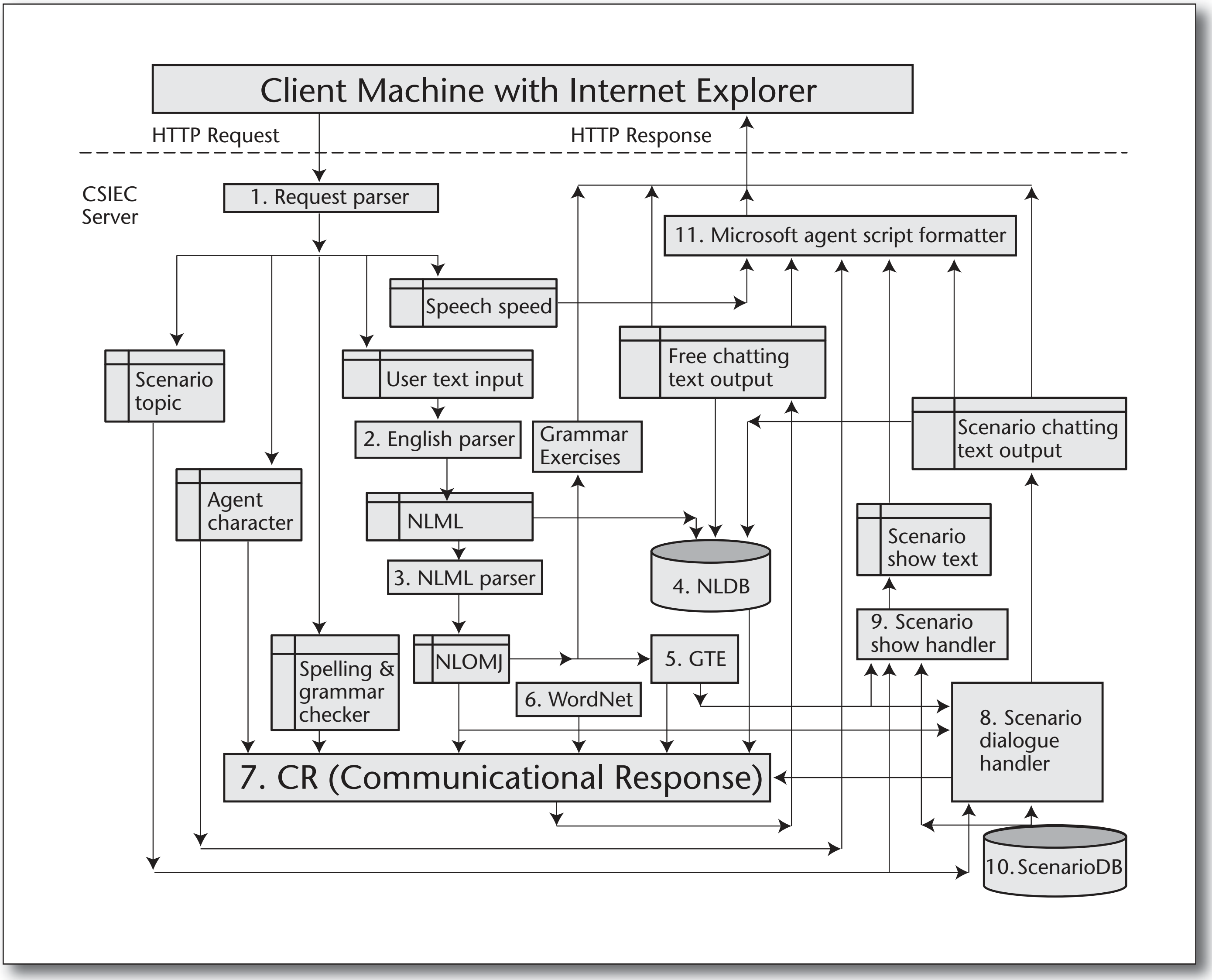
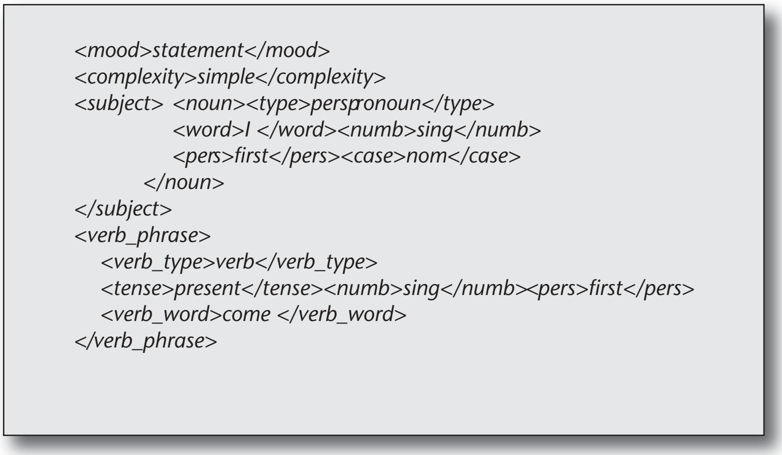


Figure 1. The Current Architecture of the CSIEC System.

facts, which are also expressed in NLML, and other data. Next, (5) generation of textual entailment (GTE) is a generation mechanism of textual entailment or inference and is a supplemental source of communicative response (Jia 2008). Next, (6) the world model contains commonsense knowledge that is the basis for response generation and logical inference. It is now represented by WordNet (Fellbaum 1998).

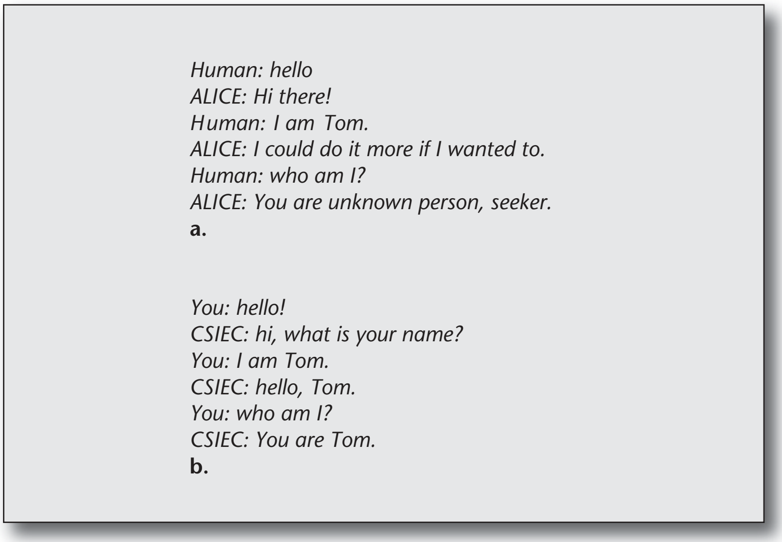
In the seventh element of data flow, (7) the communication response (CR) mechanism comprehensively takes into account the user input, the user facts stored in NLDB, the world model, the personality of the user expressed in the previous dialogue, that of the robot itself selected by the user, and GTE. The robot response is generated by this special mechanism. I illustrate it with the comparison of one human-computer dialogue piece with the ELIZA-like ALICEBOT and another one with my system. The dialogue with ALICEBOT is depicted in figure 3a. The dialogue with CSIEC is depicted in figure 3b. Considering the dialogue context or not makes the two dialogues different. More details about the CR mechanism will be introduced in next section.

In the eighth element of data flow, (8) the scenario dialogue handler creates the robot output corresponding to the user input within a given scenario. (9) The scenario show handler creates the random robot-robot scenario show scripts within a given scenario. In the tenth element, (10) the scenario database stores the robot-robot scenario show scripts and human-robot dialogue scripts, which are manually written by a designer, for example an English language teacher. Finally, (11) the Microsoft agent script formatting transforms



```
<mood>statement</mood>
<complexity>simple</complexity>
<subject> <noun><type>perspronoun</type>
              <word>I </word><numb>sing</numb>
              <pers>first</pers><case>nom</case>
          </noun>
</subject>
<verb_phrase>
   <verb_type>verb</verb_type>
   <tense>present</tense><numb>sing</numb><pers>first</pers>
   <verb_word>come </verb_word>
</verb_phrase>
```

Figure 2. NLML of the Sentence "I Come."



```
Human: hello
ALICE: Hi there!
Human: I am Tom.
ALICE: I could do it more if I wanted to.
Human: who am I?
ALICE: You are unknown person, seeker.
a.

You: hello!
CSIEC: hi, what is your name?
You: I am Tom.
CSIEC: hello, Tom.
You: who am I?
CSIEC: You are Tom.
b.
```

Figure 3. Dialogue with ALICEBOT and CSIEC.

a. Dialogue with ALICEBOT. b. Dialogue with CSIEC.

the output text into Visual Basic script, considering the selected agent character and speaking speed.

The system is implemented in JDK1.6 and uses MySQL as the database management system.

Basic Functions and Underlying AI Technologies

Among the various functions of the CSIEC system, I will introduce, first and foremost, its specific chatting function and the underlying technologies.

Multimodal User Interface and Selectable Chatting Pattern

As with other conversations between two humans, Internet users have different preferences for dialogue simulation. In order to adapt to various user preferences, CSIEC provides several user interfaces and dialogue patterns. At first, users can chat with the robot either through text or speech. They can hear synthesized voice and watch the avatar performance through Microsoft agent technology. They can speak to the robot through a microphone that is equipped with a speech-recognition program like IBM ViaVoice. Next, the robot can check the spelling and grammar of the input upon the user's request.

Finally, the topic of chatting between the user and the robot can be either free (unlimited) or given (limited). The unlimited dialogue simulation doesn't specify the dialogue topic and content. It benefits users whose English is fluent or who are at least good at written English, as well as users who are extroverted or talkative. However, users whose English is poor or who are introverted often have little to chat about with the virtual chatting partner. For them, an instructive dialogue guided by the agent is more helpful. Language teachers also acknowledge that conversation practice normally centers on a specific topic during the learning of topic-specific vocabulary and language. It is noticeable that in normal human conversation these two chatting patterns are not mutually exclusive but are often interwoven. In the system design, I incorporated this interaction, too.

Free Chatting Adaptive to User Preference and Topic

In free chatting, users may choose different types of chatting patterns based on their preferences. For the sake of dialogue personalization, I have designed five Microsoft agent characters that represent different kinds of chatting patterns. Christine always tells the user stories, jokes, and world news. Stephan prefers to listen quietly when the users share with him their own experiences. Emina is a curious girl and is fond of asking users all kinds of questions related with the users' input. Christopher provides comments, suggestions, and advice based on the user's input. Ingrid behaves as a comprehensive virtual chatting partner, who gives users responses considering both the input text and the discourse context.

Once a user has registered with the chatting system, the user's profile is obtained and recorded (including the gender, birthday, educational level, and address), so that the corresponding chatting topic and content can be generated based on personal information. Of course, if the user decides to change the chatting topic while the robot is narrating comments or asking questions, the robot should terminate this process and change to the topic given by the user or by the robot itself. If the user specifies a topic, for example, "I want to talk about sports," the robot changes to that topic. If

```
Parse the input text into NLML and get its NLOMJ
Pattern recognition: compare the structure of this NLOMJ with the patterns in direct response table
If there  is such a pattern matching this input text
      transform the template NLML and calculate the response text
end
```

Figure 4. Algorithm for Direct Response.

the user only expresses a wish to change the topic but does not determine a topic, such as "I want to talk about something else," the robot selects a topic from the topics waiting list, which I will discuss later on in this article.

The user's interests are also expressed in the input, that is, through nouns and verbs already mentioned. Consequently, nouns and verbs can trigger the chatting topic. More frequently one noun or several related nouns are talked about, and the related topic is more emphasized. Thus the chatting between the user and robot can be regarded as guided chatting or chatting on topic.

I deal with chatting on a given topic in two ways. One is by predefining some comments or asking some questions about this topic. By talking about it only one statement or question will be randomly selected and given out. Another way is to search the topic in the guided chatting within a given scenario and then transfer the chatting to the guided chatting in a given scenario (more about this later). The arrow from the scenario dialogue handler to the communicational response in figure 1 indicates this relation.

In sum, the goal of free chatting is to elicit conversation from students. To accomplish this purpose, the robot tries to adapt itself to the user's interests and to start new topics.

Communication Response

For user input, the robot looks for a response in the following sequence: personality knowledge considering the discourse context, direct response, inference, and common sense. In this subsection I introduce the direct response at first, which is just a pure response to the input text ignoring the dialogue context. Then I introduce the complicated response considering the personality knowledge and the discourse context.

Direct Response

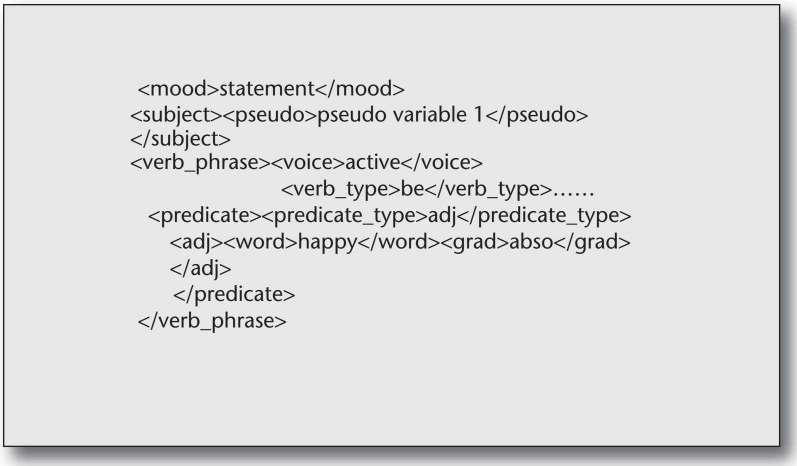
The key words or pattern-matching mechanism used in ELIZA (Weizenbaum 1966), ALICEBOT,[2] and other similar chatbot programs requires an exact description of both input pattern and template output with specific key words. However, it is very laborious to write down all these pairs.

With its syntactical and semantic analysis capability, the CSIEC system can define a more general pattern that includes all those input sentences with a specific syntactic or semantic feature, a more general template that includes all outputs, and a mechanism that transforms the output template into an appropriate output. The pattern and template are described with the form NLML, including some pseudovariables, which stand for variant grammar elements in the input. The pairs of an input pattern and an output template are written into a table "direct-response" in NLDB. The algorithm for the direct response is described in figure 4. As the figure shows, the direct response mechanism consists of three procedures: (1) input pattern and output template annotation with NLML, (2) pattern recognition (matching), and (3) output template transformation.

I illustrate the mechanism with an example of how to respond to the inputs such as: *I am happy*, *my sister was very happy yesterday*, and so on, which are input by the user for the first time. These inputs can be described as: *Somebody be happy*. A response to them can be generalized as a question: *Why be somebody happy?* just like: *Why are you happy? Why was your sister very happy yesterday?*

The traditional key words mechanism used in ELIZA and ALICEBOT has difficulties describing this pair of input patterns and output templates both completely and exactly, whereas the direct response algorithm can do it. The input pattern can be expressed with the NLML format shown in figure 5. The response is shown in figure 6. Except for these two input and output examples, the input text in figure 7 can be found matching the input pattern and can get the corresponding responses with the help of the output template.

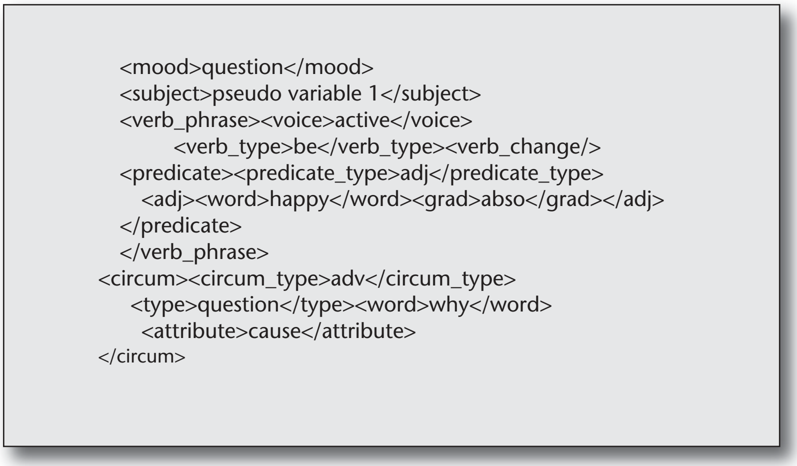
There can be multiple responses to a given input text, so several output templates can be generated for an input pattern. The pairs of pattern and template are indexed so that the robot can generate different responses for a given text of a



```
 <mood>statement</mood>
<subject><pseudo>pseudo variable 1</pseudo>
</subject>
<verb_phrase><voice>active</voice>
                    <verb_type>be</verb_type>……
  <predicate><predicate_type>adj</predicate_type>
     <adj><word>happy</word><grad>abso</grad>
     </adj>
      </predicate>
 </verb_phrase>
```

Figure 5. Input Pattern Expressed with the NLML Format.

Due to the space limitation only the core parts are listed, while the other parts are represented by ellipsis. The rest of the NLMLs in this article are also described as such.



```
   <mood>question</mood>
   <subject>pseudo variable 1</subject>
   <verb_phrase><voice>active</voice>
          <verb_type>be</verb_type><verb_change/>
   <predicate><predicate_type>adj</predicate_type>
      <adj><word>happy</word><grad>abso</grad></adj>
   </predicate>
   </verb_phrase>
<circum><circum_type>adv</circum_type>
     <type>question</type><word>why</word>
       <attribute>cause</attribute>
</circum>
```

Figure 6. Summary of Response.

user to avoid sole and tedious output.

Direct response generation only considers input text, but not dialogue context. User input can be a declarative, investigative, imperative, or an exclamatory sentence or phrase. I have designed a GUI pattern-template editor to assist with swift and convenient annotation. Consequently, the pattern and response annotation in NLML must be completed by an author who is fluent in English grammar and dialogue generation but not necessarily knowledgeable of the complicated description of text with NLML.

Direct Response Taking into Account Dialogue Context and Personality Knowledge

If a user talks about the personality of the user or of the robot or repeats an input, the user facts and robot facts along with the user discourse history must be taken into consideration. Searching the user personality or robot personality tables, taking into account the dialogue history, the robot generates the response.

For example, if the user inputs *I am happy* for the second time, the same response *why are you happy?* may sound repetitive. However, from the user dialogue history table the robot knows that this response template has been used once, so it will look for another response template, such as *enjoying life will make you happier.* Moreover, the robot finds the same fact in the user facts table (personality knowledge), which was stored by the last parsing of the declarative sentence; thus, it will respond with this knowledge: *Yes, I know. I am very happy too to hear that you are happy.*

Searching in Inference Knowledge and Commonsense Knowledge Database

If, for a given input, no response can be generated based on the direct response or on personality facts, the inference knowledge (GTE) table and then the commonsense knowledge table will be searched. The GTE algorithm simulates the process that a human being who is learning English as a foreign language goes through and consists of rule annotation in NLML, pattern recognition, and entailment transformation, too. Additional details about the GTE mechanism can be found in Jia (2008). If there is still no corresponding response, WordNet will be searched to retrieve the definition of a noun in the input (or its symposium, hypernym, or hyponym) as the response output.

Communicational Response for Different Moods: Cases

I illustrate the response generation algorithm with some user input and robot output pairing examples. For the sake of clarification I write the input and output pair at first, and then the technical justification.

Statements. If the input is a statement, the response to this statement can be a statement, or a question, or their combination. For example:

> "Hello, I am Tom." → "Hi, Tom. How are you?"

This output is a direct response to "I am <name/>." The user's new fact *I am Tom* will be added into the user fact table. By this user's next login, the robot will greet with: "Hello, nice to meet you, Tom."

> "I am a student." → "How do you like being a student?"

This output is a direct response to the user input, and associated with the user input.

> "I feel very happy to be a student." → "I'm a college student and my major is English. What is your major?"

The robot output is still another direct response to

"My classmate will be very happy tomorrow"→ "Why will your classmate be very happy tomorrow?"
"I am not happy today"→ "Why are you not happy today?"
"My teacher has been happy for a long time"→ "Why has your teacher been happy for a long time?"

Figure 7. Input Text Matching the Input Pattern.

be a student, but consists of two sentences. One is a statement about the robot fact, and another is a question about the user.

> "Mathematics." → "alas. You study mathematics. Do you like it?"

Combining the user answer with its original question, the robot formalizes a new user fact: your major is mathematics. Based on this fact and the inference knowledge, that is, *somebody's major is mathematics Somebody studies mathematics,* it gets the output *you study mathematics* as a fact, and adds one direct response (question) to the new fact: *do you like it?*

> "Surely." → "okay, you like mathematics."

The robot analyzes the user input and obtains a new user fact according to the dialogue context, and repeats this new user fact.

> "You are boring." → "No, I am not boring. Are you boring?"

This input is a comment on the robot, and the robot can't find such a statement *I am boring* from its own facts table, so it denies the user comment. Moreover, the robot can't find such a fact from the user facts table, thus it asks the user again.

Questions. If the input is a question, it tries to answer the question.

> "Do you like mathematics?" → "Certainly, I like mathematics, because mathematics makes me clever. I know you also like mathematics, because you have told me."

The user asks a question about the robot itself, so the robot looks for the fact *I like mathematics* from its own facts table. It finds not only such a fact, but also the explanation to this fact. In addition, it finds also such a fact from the user facts table; thus it repeats this fact to demonstrate its memory capability.

> "What is a plane?" → "A plane is an aircraft that has a fixed wing and is powered by propellers or jets."

The user asks a concept definition, and the robot finds the definition from WordNet.

> "Who am I?" → "Tom."

The user asks a question about him or herself, and the robot finds the answer from the user facts table.

> "Do you speak Japanese?" → "No, I do not speak Japanese because I can't speak Japanese. Do you speak Japanese?"

The user asks a question about the robot itself, so the robot looks for the fact *I speak Japanese* in its own fact table. It finds not only a negative answer to this fact, but also its explanation. However, it cannot find such a fact from the user facts table, thus it asks the user about it.

> "What is eighty-three plus ninety-nine?" → "182."

The robot can calculate a mathematic expression with the direct response.

> "Where is Germany?" → "Sorry I do not know. Do you know the answer? Where is Germany?"

For a question whose answer can't be retrieved from the personality knowledge and commonsense knowledge, the robot acknowledges its lack of knowledge and asks the user for the answer.

> "Germany is in western Europe." → "Oh, Germany is there."

The robot adds the fact uttered by the user into the commonsense knowledge table, as it has nothing to do with the user and the robot, and responds to the user input.

> "Where is Germany?" → "In western Europe."

Now the robot finds the answer to the question from the updated commonsense knowledge table.

Imperitive. If the input is an imperative, it responds and acts.

> "Please tell me what a plane is." → "A plane is an aircraft that has a fixed wing and is powered by propellers or jets."

The robot reduces the user request into a question about the concept definition and gets the answer from WordNet.

> "Tell me a joke." → "This story is a joke about Baby Drink. How do you make a baby drink? "

The robot gets a joke from the table "story" and narrates it sentence by sentence.

> "Sing a love song please." → "There comes a time when we heed a certain call ...."

The robot retrieves a song from the table "song" and recites it.

Exclamation. If the input is an exclamation, the robot responds as it feels the surprise.

> "How simple you are!" → "You are surprised, aren't you? Yeah, I am simple. Are you simple?"

The robot transforms the user exclamation into a

```
1. (Can I help you?)<coat<hyponym><n>:coat>
1. (May I help you?)<coat<hyponym><n>:coat>
2.<1,coat<n>:coat>(What color do you like?)<color <hyponym ><n>:white, any>
3.<1, coat<n>:>(We have only coats. Good bye.) <terminator>
4.<2,color<n>:white>(How about this one?)  <price:price>
5.<2,color<n>:any>(How about this white one?)  <price:price>
6.<2,color<n>:>(We have only white ones. Bye bye.)<terminator>
7.<price:price>(It costs only twenty Yuan.) <response <j>:cheap,  expensive>
8.<price:> (But it is very cheap. It costs twenty Yuan.)   <response<j>:cheap, expensive>
9. <response<j>:cheap>(Thanks.)
10.<response<j>:expensive>(Sorry. That's the best we can do.)
11. <response<j>:>(Thanks. Bye bye!)
```

Figure 8. Salesman and Customer Example Script.

declaration about itself, *I am simple*, and gets this entry also from its robot facts table; therefore it acknowledges it and asks the user the same question since it cannot find the same entry from the user facts table.

Guided Chatting in a Given Scenario

The dialogue on a given topic requires not only commonsense knowledge, but also domain knowledge, in addition to the explicit usage of the knowledge with the form of natural language. The dialogue should be developed step by step around a topic for this scenario. It can be illustrated by a decision tree structure with many branches. I use scripts to describe the decision tree in the dialogue on a given topic. The script is made up of lines of dialogue states; every one of them is a branch of the decision tree. For example, if the robot speaks first, then in every line there must be the text output from the robot and its order number in the dialogue. This output may be triggered by a specific user input, which I call the prerequisite of this output text. The robot may also expect the user to input certain texts, or some texts with specific semantic or syntactical characters, which I call the expectation of this output text. I write the line in the script with in the following format:

Nr. <prerequisite> (text) <expectation>.

Nr. and *text* are two necessary components in every line. *Nr.* is an integer indicating the line order in the whole script, whereas *text* can be any text from the robot, either statement, or question, and so on, and it is written within closed brackets.

In a script line, the prerequisite and expectation are optional. However, if they do appear, they must be written within closed diamond brackets. If the prerequisite exists and is also satisfied by the user input, the robot gives the output text. The expectation means the robot hopes that the user responds to this text with some specific syntactic or semantic features that can be applied to the instructional goal. For example if the user's input does not satisfy the robot's expectation, he or she will face the previous robot output again until the expectation is fulfilled. This dialogue pattern can be used for drills. Another alternative is that the user is given a high mark if the input satisfies the robot output; if not, then a low mark is given. The robot continues the dialogue despite all. This pattern can be used in tests or examinations.

The format of the prerequisite is:

<Nr, variable 1: value 1, value 2 ...; variable 2: value 1, value 2 ...>

The format of the expectation is:

<variable 1: value 1, value 2 ...; variable 2: value 1, value 2 ...>

The prerequisite needs an order number indicating the expectation in which line this condition fulfills. There may be more than one value for a given variable. This means if the variable equals any one of the listed values, the condition is fulfilled, that is, the values for a given variable have the relation of logical disjunction. There may be also more than one variable and its corresponding values. The relation among these variables is the logical conjunction.

One example script is about the "*salesman and customer*." The text of the script is depicted in figure 8.

In the prerequisites and expectations of figure 8, there are some new symbols. The symbol <hyponym> indicates that the user's input satisfies the expectation if it contains a hyponym of the values. For example, for the expectation <coat<hyponym> <n>:coat>, the user's input "I want to buy a cutaway" fulfills the expectations because a cutaway is a kind of coat. The symbol <n> indicates that the variable and its values are nouns, and the user's input satisfies the expectation if it contains a synonym of the values. The symbol <j> indicates that the variable and its values are adjectives, and the user's input satisfies the expectation if it contains a synonym of the values. The symbol <d> indicates that the variable and its values are adverbs, and the user's input satisfies the expectation if it contains a synonym of the values. The symbol <terminator> is the last output from the robot and terminates the dialogue for this scenario regardless of the user's response.

In figure 8, the lines from 7 to 11 demand a prerequisite without any line number, while the other lines depend on the preceding ones, one by one. I use key-word detection plus semantic analysis to interpret these scripts. The script interpreter handles the variable values in the prerequisite and expectation as key words (and their synonyms, hyponyms, and so on) in the user input within the framework of syntactical and semantic analysis. Writing this kind of script is difficult for English teachers who want to use this program to train students. Consequently, I have designed a Java GUI, that is, a discourse script editor (DSE), to allow easy, step-by-step editing of the scripts.

Automatic Scoring of Gap-Filling Exercises without Defined Answers

Traditional computer-based gap-filling exercises require a definite answer or a set of definite answers. For questions whose answers are difficult to list, manual checking is unavoidable. However, without predefined answers, this kind of exercise can promote creative thinking in the students.

With its spelling and grammar check function, the CSIEC system can decide whether a filled gap-filling sentence is grammatically correct. Therefore it can be applied to assess the gap-filling exercises, thus lessening the teachers' burden. Currently, the system provides the interface for teachers to design new gap-filling exercises, as well as the interface for learners to do these exercises and receive automatically assessed results. An example of a gap-filling exercise is: *I ( ) a student.* The correct answer to the gap can be: *am*, *want to be*, *will be*, *have been*, *need*, *help*, and so on.

Scenario Show of Two Robots

The scenario show of two robots is designed to aid users with their chats with the robot on a given topic. With it, users can watch the scenario show of two robots before the human-computer interaction. The talking texts are predefined by the teacher for specific contexts or topics. However, actual texts for a given meaning can be expressed randomly, so this kind of scenario is different from monotonous scripts presented in traditional instructional videos or audiocassettes. It will reinforce the learner's spontaneous listening ability. The teachers can readily write the scenario texts with any text editor.

Listening Training

I use Microsoft agent technology to synthesize output text, because the agent's voice is lifelike, and its appearance, movements, and actions can be designed in vivid detail. Moreover, the technology can synchronously display spoken text, which facilitates aural understanding and engages the user. I have also designed a free web page whose agent can read any texts input by the user. The users can adjust the robot's reading speed at any time. Unlike traditional audio technologies such as audio players, users encounter unexpected text and voices generated by the robot, similar to talking with a real human being. Thus, this function will increase users' listening comprehension and prompt response ability.

Scoring Mechanism

To motivate users to learn English, I trace users' usage of different functions and give them certain scores. The underpinning score principle is designed to encourage chatting with agents and spelling and grammar checking. By chatting on a given topic, users are given a high mark if the input satisfies the robot's previous output and a low mark otherwise. This mark also contributes to the total score.

Users can review their performance and scores after entering the system. This function is very important and conducive to self-learning and evaluation. A special user who is labeled as such by the teacher can access the performance and scores of all the users who are classified as his or her students. This automatic monitoring function allows the teacher to assess the students' learning behavior and progress.

Application and Payoff

Internet users come to the CSIEC web page primarily through search engines, since the website has become one of the top five search results in popular search engines such as Google, Yahoo, and Baidu by related keywords such as "English chatbot," "online English learning" in Chinese or in English. Although I haven't made any large-scale advertisements, the CSIEC web page popularity demonstrates the effectiveness and appeal of the system.

With recorded human-computer dialogues, I can summarize the system's chatting function from January 20, 2007, to June 20, 2007. Unique users who accessed CSIEC during this period numbered 1783.

The quality of chatting can be measured by its duration, as defined in my earlier work (Jia 2004c) on the experimental report of using ALICEBOT for English learning. To calculate the chatting duration I define two terms: *round* and *number of rounds*. A round means a user input and a corresponding

Duration of the Dialogues	Range of Round Count	User Count	User Count / Total User Count (percentage)	User Count / Total User Count (percentage) in Jia (2004c)
Short	(0, 10]	871	48.85	62,34
Long	(10, 50]	685	38.42	30,10
Longer	(50, 100]	136	7.63	4,78
Very long	(100, 580]	91	5.10	2,79
	Total user count	1783	100.00	100,00

Table 1. The Relationship between Duration of Dialogues and Number of Users.

robot output. The total rounds of a given user cover all dialogues between the user and the chatbot and can be used to measure the duration of the chatting.

Sometimes the user cannot get any response from the chatbot after a long interval. Possible reasons include many users' simultaneous visits to the server and flaws in the system's design. After scrutinizing the user dialogue records, I found 313 error rounds and 48,840 effective rounds with both input and output. I define the natural error rate as the ratio of the error rounds number divided by the total rounds number. In this study period the natural error rate is 313/(313+48840), that is, 0.64 percent.

The rounds count distribution is listed in table 1. The average rounds count is 27.4. The count of rounds varies from 1 to 580. Compared with the finding in Jia (2004c), the percentage of short chatting with the robot has decreased by 21.64 percent. Proportionally, the percentage of the long and longer chatting has increased.

User Feedback

At the foot of almost every web page of the CSIEC system, there is a feedback text area where users can leave comments, criticism, and suggestions—either in Chinese or English. Through the detailed securitizing and analysis of these texts, I hope to find what problems the users address. Many users still input normal chatting text into this area, such as *hello* and other meaningless texts. After excluding such texts, I found, at the time of this writing, 341 lines of real feedback.

There were 79 positive comments; 37 of them are very simple positive comments such as *(very) good*, *(very) well*. Six praised the system without any reason, for example: *clever*, *I like this*, and *I love you*. Thirty-six expressed positive comments with reasons such as *the robot is more advanced than before, and also personalized. The access speed is faster than before* and *the kind of communication can improve our English*.

Seven comments were very simple negative comments, such as *not good, just so so, simple,* and *stupid*. Others pointed out the problems they met: that they could neither see the agent animation nor hear the agent's speech, the agent character is not good, the agent voice sounds weird, the agent speaks too fast or too slowly (12); they can't understand what the robot is saying, or the meaning of new words (13); there were grammatical errors in the robot's responses (5); the dialogue in a given scenario was too short (10); the robot only asks the user a similar question, but can't answer the user's question or the answer is false (15); the robot talks too little, or often changes the topic (12); or the response from the robot is (too) slow (16).

Ten of the 341 feedback lines were both positive and negative, like *the system is very good, but the robot responds too slowly.*

This feedback highlights either technical problems or content shortcomings, which will be addressed as I further improve the system.

Formative Evaluation of English Class Integration

After some discussion with English teachers about class integration and evaluation of the CSIEC system, I decided that instructional instruments are (1) the scenario show by two chatting robots with the scenario content and (2) students talking with one robot on the teaching topics from the textbook.

In the first term, 86 graduate students from two English classes—taught by the same teacher—participated in the study. The teacher recommended, rather than required, that the students use the system. For 12 teaching units the CSIEC research group designed 25 scenario scripts of both human-robot chatting and scenario show. In the second term, 45 second-year high school students attended the study, where the teacher required the students to use the system together in the computer room. For 10 teaching units we designed 40 scenario scripts.

We collected data from questionnaires completed at the end of each experimental term. Because in high school the content learning in every course unit is stressed, an item of "reviewing key points in the course units" was incorporated into the ques-

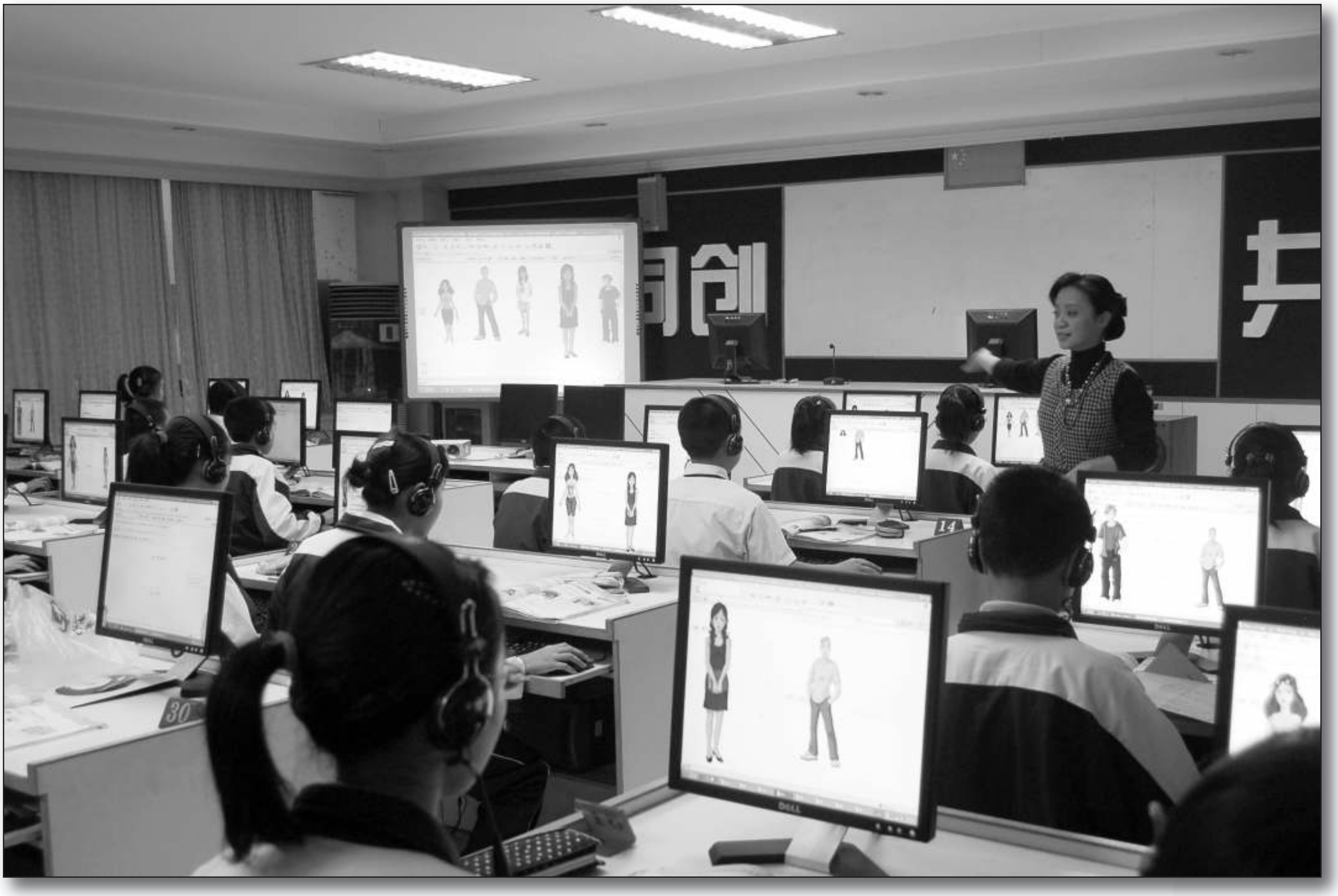


Figure 9. Junior School Students Using CSIEC in the Computer Classroom.

tionnaire. All the items were measured with a five-point Likert agreement scale, that is, 5 indicates the maximum agreement and 1 means no agreement. The Cronbach's Alpha for the five items of the graduate students was 0.933, so reliability of the surveyed items was very good. The Cronbach's Alpha for the six items of the high school students was 0.741. The reliability of the surveyed items here was not as good, but acceptable.

I couldn't recognize a big difference between the attitudes of the graduate students and those of the high school students. I conclude that the students feel CSIEC-based English learning can help with course unit review, make them more confident, improve their listening ability, and enhance the interest in language learning. Another item in the questionnaire for high school students showed that 60.5 percent of the students liked or liked very much this form of English learning, whereas only 2.3 percent disliked it; 60.5 percent of them will continue using the system after class, even without the teacher's urging.

In third term, the system was integrated into an English class in grade one of a junior middle school, while the other 15 classes didn't use it. The comparison of two examination results before and after the integration class shows great improvement of students' performance, and the survey data also indicates the students' preference for this system. The average exam score of the whole class improved from 64.39 to 90.81, whereas the standard deviation was decreased from 20.129 to 9.572. Moreover, compared with other classes not using CSIEC, the collective performance improvement of this class was remarkable. All the students hoped to continue to use this system in English learning. The average score before the integration was ranked number 16 in all 16 classes and 15.3 less than the number 1 class. After the integration, the average score was ranked number 2 and only 0.2 less than the number 1. Surely many factors influenced the score improvement, but because only this experimental class used the CSIEC system between the two tests, the significant score improvement must correlate with CSIEC integration.

Figures 9 and 10 depict students in middle school computer rooms using CSIEC under a teacher's guidance.

Figure 10. High School Students using CSIEC in the Computer Classroom.

Application Development and Deployment, and Maintenance

This application-driven research project was developed in 2002 and has been used by Internet users free of charge since its inception. Registered users now number more than 30,000, and the CSIEC homepage is visited more than 500 times every day.

I also cooperate with English teachers and integrate the system into English instruction. Since 2006, CSIEC has been used by four university classes and three middle school classes. All together, more than 500 students have used it in English class.

Konrad-Adenauer-Foundation Germany, the Ministry of Education China, Capital Beijing, Peking University, and the Korea Foundation for Advanced Studies have sponsored this project. It won the second prize in the national Innovation Competition in 2006, held by China's Ministry of Education and Ministry of Science and Technology. With much support, in the same year I founded the CSIEC research group, which includes graduate and undergraduate students, and I still lead it. The system is now located in the campus network of Peking University, where I continue to maintain and update it.

Discussion and Conclusion

The original goal of the CSIEC system was to provide a virtual chatting partner for English learners. Hence chatting is the fundamental function. The statistical analysis about users' behavior indicates that they have a preference for chatting without spelling and grammar checking. This fact proves that users prefer this unique chatting function, something that other systems lack. I must continue to reinforce this primary function.

The chatting quality can be somewhat shown by its duration. Thus an increasing percentage of longer and longer chatting shows that the free chatting quality of CSIEC is getting better. The underlying design principles—fully syntactical and semantic analysis of the user input, communicative response mechanism, as well as the effort of chatting personalization and adaptation—all contribute to the improvement of the chatting quality.

Chatting on a given topic is primarily used by students in the evaluation study and is also the main function of the whole system. The formal evaluation results indicate the application of the CSIEC system in English class can better assist their language learning by, for example, increasing their confidence in English communications and their interest in learning English, helping them master practical expressions, as well as improving listening skills. System functions such as free chatting, chatting on a given topic, and listening training have been brought into actual pedagogical play.

Through application and evaluation, I have also determined user requirements that weren't fulfilled, such as the system's stronger ability in natural language understating and generation, a fatal factor that plagues human-computer communication, the lifelike synthesized agent voice, and high response speed, which also have been addressed in the users' feedback. In natural language processing alone, many difficult problems are left unsolved, such as textual ambiguity and entailment, elements critical to natural language understanding and generation capability of the CSIEC system. Tackling these problems is a great challenge.

As for the system application and evaluation, I will continue cooperating with English teachers and monitoring the improvement in English ability of the students using the CSIEC system. In addition, I will explore more powerful applications of the underlying techniques of this system in other related fields, such as a computer-aided test for language learning, computer-assisted writing and translation, and so on.

Notes

1. www.csiec.com.

2. See the ALICEBOT website (www.alicebot.org).

References

Abu Shawar, B., and Atwell, E. 2007. Fostering Language Learner Autonomy Via Adaptive Conversation. Paper presented at the Corpus Linguistics Conference, Birmingham, UK, July 27–30.

Brennan, K. 2006. The Managed Teacher: Emotional Labour, Education, and Technology. *Educational Insights.* 10(2): 55–65.

Brown, J., Collins, A., and Duguid, P. 1989. Situated Cognition and the Culture of Learning. *Educational Researcher* 18(1): 32–42.

Fellbaum, C. 1998. *WordNet: An Electronic Lexical Database.* Cambridge, MA: The MIT Press.

Frege, G. 1879. *Begriffsschrift, eine der Arithmetischen Nachgebildete Formalsprache des Reinen Denkens.* Darmstadt: Wissenschaftliche Buchgesellschaft.

Fryer, L., and Carpenter, R. 2006. Emerging Technologies Bots as Language Learning Tools. *Language Learning &Technology,* 10(3): 8–14.

Graesser, A. C.; Chipman, P.; Haynes, B. C.; and Olney, A. 2005. AutoTutor: An Intelligent Tutoring System with Mixed-Initiative Dialogue. *IEEE Transactions on Education,* 48(4): 612–618.

Jia, J. 2004a. CSIEC (Computer Simulator in Educational Communication): A Virtual Context-Adaptive Chatting Partner for Foreign Language Learners. In *Proceedings of ICALT04 (International Conference on Advanced Learning Technology)*, 690–692. Los Alamitos, CA: IEEE Computer Society.

Jia, J. 2004b. NLML—A Markup Language to Describe the Unlimited English Grammar. Unpublished paper (see CoRR cs.CL/0404018).

Jia, J. 2004c. The Study of the Application of a Web-Based Chatbot System on the Teaching of Foreign Languages. In *Proceedings of Society for Information Technology & Teacher Education International Conference 2004*, ed. C. Crawford et al., 1201–1207. Cheasapeake, VA: Association for the Advancement of Computing in Education.

Jia, J. 2008. The Generation of Textual Entailment with NLML in an Intelligent Dialogue System for Language Learning CSIEC. In *Proceedings of the IEEE International Conference on Natural Language Processing and Knowledge Engineering 2008,* 194–201. Piscataway, NJ: Institute of Electrical and Electronics Engineers.

Jia, J.; Ye, Y.; and Mainzer, K. 2004. NLOMJ—Natural Language Object Model in Java. In *Intelligent Information Processing II,* ed. Z. Shi et al,, 201–209. Berlin: Springer Verlag.

Jonassen, D. H. 1994. Thinking Technology: Toward a Constructivist Design Model. *Educational Technology* 34(4): 34–37.

Kerfoot, P., et al. 2006. A Multi-Institutional Randomized Controlled Trial of Web-Based Teaching to Medical Students. *Academic Medicine,* 81(3): 224–230.

Kerly, A.; Hall, P.; and Bull, S. 2007. Bringing Chatbots into Education: Towards Natural Language Negotiation of Open Learner Models. *Knowledge-Based Systems* 20(2): 177–185.

Seneff, S. 2006. Interactive Computer Aids for Acquiring Proficiency in Mandarin. In *Proceedings of the Fifth International Symposium on Chinese Spoken Language Processing (ISCSLP 2006),* Lecture Notes in Artificial Intelligence 4274, 1–12. Berlin: Springer Verlag.

Von Glasersfeld, E. 1996. Introduction: Aspects of Constructivism. In *Constructivism: Theory, Perspectives, and Practice*, ed C. Fosnot, 3–7. New York: Teachers College Press.

Weizenbaum, J. 1966. ELIZA—A Computer Program for the Study of Natural Language Communications between Man and Machine. *Communications of the ACM* 9(1): 36–45.

Wittgenstein, L. 1918–21. *Tractatus Logico-Philosophicus.* Frankfurt am Main: Suhrkamp.

Jiyou Jia is an associate professor in the Department of Educational Technology, School of Education, Peking University, Beijing, 100871, China. His research interests include the application of artificial intelligence (natural language processing, expert systems, and so on) in education. He received a B.S. and a M. Education from Peking University and a Ph.D. in artificial intelligence from University of Augsburg, Germany. His email address is jjy@pku.edu.cn.