Computational Linguistics and Chinese Language Processing
Vol. 14, No. x, XXXXX 2009, pp. - 1

Speech-Based Interactive Games for Language Learning: Reading, Translation, and Question-Answering

Yushi Xu*, and Stephanie Seneff*

Abstract

This paper concerns a framework for building interactive speech-based language learning games. The core of the framework, the "dialogue manager," controls the game procedure via a control script. The control script allows the developers to have easy access to the natural language process capabilities provided by six core building blocks. Using the framework, three games for Mandarin learning were implemented: a reading game, a translation game, and a question-answering game. We verified the effectiveness and usefulness of the framework by evaluating the three games. In the in-lab and public evaluation phases, we collected a total of 4025 utterances from 31 subjects. The evaluation showed that the game systems responded to the users' utterances appropriately about 89% of the time, and assessment of the users' performances correlated well with their human-judged proficiency.

Keywords: Computer Aided Language Learning, Machine Translation, Automatic Question Generation, Automatic Answer Judging

1. Introduction

Computer aids for second language learning have long been a promising yet difficult research topic. Despite much argument about the best way to teach a second language based on pedagogy, the most natural and effective source of second language education is the classroom and human tutors. Statistics, however, have shown a severe shortage of language teachers, compared to the number of language learners. For example, the current estimated number of Chinese language teachers worldwide is around 40,000, while the number of people trying to

* Spoken Language Systems Group, MIT Computer Science and Artificial Intelligence Laboratory, USA
E-mail: {yushixu, seneff}@csail.mit.edu

[Received July 3, 2009; Revised October 7, 2009; Accepted October 22, 2009]

learn Chinese is about 1,000 times that[1]. The dramatic difference in the numbers not only results in many students not having a chance to find a suitable teacher, but also results in an under-emphasis on spoken communication, which many pedagogists agree to be an important skill, and which cannot be practiced by the student alone.

Given this situation, it is natural to think of replacing a costly human tutor with a computer. Several criteria, however, must be satisfied for such a machine tutor to be interesting to the students. The computer needs to understand the student's speech, and act intelligently enough to avoid being perceived as just an e-textbook. It should be able to offer a variety of activities, and to constantly provide rewards in order to motivate students to invest further effort to improve their skill level.

In an attempt to meet these requirements, we have developed a versatile framework for building speech-based language learning games. The core of the framework is a dialogue manager, which is supported by a set of building blocks, each providing some high-level natural language processing operations. By combining these operations in different ways using a control script, we have implemented three distinct games in two domains. The three games, a reading game, a translation game, and a question-answering game, provide different types of challenges to beginner learners of Mandarin Chinese. The two domains, general travel and flights, expose the students to different sentence patterns and vocabulary. The language processing operations provided by the building blocks are general-purpose, and the control script can be viewed as a high-level programming language. The whole framework thus makes it relatively straightforward to develop other speech-based language learning games, or to export the existing games to other domains of interest with minimal effort.

This paper will be organized as follows. We will first summarize some related work in Section 2. In Section 3, we will give a brief introduction of our three games. Then, in Section 4, the dialogue manager and its core building blocks will be described. Section 5 will describe the implementation of the three games in more detail, followed by their evaluations in Section 6. We will conclude and point to some future work in Section 7.

2. Related Work

There has been a significant amount of previous research in the computer aided language learning (CALL) field. Most of the research has a single focus, for example, vocabulary training (Brown, Frishkoff, & Eskenazi, 2005), or reading comprehension tests (Kunichika, Katayama, Hirashima, & Takeuchi, 2003). Only a few systems have been designed to provide alternative types of activities. Many of these integrated systems have been packaged as a CD-ROM as a delivery mechanism. The software is then installed on a local machine for

[1] Statistics according to China's Ministry of Education, 2006.

deployment. On the other hand, there are some Web-based language learning systems, such as Chengo Chinese (Chengo Chinese, 2004) and Active Chinese (Active Chinese, 2006). Both of these provide online Mandarin learning, which the user can access simply by opening up the web browser. These two systems provide several lessons ranging from easy to hard. In each lesson, a couple of activities and exercises are presented. Typically, the student first watches a conversation between some animated characters. Then, several important sentences are taught along with the vocabulary. After that, the student is expected to complete some pre-designed exercises. Although speech is enabled in both systems, the systems do not go beyond speech recognition. The user interacts with the system mainly via keyboard and mouse.

Examples of language learning systems that use speech as the main input modality are WordWar (McGraw & Seneff, 2008) and Rainbow Rummy (Yoshimoto, McGraw, & Seneff, 2009). In these two systems, the user talks to the system to select and move playing cards. Nevertheless, the systems are designed mainly for vocabulary learning, and do not emphasize other aspects like sentence formation or comprehension.

The work we present in this paper relies on several previously developed language processing systems. TINA (Seneff, 1992), the language understanding system, is a top-down parser, which uses a core context-free grammar augmented with additional rules to enforce long-distance constraints. Special features have been recently added to improve parsing efficiency for Chinese input (Xu, Liu, & Seneff, 2008). The output of TINA is a hierarchical meaning representation that does not explicitly encode word order information. The meaning representation can be converted back into a sentence via a language generation system GENESIS (Baptist & Seneff, 2000). GENESIS uses a context-sensitive lexicon to select appropriate word senses and a set of recursive rules to decide the order of the constituents. Depending on the choice of the rules, GENESIS can produce strings in any format, representing not only natural languages, but also formal languages, such as SQL and HTML.

3. The Games

In this section, we will briefly introduce the three games we have developed for Mandarin learning within the common framework. The games are Web-based and accessible from a shared URL. At the login page, the user chooses the genre of the game, the domain, and the starting level. Figure 1 shows screenshots of the translation game and the question-answering game.

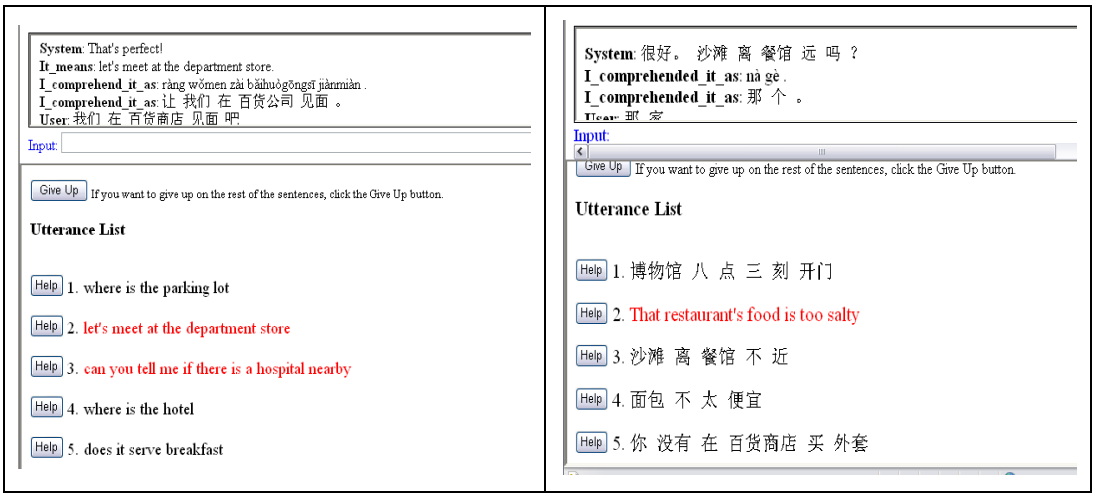

Figure 1. Screenshots of the translation game (left) and the question-answering game (right).

The main goal of the reading game is to help students learn Chinese characters. The student's task is to read out loud a list of Chinese sentences randomly generated by the system. The sentences can be displayed in either Pinyin or Chinese characters, depending on the student's preference. A help button is associated with each of the task sentences to provide a synthesized speech demonstration. To make the game more interesting, the student can read the sentences in any order. When the student records a spoken sentence, the system will not only echo his speech, but also provide an English translation of the sentence, even when it is not in the task list. If the student's speech matches any of the task sentences, the system will congratulate him and mark the sentence as completed. When all of the sentences are cleared, the system assesses the student's performance and reports a score. The game level is then adjusted according to the score.

When the student becomes more familiar with the Chinese characters and accumulates some vocabulary, he can start to play the translation game. In the translation game, instead of a list of Chinese sentences, the student is given a list of random English sentences. The student needs to construct a Chinese sentence of equivalent meaning by himself. Again, he can choose any order to translate the list. The system will echo the student's input, give a Chinese paraphrase and an English translation of it, and judge whether the input sentence is a correct translation of any of the task sentences. The judgment is based on syntax and semantics, so the student is allowed to translate in different ways. If he does not know how to translate a sentence, he can click the help button to hear and see a reference translation, presented in both characters and Pinyin. He can also type an unfamiliar English word or phrase in the input box to get a translation.

After playing the reading game and the translation game, the student should be prepared to try the question-answering game, in which the game scenario is almost completely in Chinese. The system randomly generates a list of Chinese statements, and then poses a question in Chinese based on one of the statements. The student needs to be able to read the displayed statements, to understand the spoken question, and to answer the question correctly in Chinese. Therefore, in this game, all of listening, reading, and speaking abilities can be practiced. Three chances are given for each question. The student can answer the question in various ways, either in short, in full, or somewhere between, as long as it is acceptable in Chinese. If the answer is correct, the corresponding statement will be turned into English. Otherwise, the system will give feedback according to the student's input, and guide her to a desired answer. As in the other two games, the student can ask for help, or ask the system to repeat the question if necessary.

In all three games, the student has an alternate input method. In a noisy environment where speech input is compromised, or if the student is having trouble being understood due to a heavy accent, they can opt to type their sentences into the input box using Pinyin format. The system will propose the character sequence based on the Pinyin input, and will also identify and mark all the characters that the student typed with an incorrect tone.

4. The Framework

The framework of our games is illustrated in Figure 2. The system consists of one or multiple speech recognizers, one or multiple speech synthesizers, the GUI interface, and the dialogue manager with a set of building blocks providing different NLP operations. The recognizers send N-best hypotheses of the student's input to the dialogue manager. After processing, requests are sent to the synthesizers to output the spoken responses. The dialogue manager also communicates with the GUI to receive user information and text input, along with updating the displayed content.

The dialogue manager is the core component of the framework. Together with its building blocks, it provides easy control over the processing steps during a dialogue turn. The control flow is managed by a set of control rules, called a control script. Each rule contains a parameterized operation and an optional trigger condition. The operations are provided by the building blocks. The framework contains six core building blocks. Two blocks, "create frame" and "paraphrase frame," use our pre-existing language understanding and generation systems. In addition to these two most basic NL operations, we have also developed four other core building blocks to handle game creation, management, and evaluation, which are very useful in developing language learning games. Besides the core building blocks, game developers can also provide their own specialized blocks to extend the capabilities of the dialogue manager.

The dialogue manager maintains a shared space representing the dialogue state. Both the dialogue state and the control rules are represented in Galaxy frame format (Seneff, Hurley, Lau, Pao, Schmid, & Zue, 1998). When executing a control script, the dialogue manager examines the conditions of each rule sequentially against the dialogue state. If the conditions are satisfied, the operation specified in the rule is executed. Several control rules can be grouped to form a macro to be reused in the script. An example is shown in Figure 3, in which the sequential operations of "create frame" (parsing) and "paraphrase frame" (language generation) form a Chinese-English translation macro. The macros are an important improvement over our previous design. Not only do they improve the readability of the control script, but they also support disjunction and iteration through recursive macro calls. These extensions provide much better control over program flow. Together with the high level NL operation, the framework provides an easy way for developers to construct different systems through specialized control scripts.

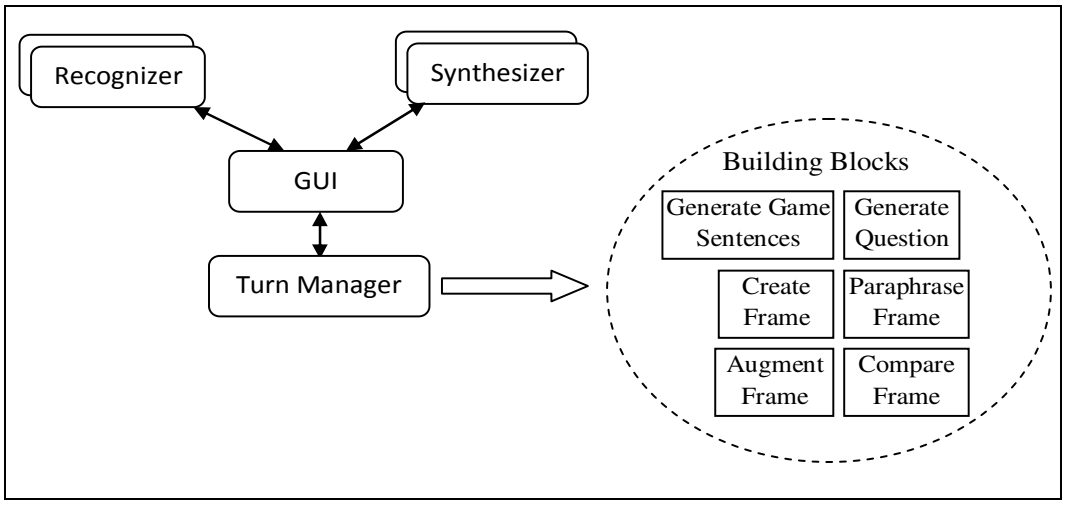

Figure 2. The framework

```
:Chn-Eng-Translate (
        {c rule
            :condition ":input_string"
            :variables {c variables
                            :domain "hanyu" }
            :operation "create_frame" }
        {c rule
            :condition ":parse_frame"
            :variables {c variables
                            :language "English" }
            :operation "paraphrase_frame" } )
```

Figure 3. An example of a Chinese-English translation macro.

In the following subsections, we will describe all of the operations that our core building blocks provide. We will also introduce some of the key macros that are useful in building the three game systems for language learning.

4.1 Generate Game Sentences

This operation controls the game level and generates one or a list of game sentences for the current game level. In the beginning of each game round, a list of task sentences is randomly generated from specified lesson templates. During the round, the operation marks the sentences that the student has completed, and, for games running in system control mode such as the question-answering game, it also chooses another sentence in the list for the next turn. When a round is completed, the operation calculates a performance score for the student and decides how to adjust the game level.

The operation generates the game sentences from a set of templates. The templates are divided into lessons, which can be organized by topics and/or grammar points. Each lesson has a list of sentence patterns and the associated vocabulary, just as in traditional textbooks. The patterns are essentially forest-like nodes, and the vocabulary is contained in the leaf nodes. The patterns and vocabulary introduced in the previous lessons are augmented by later lessons, which makes the templates easy to maintain. In generation, a starting pattern is randomly chosen from the specified lesson, and each non-leaf node is expanded with one of its child nodes based on a random selection process, until every node in the pattern is a leaf node.

In addition to the sentence generation that produces a single sentence in one language, we have also developed a more sophisticated generator that makes use of special non-context-free rules to automatically generate a pair of "synchronized" sentences in two different languages with the same meaning. By "synchronized," we mean that the sentence generator can generate a bilingual pair, guided by a special notation scheme in the templates. As shown in Figure 4, the vocabulary entries of the synchronized templates contain a vertical bar to separate the lexical entries for the two languages. Two special tags, "_L" and "_R", are used to deal with the different word order between the two languages. For instance, English and Chinese demand different positions for the prepositional phrase "from." In the example template, the pattern ":from" can generate a bilingual phrase "from the beach | 离 沙滩". ":from_R" means to take the right part of the output, which is the Chinese string, and put it before the adjective. Likewise, ":from_L" instructs it to take the left part of the output, which is the English string, and put it after the adjective. With this feature, it is easy to provide a generated string and its associated high-quality translation, which is very useful for many aspects of the games.

```
{c lesson
    :templates ( ":place :is :from_R :far :from_L" )
    :place ( "(the hotel | 宾馆)" "(the restaurant | 餐厅)" )
    :is ("(is | )")
    :from ("(from | 离) :attraction")
    :attraction ( "(the beach | 沙滩)" "(the park | 公园)" )
    :far ( ("very far | 很远") ) }
```

Figure 4. An example of the synchronized template. One possible output of this template can be "the hotel is very far from the beach \|宾馆离沙滩很远"

4.2 Create Frame and Paraphrase Frame

"Frame" here stands for "linguistic frame," which is a hierarchical meaning representation in the Galaxy frame format. "Create frame" and "paraphrase frame" are a pair of operations which convert between a string and a frame. Going from a string to a frame is the parsing process, and going in the other direction is essentially language generation.

As mentioned in Section 2, we rely on TINA and GENESIS for language understanding and generation in the games. TINA can be used to parse the template-generated game sentences, as well as the N-best list of the student's input. Besides the features mentioned briefly in Section 2, we further implemented a special two-pass parsing scheme in the operation "create frame". In Chinese, the way numbers and proper noun phrases are constructed often causes the parser's theories to grow exponentially when a generic grammar is applied. To avoid this situation, the two-pass parsing scheme first tags out these troublesome phrases using a very small shallow grammar, then creates parse trees for each of them (which we call element trees), and replaces the phrase with a single tag representing each element tree. Then, in the second pass, the parser creates a parse tree for the tagged sentence. Finally, the element trees are inserted at the appropriate locations in the second-pass parse tree to form a complete tree.

The language generation unit, GENESIS, also plays an important role in the system. It can be used to generate a paraphrase in the same language as the input, a translation into another language, a system's response, or an HTML string that can be displayed to the student.

For both TINA and GENESIS, we have developed generic grammar rules and generation rules in both English and Mandarin Chinese. The rules were developed based on the IWSLT[2] corpus, a spoken corpus of telephone quality speech collected from travelers. It covers a wide

[2] Internation Workshop on Spoken Language Translation

range of topics such as weather, flights, navigation, dining, shopping, sports, etc., is quite appropriate for everyday language, and is especially well suited to the needs of a traveler, which fits well with realistic roles for a language learner. With these generic rules, to export an existing game into a new domain of interest only involves adding a new lexicon corresponding to that domain, along with some other minor changes. The form of TINA's output, the linguistic frame, is quite suitable for language portability, especially because it disregards word order information. As most parts of the frame are language independent, we can convert the games for teaching Chinese into teaching English simply by reversing the grammar and generation rules.

For further information about TINA and GENESIS, along with their ability in paraphrasing and translation, we refer you to (Seneff, 1992) and (Baptist & Seneff, 2000).

4.3 Transform Frame

The function of this operation is to alter the elements in the frame. This has many uses, one of which is to convert a frame representing a statement into another frame that represents a question.

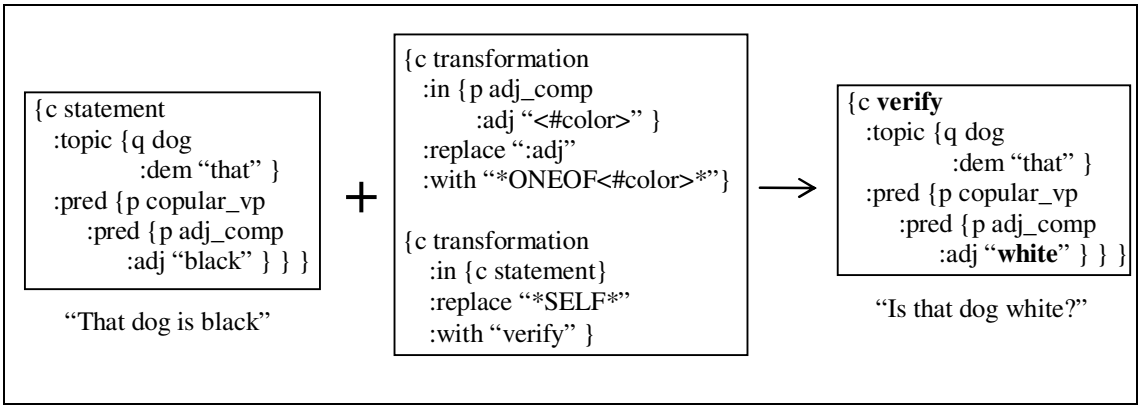

Figure 5. Example of a transformation rule with alternative choices.

The transformations are guided by formal rules. Each rule has three basic clauses, which describe the conditions under which the rule should be triggered, the part to be transformed, and the result after transformation. Wildcard values like ANY, NONE, SELF, etc., are adopted in the syntax to make the rules simple to write but powerful to express all kinds of transformations. A detailed description of the transformation rules can be found in Xu (2008). These transformation rules also support some randomness, allowing alternative outputs depending on a randomly generated outcome. Thus, as exemplified in Figure 5, the color adjective can be replaced with another random color via the rule.

4.4 Augment Frame

For our question-answering game, we need to deal with context resolution, since the answer would oftentimes be a fragment. Also we need to resolve its correctness in terms of both answering the question and not providing additional information that may be inconsistent with the given statement. For this task, we have developed a new building block which provides the "augment frame" operation. The operation does not depend on any domain-specific knowledge. In this algorithm, the frame representation of the previous utterance is aligned with the frame representation of the current utterance. Then, we can determine the omitted information and the pronoun referral in the current utterance, and we can augment the frame to include the complete information.

The alignment algorithm is based on two aspects: the anchor point and the similarity of the aligned frames. Depending on the type of the previous utterance, different anchor points are chosen. For wh-questions, the anchor point is the element that is questioned. For other types of utterances, the top level predicate is chosen. The best alignment is computed based on the constraint that the anchor point should be overlapping, and the similarity score of the two aligned frames is maximized.

Two examples are given in Figure 6. In the first example, the current short utterance "not far" is augmented into "the beach is not far from the hotel." by looking at the previous utterance, which is a yes-no question "is the beach far from the hotel?" In the second example, the previous utterance is a wh-question "where is the beach far from?". After augmentation, "hotel" becomes "the beach is far from the hotel". Note that, in this example, "hotel" is the topic of the current utterance. It, however, becomes the value of the key ":from" after augmentation, so that the anchor point "*question*" is overlapped.

This context resolution by augmentation approach has limited usage. It requires the topic and the basic structures of the utterances to remain unchanged. In semi-dialogue scenarios, like question-answering, this condition holds, and the augmentation algorithm is very effective. A short answer can be augmented into a complete answer by aligning it with the question. Then, if a follow-up question is posed based on the answer, the kv-frame of the second question can be augmented by aligning it with the augmented kv-frame of the previous answer.

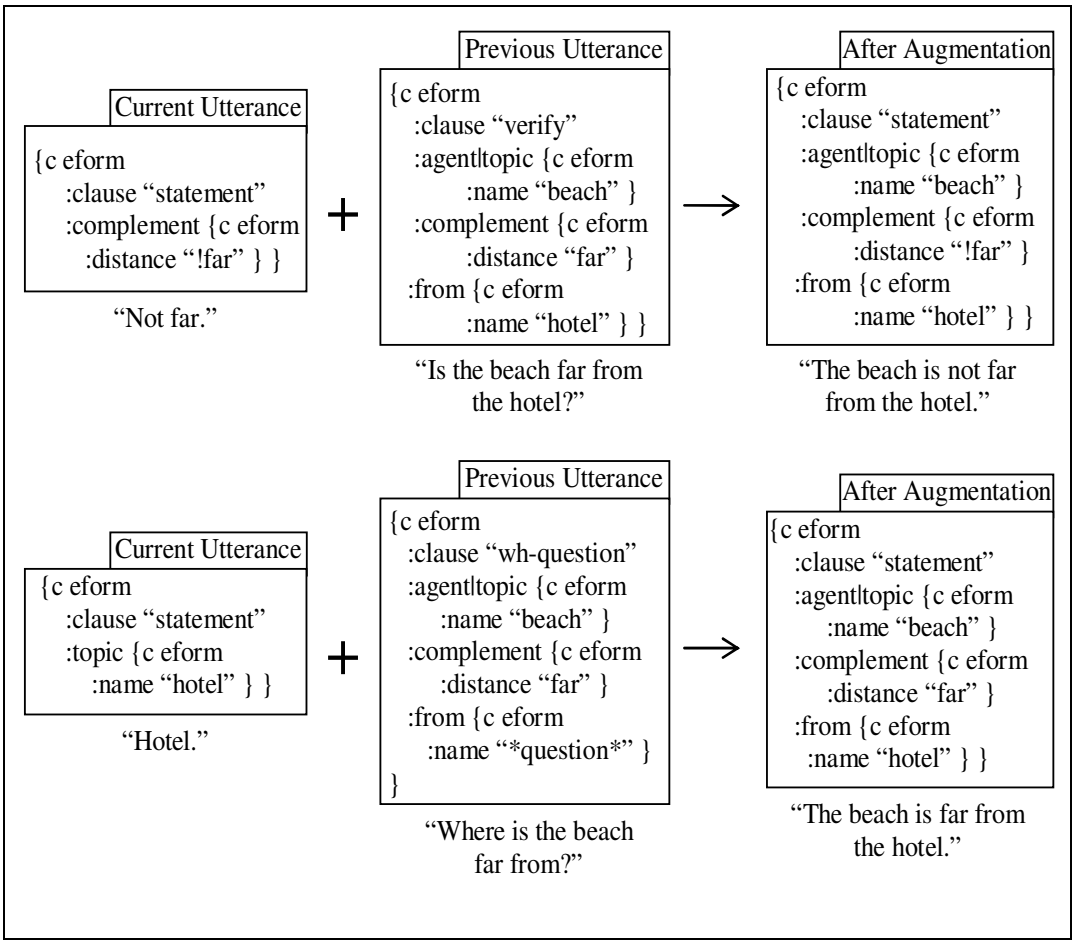

Figure 6. Examples of Context Resolution by Augmentation.

4.5 Compare Frames

This operation provides the ability to examine the differences between two frames. The comparison can be done blindly, *i.e.*, treating each element in the frame as equally important as the others. This setting is suitable when a direct match is desired. For example, in the translation exercise, students are encouraged to follow the way the original sentence is expressed, instead of paraphrasing "not far" into "close". If flexible expression is tolerable, the algorithm can perform a more heuristic comparison according to the parameters sent to it. Thus, it can be instructed to treat "not far" and "quite close" as having equivalent meaning. It can treat head words and modifiers differently, so that a mistake in the name of the patient will result in more deduction than a mistake in its color. It can also make different judgments for binary-value elements and multi-value elements, so that an insertion of the negation "not" will have a different comparison result from an insertion of a degree "quite".

The operation produces a summarization after the comparison, including the substituted elements, the inserted elements, the deleted elements, and an overall score. This output can be used not only to judge the correctness of the student's answer, but also to identify duplication or contradiction in the game sentences that were randomly generated.

4.6 Macros

Macros are formed when several operations are sequentially grouped together. We will introduce four useful macros in this subsection, which are diagrammed in Figure 7.

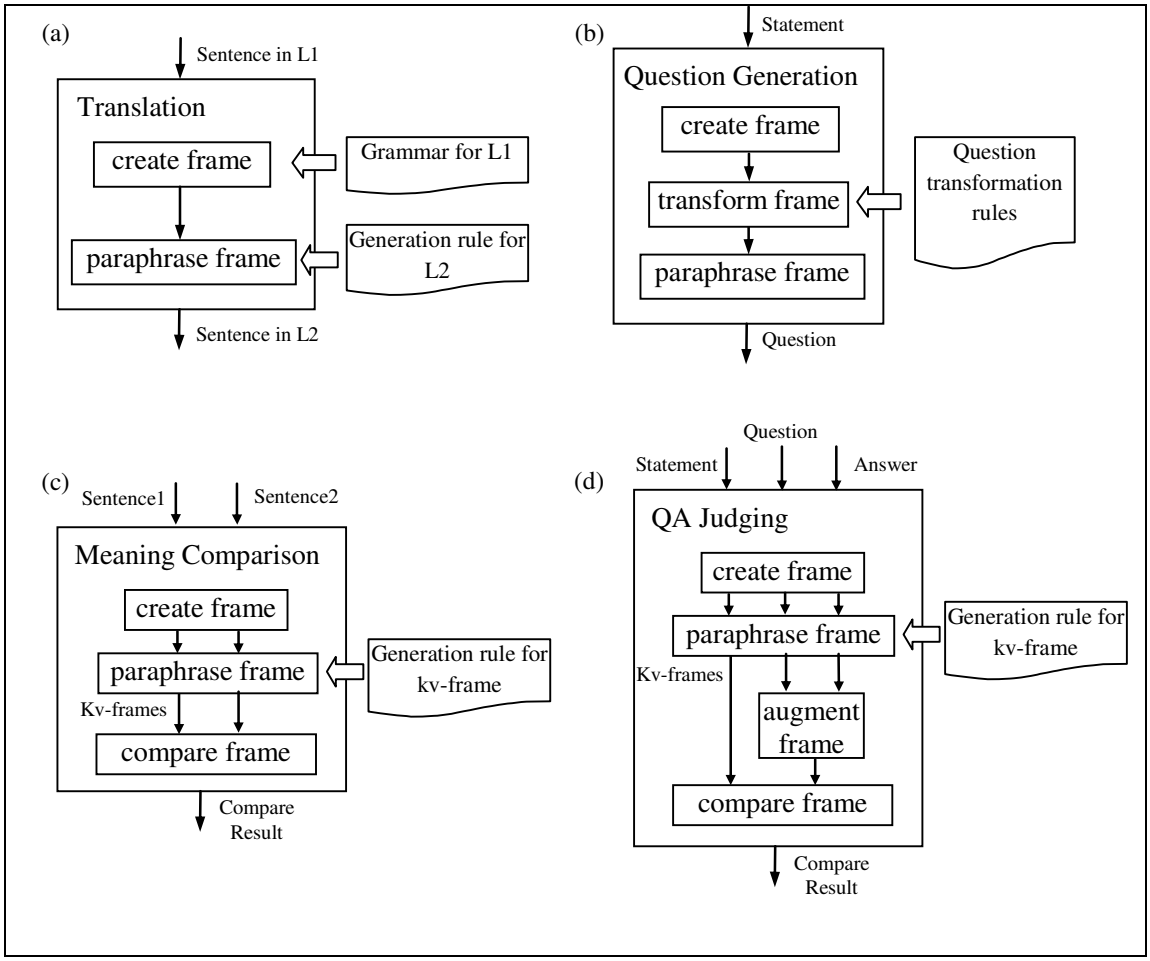

Figure 7. Macros: (a) Translation, (b) Question Generation, (c) Meaning Comparison and (d) QA Judging.

- **Translation.** The translation macro is very simple. The macro parses a string and produces a linguistic frame. Then, the generation rules for the target language are used to convert the linguistic frame into a well-formed text string.

- **Question generation.** This macro produces a question string from a statement string. The statement string is parsed into a linguistic frame. The linguistic frame is then transformed by question transformation rules, and, finally, a question string is generated from the transformed frame.
- **Meaning comparison.** This module compares the meanings of two linguistic frames by first converting them into kv-frames (key:value frames), which provide a more succinct representation of their semantic content. We use the "paraphrase frame" operation to generate the kv-frame. The actual comparison is then performed on the kv-frames instead of the linguistic frames from the parser.
- **QA judging.** This is very similar to the meaning comparison macro, except that, for question-answer judging, an additional step is taken to augment the answer kv-frame into a complete kv-frame by aligning the frame with the question kv-frame. Then, it is compared against the statement kv-frame.

5. The Game Implementation

In this section, we will show how basic operations and the macros described in the last section can be used easily to build different systems. The architectures of the arrangements of the operations and macros will be illustrated, with brief literal descriptions.

5.1 Reading Game

The first game we implemented was the reading game in the travel domain. Although the basic content of the game is very simple, interesting features were added to lessen the possibility of boredom. We wrote the lesson templates in English, rather than in Chinese. Then, we used the translation macro to automatically translate the sentences generated from these English templates into Chinese. This capability allows students to edit and create their own lesson templates without having knowledge of Chinese characters. The system can automatically tell them the corresponding Chinese. We also created an *inverse* translation macro. Whenever the student records an utterance, the system can provide the English meaning of the utterance he just read. When the student mispronounces a word, misrecognition may lead to an amusing English translation, which is more entertaining feedback than simply responding with "please try again". The system can also pronounce the sentence using the synthesizer when the student asks for help.

The framework of the reading exercise after adding these features is shown in Figure 8. The shaded blocks indicate the macros. Although it is a simple game, after utilizing the translation macro, the system already gives the student the impression that it understands what the student is speaking by providing an English translation.

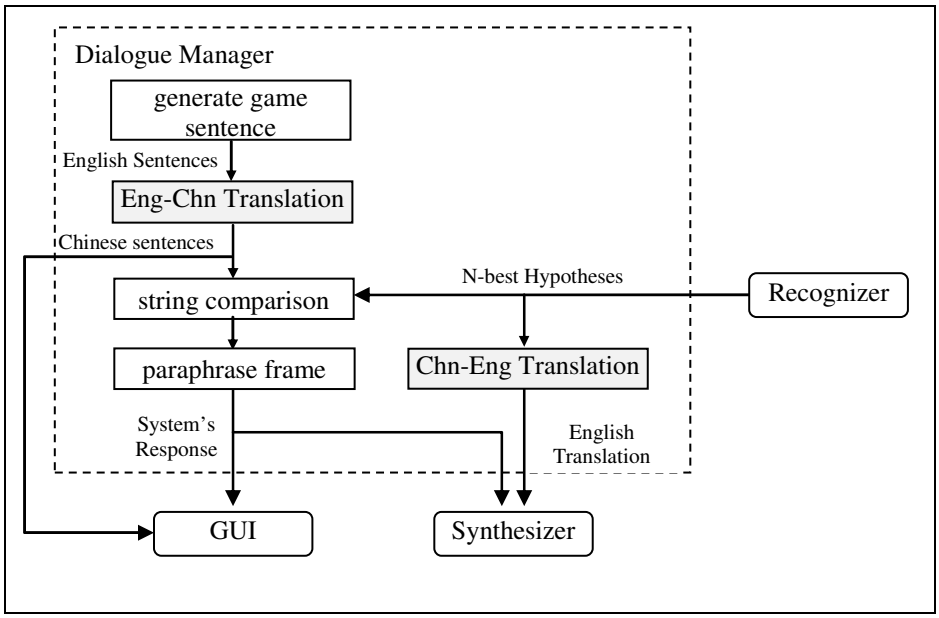

Figure 8. Framework of the reading game.

5.2 Translation Game

It is not difficult to extend the reading game so that it becomes a translation game. Figure 9 shows the framework of the translation game. It is almost exactly the same as the reading game shown above, except that the string comparison is replaced with the meaning comparison macro.

In the translation game, the system generates a list of game sentences from the English lesson templates and translates them into Chinese by the translation macro. This time, however, the English sentences are displayed instead of the Chinese or the Pinyin sentences. Another difference between the two games is that the reading game requires the student to read off the exact characters shown on the screen; in contrast, for translation, there is no unique answer. The student can translate a sentence correctly in multiple ways. So, instead of string comparison, the meaning comparison macro is adopted. To encourage the student to translate as literally as possible, the heuristic frame comparison is not used. Table 1 gives some examples of acceptable and unacceptable translations. The system echoes the student's speech, gives a Chinese paraphrase and an English translation of what the student said, and tells the student if the speech a match.

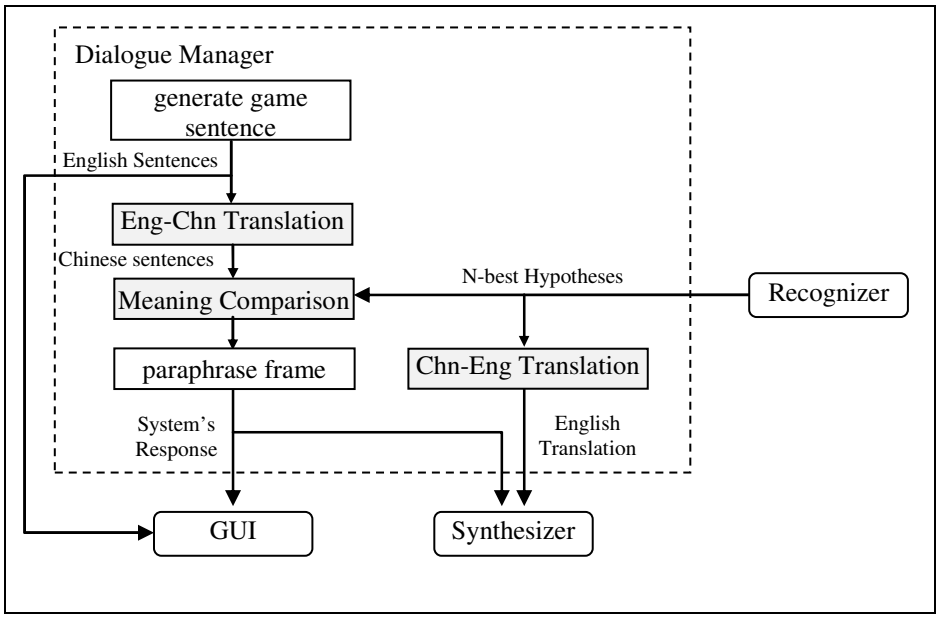

Figure 9. Framework of the translation game.

Table 1. Examples of accepted and rejected translations.

The museum opens at ten thirty.	Let's meet at the stadium.
√ 博物馆十点半开门	√ 让我们在体育馆碰头
√ 博物馆十点三十分开门	√ 咱们在体育馆见面吧
√ 博物馆于十点半开门	X 我们碰头在体育馆吧
√ 十点半博物馆开门	X 在体育馆见面
X 博物馆开门十点半	

Both the reading game and the translation game were developed in the travel domain first, and then exported to the more specific flight domain. The whole process of exporting, including writing new lesson templates, adding flight domain specific lexical and semantic information into the grammar and generation rules, training a new recognizer, and testing the system, took less than three weeks.

5.3 Question-Answering Game

The third game we built is a question-answering game. With the experience of the previous two games, this game was developed within two months, including the time spent developing the frame transformation rules for generating questions from statements. In this game, the student reads a list of statements on the screen, listens to the question posed by the system, and speaks the answer. When the answer is correct, the statement will be marked and turned into the English equivalent.

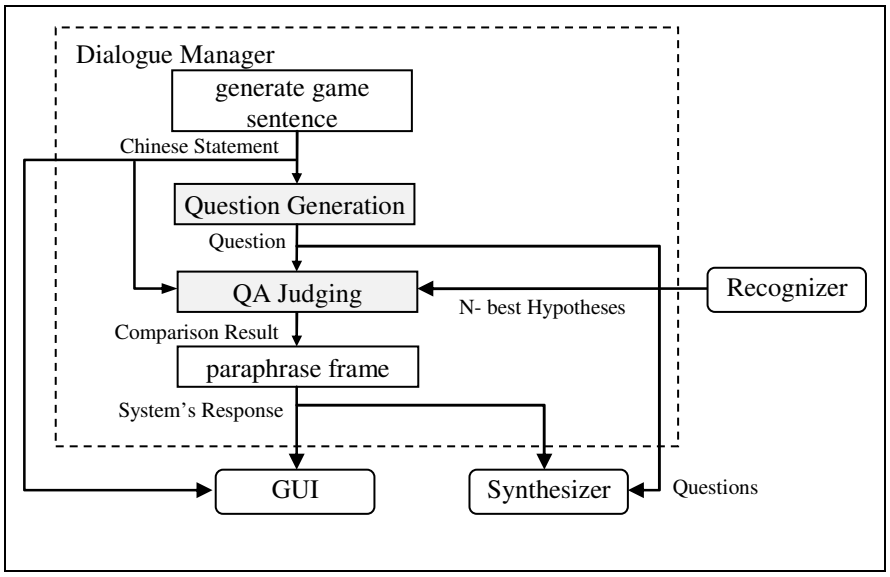

Figure 10. Framework of the question-answering game.

On the screen: 那只狗是黑色的 (That dog is black.)
你很喜欢吃蔬菜 (You like to eat vegetables very much)
System: 那只狗是白的吗？ (Is that dog white?)
Student: 不对，不是白的 (No, not white.)
System: 那么是什么颜色？ (So what color is it?)
Student: 是黑的 (It's black)
System: 很好。你喜欢吃蔬菜吗？(Good job. Do you like to eat vegetables?)
Student: 喜欢 (I do.)
System: 再具体一点 (Please be more specific.)
Student: 我很喜欢吃蔬菜 (I like to eat vegetable very much)

Figure 11. An example conversation between the system and the student in the question-answering game.

The framework of this game, shown in Figure 10, is a little different from the previous two, but not significantly. The translation macro is removed. Instead, we use the synchronized templates to generate Chinese statements and their English meanings at the same time. The Chinese sentences are processed into questions through the question generation macro. Transformation rules are written to include all kinds of possible questions. The question generation macro determines which rules apply to the current game statement, and randomly selects one to apply. After the student has spoken the answer, the QA judging macro takes in the N-best hypotheses from the recognizer, the questioned statement, and the question, and judges the correctness of the answer. Based on the comparison result, the system might give some advice or pose a follow-up question to guide the student to include the desired content in

the answer. An example of a conversation between the system and a student is given in Figure 11. From the conversation, we can see that the game is actually a simplified dialogue game, only that the dialogue is strictly limited in the scope of the game statements.

6. Evaluation of the Games

The most straightforward way of evaluating the framework and the dialogue manager is to evaluate the three games we implemented. As the games utilized different operations and macros, along with being connected in different ways, the effectiveness and flexibility of the framework can be proved by the successfulness of the three games.

We conducted the evaluation of the three games in two phases. In the first phase, we recruited several subjects to come to our lab, and gave them detailed instructions. In the second phase, we advertised our games to a list of users who are interested in Mandarin learning games and asked them to play the games by accessing a public URL via the Internet. We offered them gift certificates based on the amount of data they provided. They were less instructed on the games, and they might play the game in various environments. Due to the different settings of the two phases, we provide separate analyses for the two data sets. In both phases, we focused our evaluation on the system's performance, rather than proving pedagogical effectiveness. The reading game was not evaluated, because of its similarity to the translation game in terms of the architecture and the game procedure.

In all three games, we used SUMMIT, a landmark-based recognizer (Glass, 2003). The recognition output is constrained by an *n*-gram language model, that was trained using data automatically generated from our game templates. We developed an N-best selection process to score and select the hypotheses, choosing the one that best matched the dialogue context, if such an utterance existed.

We use two separate off-the-shelf synthesizers for synthesizing English and Chinese, respectively. Dectalk is used to synthesize English, and, for Chinese, a synthesizer provided by the Chinese Academy of Sciences is used.

6.1 In-Lab Evaluation Phase

6.1.1 The Translation Game

We implemented the translation game in two domains: travel and flights, which we did not distinguish during the evaluation. The lesson templates include twelve lessons for the travel domain and ten lessons for the flight domain. A single recognizer was used for both domains. The acoustics were trained from native speakers' data. An *n*-gram language model trained on the template-generated sentences augmented with IWSLT 2006 data was used to constrain the

recognition output. The vocabulary size was about 8.6K.

We recruited 5 subjects, 3 females and 2 males, to come to the lab. Each subject started at the first level, and was given five randomly generated utterances to translate in each round. We recorded the waveforms and the system's activity, as well as watching their behavior throughout their play. Advice was provided when they got stuck. Altogether, 615 utterances were collected from these five subjects.

We calculated the false rejection and false acceptance rate based on manual judgment. The false rejection rate was 8.6%, with almost all of the cases being caused by recognition errors. We listened to all of these waveforms and determined that most of the mis-recognized utterances were pronounced poorly or disfluently by the learners. The false acceptance rate was 0.9%. All of the false acceptances occurred when there was a minor syntactic problem in the sentence that was not identified by the system. For example, the user used an incorrect measure word for the noun. Encouragingly, we found that in the Chinese paraphrase the system gave back to the student, the syntactic problem had been automatically fixed, and we observed that the subjects did notice the implicit correction.

We calculated the average number of utterances the users spoke to complete one round, the average number of rounds they took to advance one level, and the average number of times per utterance they asked for help. The results are shown in Figure 12. The users are sorted on the horizontal axis to indicate their human-judged Chinese proficiency. The leftmost user is a native Chinese speaker. We can see that there is a good correlation between their real proficiency and the three values we measured. The users with lower proficiency tend to produce more utterances in one round, and tend to ask for help more frequently. The two numbers are the major factors for the system to assess the student's performance and to decide whether to adjust the game level. The result is that the poorer students tend to stay longer in the same level, as illustrated in the figure.

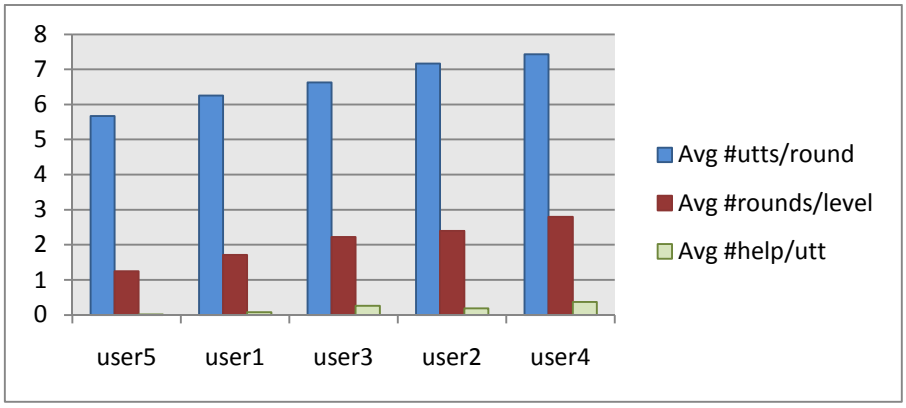

Figure 12. Performances of the users in the translation game. Users are arranged left-to-right in order of decreasing proficiency.

The game received positive feedback from the users. The users liked the feature that the system praised them, and they also appreciated the gradual introduction of new vocabulary and sentence patterns.

6.1.2 The Question-Answering Game

The question answering game was evaluated in a similar way as the translation game, but, a simulation phase was conducted as well to evaluate the quality of the questions and the coverage of the question types. The lesson templates are composed of seven lessons. Forty frame transformation rules were written to create 17 types of questions. We simulated 42 game rounds, 6 for each lesson. In each round, 5 statements and questions were generated. We determined manually that all the questions were well-formed. The distribution of the question types is illustrated in Figure 13. A fair percentage of yes-no questions and wh-questions were generated in the 210 questions, and within the wh-questions, the different types of questions were distributed reasonably.

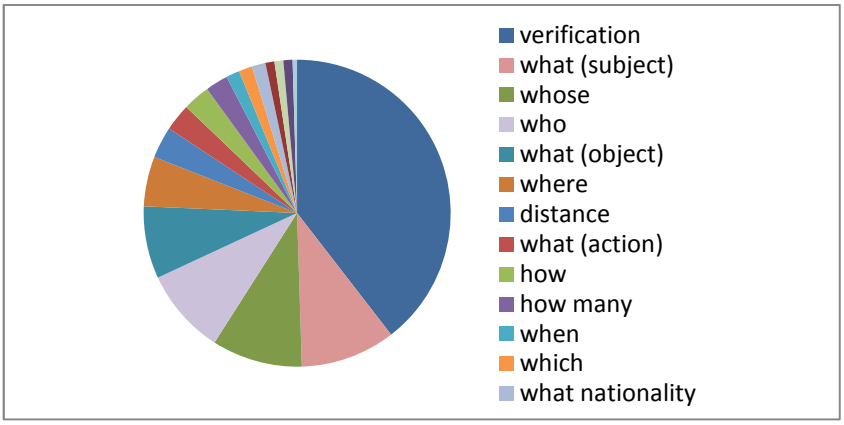

Figure 13. Distribution of the question types.

For the game system evaluation, we retrained the recognizer on an augmented synthetic corpus of utterances to model the statistics of both the translation game and the question answering game. The vocabulary size of the language model was enlarged to around 9K. Seven subjects, 3 males and 4 females, participated in the in-lab evaluation. Three of them were native speakers. Although the participants accessed the game from different computers, we ensured that they all used a high-quality microphone in a quiet environment. 732 utterances were collected from these subjects.

We categorized the utterances into three types of answers: blank-filling style short answers, such as a single yes/no or a single noun; full answers which essentially are a repetition of the statement in the list that answers the question; and other answers that are somewhere between the short answers and the full answers. The distribution of the three types,

shown in Figure 14, is quite balanced.

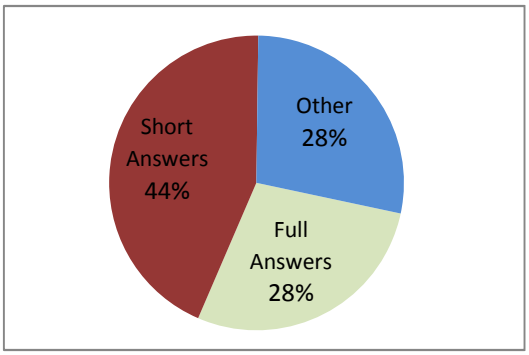

Figure 14. Types of answers

In the question-answering game, the system has several different responses instead of binary choices. Due to this, we calculated the accuracy of the responses instead of the FA/FR rates. The accuracy was 91.7%, with 57 out of 61 incorrect responses caused by recognition errors. The rest of the errors were caused by ill-formed kv-frames, which were fixed before the public evaluation phase.

6.2 Public Evaluation Phase

In this phase, we opened our games to the Internet users. An email message containing the URL and some game instructions was sent to a list of possibly interested users worldwide. We provided awards for the users who completed a certain number of game rounds. The users were free to choose to play any of the three games they liked, as well as to select their own initial game level. The number of utterances in each round was fixed at five.

In ten days, 23 users accessed our games, including three users whose data we discarded in the analysis due to quality issues: User 3 only provided two utterances in the middle of two game rounds of User 2; User 11 recorded his almost inaudible speech in an extremely noisy background; and User 12 used a poor-quality microphone which output highly saturated waveforms and resulted in a very high recognition error that was not comparable to that of any of the other users. All of the remaining 20 users tried the translation games; 9 also played the question-answering game; and 1 also tried the reading game. The 20 users include 7 females and 13 males. We manually judged their Chinese proficiency on a 5-point scale based on their pronunciation and intonation. Five points indicates a native speaker, and one stands for really poor pronunciation. The average proficiency score was 3.1, with four of the users judged to be native speakers.

From the 20 users, we successfully collected 1754 utterances for the reading/translation game, and 924 utterances for the question-answering game. We discarded 151 empty

utterances and 26 utterances that the dialogue manager did not receive due to communication problems. We also discarded utterances related to one problematic game sentence pattern, which produced an incorrect reference translation and led to confusion. This problem was fixed after the first two days of the experiment. After pruning, we were left with 1530 utterances for the reading/translation game, and 875 utterances for the question-answering game.

The overall sentence recognition error rate for all three games was 29.6%. Although this number is quite high, two factors played a critical role. Nearly a third (30.4%) of the mis-recognized sentences were either not a Chinese sentence, an ungrammatical Chinese sentence, or contained a totally mispronounced word. The other factor is that there were many repeated errors. When an utterance was not recognized correctly, the user usually spoke it again, essentially repeated verbatim, and it was very likely that the second utterance would not be recognized correctly as well. To verify this theory, we calculated the rate of repeated recognition errors. We define the rate of repetition to be the total number of mis-recognized utterances divided by the unique number of mis-recognized utterances. The unique number of mis-recognized utterance with recognition errors were counted independently within each game round, so that two identical misrecognized utterances in two different game rounds are distinguished. The rate of repetition of the three games was 1.77, which means that each unique recognition error is repeated almost twice. If the repeated errors are excluded, the sentence error rate for recognition goes down to 19.2%.

The recognition error rate also varies greatly among users, as shown in Figure 15. The users in the plot are sorted by their human-judged proficiency. It is clear from the plot that the recognition error is influenced greatly by factors other than their nativeness, which are likely to be microphone quality and environmental noise.

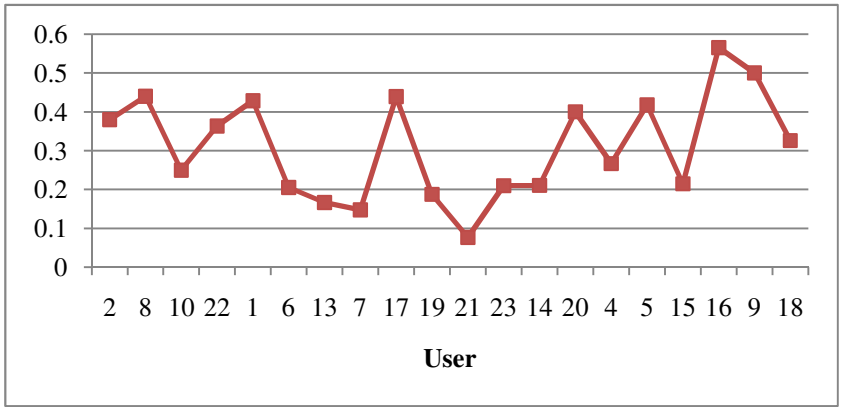

Figure 15. Sentence recognition error rate by users. Users are arranged left-to-right in order of decreasing proficiency.

Table 2 shows the error rates of the system responses. As in the in-lab evaluation, we calculated the false acceptance rate and the false rejection rate for the reading/translation game, and we did not distinguish the detailed error type for the question-answering game. We can see that the error rates were similar to those in the in-lab evaluation. Most of the errors were still caused by recognition errors. Others were mainly due to incorrect or missing information in our meaning representations. For example, "饭店" can mean either restaurant or hotel, but our linguistic frame only contains one of these interpretations. Also, we did not handle verb reduplication appropriately, so that in the utterance "请帮帮我" (please help me), we treated the two occurrences of the verb "帮" as two different verbs, and falsely rejected the utterance.

Table 2. Error rates of the system responses in the public evaluation phase

Game Genre	Error Type	Error Rate	% Caused by Recognition Error
Reading/Translation	False Acceptance	2.0%	90.3%
	False Rejection	11.6%	89.8%
Question-Answering	Incorrect Responses	9.8%	88.3%

In the public evaluation, it is more difficult to determine whether the users with poorer Chinese got more practice from simple statistics like average number of utterances they took per round. The problem is that the number of utterances per round is also dependent on environmental factors such as microphone quality and background noise level. We also notice that some users inexplicably repeated an already matched utterance, and thus had more utterances in each round. To take these two factors into consideration, we define a normalized average number of utterances per match as in Equations (1) and (2). In the equations, SER is the sentence recognition error rate, SER_{user} is the sentence recognition error rate attributed to users' mistakes. SER - SER_{user} gives the recognition error rate caused by other factors like background, channel, and acoustic models. Thus, a high c_{norm} means the user recorded in a quiet environment with a high-quality microphone. On the other hand, a low c_{norm} means the user probably used a poor recording device or played the game in a noisy environment.

$$\bar{u} = c_{norm} \times \frac{\#Total\ utterances}{\#Total\ matches} \tag{1}$$

$$c_{norm} = 1 - (SER - SER_{user}) \tag{2}$$

Figure 16 shows a plot of $\bar{u}$ for the users who completed at least one round of the reading/translation games. The users are sorted by decreasing Chinese proficiency. The logarithmic trend line illustrates that it took more effort for the lower proficiency user to complete a match. Two anomalously low points for User 7 and 16 result from their frequent actions of asking for help. They clicked help every two utterances on average, so their translations were mostly our reference translations, which were mistake-free and easy to

recognize. The high value of User 17 is due to his multiple repetition of two wrong translations which he probably thought to be correct.

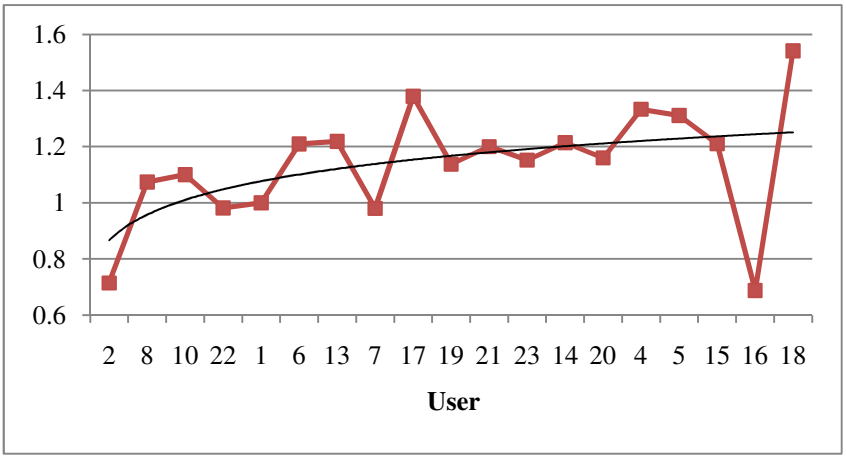

Figure 16. Normalized average number of utterances per match with the logarithmic trend line for the reading/translation game. Users are arranged left-to-right in order of decreasing proficiency.

For the question-answering game, we did not find a good correlation between ū and proficiency. In examining the log files, we determined that many users were confused with the pronoun reference of "you" and "I". Many users did not catch the conversational design of the game, and answered "your dad is Mike" when the system asked "who is your dad?". This confusion added much noise to ū, which resulted in it not being representative of the proficiency level.

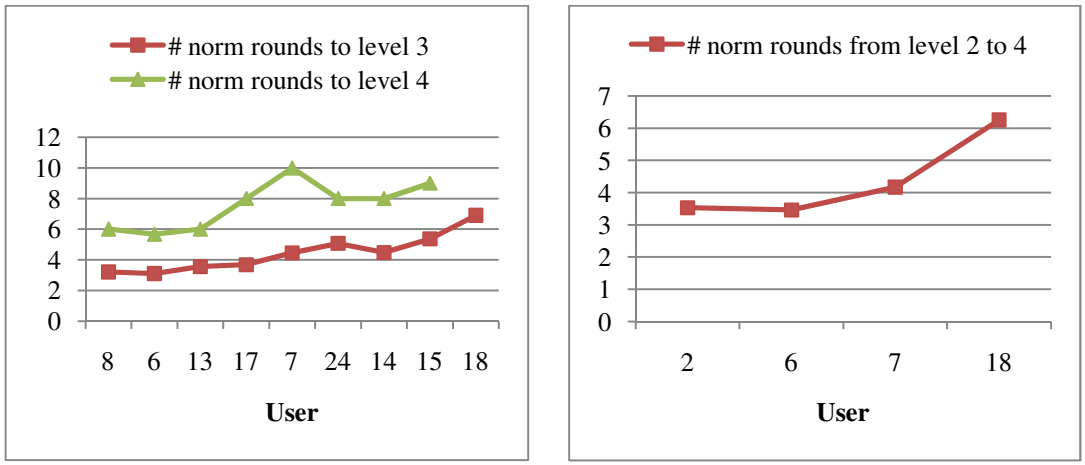

Figure 17. Normalized number of rounds to reach Level 3 and Level 4 for reading/translation game (left), and from Level 2 to 4 for the question-answering game (right). The users are sorted by decreasing human-judged proficiency.

We also analyzed how closely the system's assessment is related to the user's Chinese proficiency. Since many users did not play enough rounds, and often quit the last round in a session without completing it, it is not meaningful to calculate the average number of rounds per level. Instead, we counted how many rounds they took in one game session to reach Level 3 and Level 4 from Level 1 for the translation game. For the question-answering game, we noticed that it took the users one or two rounds to understand how to play the game, as well as the pronominal reference, so we discarded the information in Level 1 and counted the number of rounds they took from Level 2 to Level 4. The numbers of rounds are normalized by coefficient c_{norm} to reduce the differences in the recording conditions. The result is plotted in figure 17. It can be observed from the plots that as a whole, to reach the same level, users with lower proficiency spent more rounds, which means that our game has a reasonable assessment algorithm. The exceptional high number for User 7 to reach Level 4 resulted from an incomplete round at Level 3 which dropped him back to Level 2.

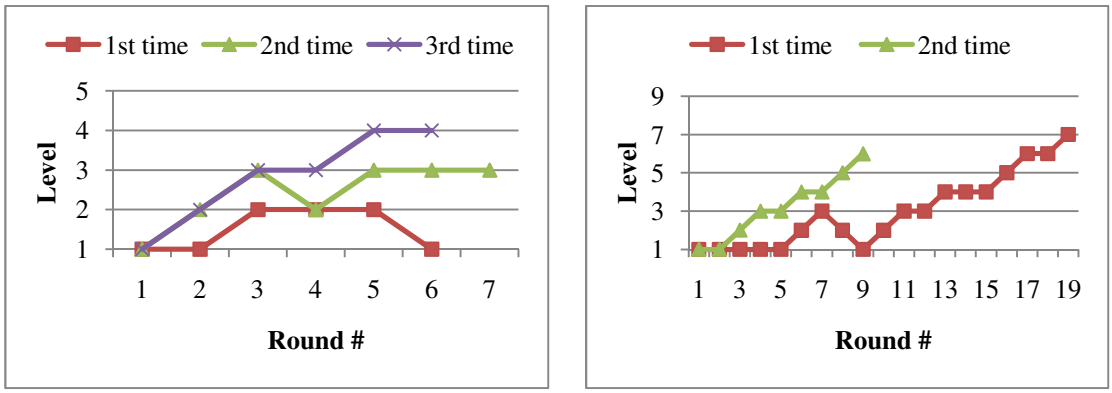

Figure 18. Levels User 18 achieved in different game sessions for translation game (left) and question-answering game (right).

Several users accessed our system multiple times. Among them, we noticed a low-proficiency user who played a total of 70 rounds. We found her making a lot of progress during these game plays. Figure 18 illustrated the levels she achieved in different game sessions. We can see that for the same number of rounds she reached a higher level when she repeated the game for a second and third time. The progress can be attributed to both increased acquaintance with the game and improvement in Chinese proficiency. For example, she had trouble with the syllable "chi" which she pronounced as "qi" causing much misrecognition. After several rounds, she realized the problem and tried hard to correct it. Finally, she learned the correct pronunciation and had it recognized correctly.

The users gave us considerable feedback on the games. In most of the feedback, the users showed their fondness for the games. Figure 19 shows some of the comments we received from the users. Most of the users found the games to be fun and helpful. They would like to

play again and recommend them to their friends. Some of the users also advised that the interface should be improved to become easier for first-time users. Some of the users were very careful and pointed out mistakes in the synthesized replies. Several users tried to explore the space that our system is able to handle by speaking their own utterances. Their feedback was very helpful for our future development.

"It's a confidence booster for one. When practicing speaking, it's nice to have it repeat back what I said and to know I said it right. You can't really get that with a human, it would probably drive them nuts."

"The hardest part of learning Chinese to me is finding someone to practice with. I haven't used any tool thus far that had such a great amount of feedback."

"It's a good way to learn new words."

"I think this is just good. Besides you already have other games focusing on vocabulary. Though for me building my vocabulary is important, making proper sentences in Chinese is even (more) important and compelling."

" (The game helps) Recalling different ways of saying the same thing."

Figure 19. Some of the comments from the users.

7. Conclusions and Future Work

We have developed a framework for building interactive speech-enabled language learning games. We introduced the Galaxy frame representation based dialogue manager, which operates according to a control script to enable the game developers to access natural language process capabilities in an easy way. Several generic building blocks have been newly developed, or adapted into the framework to provide different natural language operations, including game sentence generation, parsing, language generation, frame transformation, frame augmentation and frame comparison.

Three games have been built using the framework: a reading game, a translation game, and a question answering game. From the subject-based evaluation, we verified that the game systems were successful. The system responded to the users appropriately about 89% of the time. The assessment of users' performance correlated well with the users' true proficiency. The users were generally positive towards the systems. The success of the three games showed that the framework is useful. The dialogue manager handles the different game procedures correctly according to the control scripts, and the building blocks performed the desired functions correctly.

The complexity of the three games increases gradually. Starting from the simple reading game to the question answering game, more language processing units were utilized. As stated

in Section 5, the question answering game can be viewed as a semi-dialogue game, so the next step is to build a real dialogue game on the framework. In the question answering game, the approach to context resolution is simply by augmentation. This approach is simple, but it also limits the complexity of the dialogue. For a real dialogue game, a more generic approach would be needed. Also, more sophisticated dialogue management is required. Our group has developed dialogue systems in specific domains, and we believe that, with the help of these existing technologies, it would not be too hard to build a dialogue game for language learning purposes with domain and language portability. This will be the main focus of our future research.

Acknowledgements

This research was funded by ITRI and a grant from the Delta Electronics Environmental and Educational Foundation. We would like to thank Ian McGraw for his help in providing the WAMI toolkit that made development of the Web interface relatively easy. We would also like to acknowledge Anna Goldie for her help in making up the lessons for the question-answering game.

References

Active Chinese. (2006). Retrieved 2008, from Active Chinese: http://www.activechinese.com/.

Baptist, L., & Seneff, S. (2000). Genesis-II: A Versatile System for Language Generation in Conversational System Applications. *Proc. ICSLP*, (pp. 271-274). Beijing, China.

Brown, J. C., Frishkoff, G. A., & Eskenazi, M. (2005). Automatic question generation for vocabulary assessment. *Proc. HLT*, (pp. 819-826). Morristown, USA.

Chan, M. K. (2003). The Digital Age and Speech Technology For Chinese Language Teaching and Learning. *Journal of the Chinese Language Teachers Association ,* 38(2), 49-86.

CHANG, J. S., & CHANG, Y.-C. (2004). Computer Assisted Language Learning Based on Corpora and Natural Language Processing: The Experience of Project CANDLE. *An Interactive Workshop on Language e-Learning*, (pp. 15-23). Tokyo, Japan.

Chengo Chinese. (2004). Retrieved 2007, from Chengo Chinese: http://www.elanguage.cn/

Ehsani, F., & Knodt, E. (1998). Speech Technology in Computer-Aided Language Learning: Strengths and Limitations of a New CALL Paradigm. *Language Learning & Technology ,* 2(1), 54-73.

Glass, J. (2003). A probabilistic framework for segment-based speech recognition. *Computer Speech and Language ,* 17(2-3), 137-152.

Kunichika, H., Katayama, T., Hirashima, T., & Takeuchi, A. (2003). Automated Question Generation Methods for Intelligent English Learning Systems and its Evaluation. *Proc. ICCE2004.*

McGraw, I., & Seneff, S. (2008). Speech-enabled Card Games for Language Learners. *Proc. AAAI.* Chicago, USA.

Seneff, S. (1992). TINA: A Natural Language System for Spoken Language Applications. *Computational Linguistics ,* 18(1), 61 - 86.

Seneff, S., Hurley, E., Lau, R., Pao, C., Schmid, P., & Zue, V. (1998). GALAXY-II: A Reference Architecture for Conversational System Development. *Proc. ICSLP.* Sydney, Australia.

Xu, Y. (2008). *Combining Linguistics and Statistics for High-Quality Limited Domain English-Chinese Machine Translation.* Master's Thesis, MIT, Cambridge, Massachusetts.

Xu, Y., & Seneff, S. (2008). Two-Stage Translation: A Combined Linguistic and Statistical Machine Translation Framework. *Proc. AMTA.* Honolulu, USA.

Xu, Y., Liu, J., & Seneff, S. (2008). Mandarin Language Understanding in Dialogue Context. *Proc. ISCSLP*, (pp. 113-116). Kunming, China.

Yoshimoto, B., McGraw, I., & Seneff, S. (2009). Rainbow Rummy: A Web-based Game for Vocabulary Acquisition using Computer-directed Speech. *Proc. SIGSLaTE 2009.* Warwickshire, UK.