Automatic Question Generation and Answer Judging: A Q&A Game for Language Learning

Yushi Xu, Anna Goldie, Stephanie Seneff

Spoken Language Systems Group
MIT Computer Science and Artificial Intelligence Laboratory, United States
{yushixu, agoldie, seneff}@csail.mit.edu

Abstract

We have designed a question and answer game for students learning Mandarin Chinese. The game produces spoken questions from automatically generated statements, and judges the student's answers automatically. The student interacts with the system by speech, so that comprehensive reading, listening and speaking ability can be practiced. This paper focuses on the methods for question generation and answer judgment, as well as the game implementation. Evaluation results have shown that our methods and the game system are both successful.

1. Introduction

Computers have increasingly played a role in the language learning world. While many tools have been developed and have proved effective (e.g., simple flash card tools for learning vocabulary), our group is most interested in developing intelligent tutoring systems that can interact with students in a more natural way, i.e. via speech. This is not only important in the sense that the system will be more like a real human tutor, but also crucial in that it creates chances for the students to practice speaking anytime and anywhere.

The Web-based game system we describe in this paper is inspired by traditional reading comprehension exercises, in which the student is given a passage to read, along with several questions based on the passage. In order to answer the questions correctly, the student needs to understand both the passage and the questions, which are both written in the target language. Human effort is required in the preparation of the exercise as well as in the evaluation phase: someone must produce some questions from a passage, and after the student finishes the exercise, somebody must judge the answers. Both aspects involve significant human effort in the traditional language teaching situation, especially when the questions are not multiple choice. In this paper, our goal is to design an online computer exercise that can achieve a similar function. We have designed a speech-enabled prototype game to verify that the idea is feasible, although the reading comprehension exercise is simplified. The student reads several isolated statements instead of a coherent passage, listens to each system-generated question, and answers by speaking aloud. The system then judges the correctness of each answer.

2. Previous Research

The idea of automatically generating questions from sentences has been proposed and implemented as early as 1976. Since parsing technologies at that time were not mature, string pattern matching was used instead of syntactic analysis [1]. More recently, Kunichika et al. developed a multimedia language learning system for teaching English to Japanese students [2]. The system is an integrated environment for textbook authors and students. The system processes the data which the authors input through the NLP component and provides some guidance while the student is using the system. One of the intelligent functions of the system is to automatically generate comprehension tests based on the text material. The detailed approach is described in a later paper [3]. Questions were generated about the content by first parsing the English sentences into semantic representations, then replacing one word/phrase with an interrogative word corresponding to that particular semantic class, and finally changing the sentence into a question form. Although the tutor asks the questions orally, the student can only answer them in text.

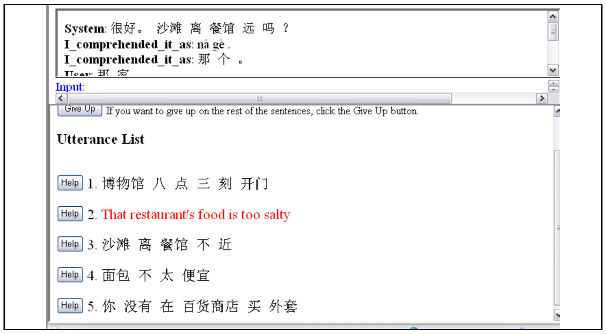
Our group has developed fairly mature language processing technologies. We have been using the language understanding system TINA [4] and language generation system GENESIS [5] for various applications. The two systems can be cascaded through an interlingua representation which we call a linguistic frame, i.e., TINA produces a linguistic frame from a string, and GENESIS generates a string from the frame. Generic grammars and generation rules for both English and Mandarin Chinese have been developed. Depending on the choice of grammar and generation rules, the cascaded systems can perform an in-language paraphrase or a cross-language translation. We have developed speech-enabled translation games [6] using these systems, and the game we will describe in this paper leverages much of the game interface and procedure from the translation game.

The rest of the paper is organized as follows: in Section 3, we will describe the game along with the approach we use to generate questions and judge answers. In Section 4, the experimental setup and evaluation results will be discussed, followed by conclusions in Section 5.

3. Game System

3.1. Game Interface

Figure 1 shows the game interface, most of which is borrowed from our previous translation game. The student chooses the difficulty level and number of sentences he/she wants to practice in each round. A list of parallel statements (in both English and Chinese) is randomly generated from lesson templates, and the Chinese statements are displayed on the screen. Then the system poses a question based on any one of the statements. The student listens to the question, reads the statements to determine the appropriate response, and answers orally. The system judges the answer, and, if it is correct, the corresponding statement will be turned into English and marked in red. The student has three chances to correctly answer each question. Upon failure, the system speaks the statement that corresponds to the answer, for instructional guidance. After all the questions have been asked, the system reports a score and also decides whether to adjust the difficulty level (up or down) for the student.



System: 很好。 沙滩 离 餐馆 远 吗 ？
I_comprehended_it_as: nà gè .
I_comprehended_it_as: 那 个 。
User: 那 家

Input:

Give Up If you want to give up on the rest of the sentences, click the Give Up button.

Utterance List

Help 1. 博物馆 八 点 三 刻 开门

Help 2. That restaurant's food is too salty

Help 3. 沙滩 离 餐馆 不 近

Help 4. 面包 不 太 便宜

Help 5. 你 没有 在 百货商店 买 外套

Figure 1. Game Interface.

3.2. System Framework

Figure 2 illustrates the high level game framework. The sentence generator produces a list of parallel statements according to the current difficulty level using a recursive grammar. A special tied-rule syntax was developed to encode a bilingual lexicon in the lesson templates, as well as to allow for different word orders in the English and Chinese sentences. The statements in the list are drawn randomly from both the current level and the previous levels. The system displays the Chinese sentences and uses them for question generation. The English half is used only when the student asks for help. The question generator randomly picks one Chinese statement, generates a question and sends it to the GUI for synthesis and playback. When the student speaks his answer, the speech recognizer captures the audio waveform, and converts it into text. The answer judger decides, based on the original statement and the question, whether or not the student's answer is correct.

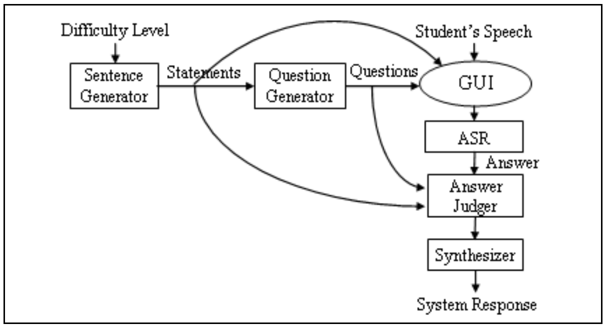


Figure 2. Framework of the game system.

This paper focuses on our methods of generating questions and judging the student's answer. Other components will only be mentioned briefly. We use SUMMIT [7] for speech recognition. The acoustics are trained from native Chinese speakers' data, and an *n*-gram language model with vocabulary size 9K is used to constrain the recognition output. The language understanding, i.e. parsing and producing the linguistic frame, is handled by TINA with a generic Chinese grammar. The language generation is handled by GENESIS. More detail on this process can be found in [6].

3.3. Question Generation

Our linguistic frame is a hierarchical representation which encodes a mixture of syntactic and semantic information. The frame elements are key-value pairs, in which the value can be a string, a number, or another frame. For example, in the linguistic frame shown in Figure 3, the frame name "cstatement" indicates the type of clause. ":topic" corresponds to a noun phrase, which might be a subject or an object depending on its position in the frame. The frame uses some syntactic notions such as complement and adjective, and it may also have some explicit semantic information such as locative, temporal, nationality, etc.

```
{c cstatement
     :topic {q beach }
     :pred {p copular_vp
               :pred {p from
                      :topic {q hotel } }
               :pred {p adj_complement
                      :adj "close"
                      :degree_adv "very" } } }
```

Figure 3. An example of a linguistic frame representing the sentence "沙滩离饭店很近" (The beach is very near the hotel).

As indicated above, we have already developed generation rules to convert a given linguistic frame into a surface string in the target language. Because the linguistic frame preserves the semantic hierarchy of a sentence, generating a question simply involves (1) changing the clause type, and (2) if it is a wh-question, replacing some element in the frame with a special interrogative element. The special element we define has a key ":trace" (this nomenclature derives from English wh-movement). The modified frame can be converted into a question automatically by GENESIS.

The transformation of the frame is realized by applying a set of formal rules [8], whose syntax is designed such that they are simple to write, yet powerful for describing many different manipulations on the frame. The rules can have a simple function such as changing the value of a key, or may describe a complex operation, especially when rules are combined in sequence. Table 1 gives one example and the linguistic frame which results from applying the rules to the frame in Figure 3. The topic "hotel" under predicate "from" is changed into a generic object with a trace "where". Then the name of the clause is changed to "wh_question". The special value "<#loc_noun>" specifies a set of location nouns so that they can share the same transformation rules. The generation output of the resulting linguistic frame is "沙滩离哪里很近？" (where is the beach near to?).

There are three advantages to transforming the linguistic frame rather than parse trees or surface strings. First, the linguistic frame is, for the most part, source language independent. This means that, whichever language it comes from, the same transformation rules can be shared. Secondly, the linguistic frame contains both syntactic and semantic information that is useful in determining the context information. For example, the location nouns are not always turned into "where". In some contexts, they are treated the same as other nouns, and "what" is chosen. Finally, the transformation produces a linguistic frame that represents a question. The question sentence is then generated from this meaning representation, rather than from the original statement sentence. Any word order differences or even the word choice differences between the statement and the question are automatically captured by the generic generation rules. These differences are sometimes context dependent, and would not be captured precisely if simple string rewrite rules were used to produce the questions.

Table 1. Example frame transformation rules, and the result of applying them to the frame in *Figure 3*. **Bold** text marks the difference from the original frame.

Transformation Rule	Resulting Linguistic Frame
{c transformation_rule :in {p from :topic {q <#loc_noun> :*submatch* 1 :*focus* 1 } :replace "*SELF*" :with {q object :trace "where" } } {c transformation_rule :in {c cstatement} :replace “*SELF*” :with “wh_question” }	{c **wh_question** :topic {q beach } :pred {p copular_vp :pred {p from **:topic {q object** **:trace “where”}** } :pred {p adj_complement :adj “close” :degree_adv “very” } } }

We wrote 35 rules to describe different kinds of questions we want to generate. During the game, the system first randomly selects one of the displayed sentences, then determines which rules apply to that sentence, and finally randomly chooses one of those rules to apply.

3.4. Answer Judgment

There are many ways to answer a question correctly in Chinese, as Chinese has a relatively free grammar. The answer to a yes/no question doesn’t necessarily start with a “yes” or “no”, and the answer can repeat some information as well as omit some information. For example, a positive answer to the question “do you like to drink beer?” can be “yes”, “like”, “I like”, “like to drink”, “yes, I like”, etc. An answer to a wh-question “which beer do you like to drink?” can be “this”, “this beer”, “drink this beer”, “like to drink this beer”, etc. The system needs to accept all these possible answers, and at the same time reject incorrect answers.

The judging algorithm is based on key-value pairs. From the original linguistic frame, we further refine the elements into a set of compact key-value pairs (kv-frame), in which synonyms are collapsed and some negations are propagated into subframes, as exemplified in Table 2(a). We first augment the student’s answer to include all the omitted information. We treat the augmentation problem essentially as a discourse phenomenon. Later information, i.e. the student’s answer, overwrites earlier information, i.e., the question. Since the question and answer kv-frames have different sizes and depths (the answer kv-frame is usually smaller and shallower), the problem becomes an alignment problem, i.e. to align the answer kv-frame with some sub-frame of the question kv-frame.

Our algorithm treats wh-questions and yes-no questions differently because the types of allowable omission are different. For wh-questions, the question kv-frame must contain one key which has the coded value “*question*”. We use that key as an anchor point to align the answer kv-frame with the question kv-frame. The alignment score is based on two criteria: (1) the similarity of aligned frames, and (2) whether the questioned key is covered. We choose the best scoring alignment, use the question kv-frame as a base and overwrite the values in the question kv-frame with the ones in the answer kv-frame.

For yes-no questions, the algorithm is more complicated. Two things must be noted. (1) In a wh-question asking about the object, it is correct to only say the object. However, this is not true in a yes-no question. The verb, or the top level predicate must also be repeated. So instead of the question kv-frame, we use the answer kv-frame as the base frame, and fill in missing information from the aligned question subframe. By doing so, no upper level information will be created after augmentation. (2) If the answer also contains “yes” or “no” at the beginning, this word should not contradict the rest of the answer. We create two augmented kv-frames to check this consistency, one from the yes/no only, and the other from the rest of the answer. Only if the two augmented kv-frames are not contradictory is the answer considered correct.

Table 2. Examples of augmenting the answer kv-frame.

a	b
Question kv-frame for “沙滩离饭店远吗” (Is the beach far from the hotel?)	Question kv-frame for 沙滩离哪里很近? (Where is the beach very near to?)
{c eform :agent|topic {c eform :name “beach” } :complement {c eform :adj “far” } :from {c eform :name “hotel” } }	{c eform :agent|topic {c eform :name “beach” } :complement {c eform :adj “close” :degree “very” } :from {c eform :name “*question*” } }
Answer kv-frame for “不远” (not far)	Answer kv-frame for “饭店” (hotel)
{c eform :complement {c eform :adj “!far” } }	{c eform :topic {c eform :name “hotel” } }
Augmented answer kv-frame	Augmented answer kv-frame
{c eform :agent|topic {c eform :name “beach” } :complement {c eform :adj “!far” } :from {c eform :name “hotel” } }	{c eform :agent|topic {c eform :name “beach” } :complement {c eform :adj “close” :degree “very” } :from {c eform :name “hotel” } }

One problem with this alignment algorithm is that the similarity score depends on the keys associated with the values. However, in short answers, the key assigned to a word might be different from the one in the question. For example, if the question is “where can Alice buy a shirt?” and the student answers “department store”, the key for the phrase “department store” might be “:topic” instead of “:loc”. To solve this problem, we create a special alignment option for the answer kv-frame: if the answer kv-frame contains only one key, the alignment can be performed on the sub-frame of the answer kv-frame, so the top level semantic label can be ignored. An example is shown in Table 2(b).

The augmented answer kv-frame is compared with the kv-frame of the original statement to decide whether the student’s answer conveys the correct meaning.

3.5. Contradiction Detection

Early on, we realized that there exists another problem even before question generation. The statements the system produces represent a scenario to some extent, so there should be no contradictory statements in the list, i.e. sentences like “the shop opens at nine” and “the shop opens at ten” should not appear in the same list. Therefore we introduced a contradiction detection component into the sentence generator. We defined a special set of keys to be “dominant”, i.e., they differentiate between different events. For example, if the topics are different, the two statements should not contradict because they are talking about different events. The algorithm first excludes all no-contradiction situations by examining the dominant keys. Then, if a difference in the values of any non-dominant keys is found, a contradiction is signaled. The possible dominant keys and their order are manually specified, and then adjusted according to the specific sentence pair. Different contradiction conditions can be achieved by manipulating the possible dominant keys and their order.

4. Evaluation

4.1. Question generation and contradiction detection

We composed seven lessons based on our previous translation game. 500 unique statements were randomly generated from the lesson templates, and 1746 questions were subsequently generated from the statements. We manually verified that all of the questions are well-formed in terms of both syntax and semantics. The questions were manually categorized into 17 different types. To assess the distribution of the question types in real game play, we simulated 6 random game rounds for each of the seven lessons, and tabulated the distribution of types as shown in Figure 4. Fifteen out of 17 types were observed in the resulting 210 questions.

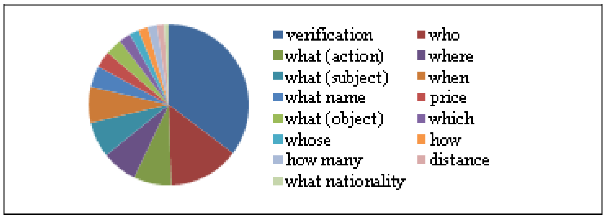


Figure 4 .Pie chart of different types of questions.

For contradiction detection (CD), we randomly generated 50 10-statement lists from each of the 7 lessons. A manual evaluation found contradictions in 141 of the 350 lists without CD. None was observed with CD in place.

4.2. System Evaluation

Six subjects, three males and three females, took part in the evaluation by accessing the game from their own computers via the Internet. Four of the subjects are learners of Chinese, and the other two are native speakers of Chinese. The evaluation is focused on how well the system responds to the student, rather than pedagogical effectiveness. We collected 636 utterances from the subjects. 41.0% of the utterances were blank-filling style, i.e., a single yes/no, or a single noun, 28.6% were a complete repetition of the statement that answers the question, and the rest 30.4% were answers somewhere between the two types above. The false rejection rate was 9.0%, 98.2% of which were caused by recognition error. 0.6% of the utterances were falsely accepted. Most of the problems are due to an ill-formed kv-frame representation, which can be easily fixed.

5. Conclusions and Future Work

We presented a Web-based question and answering game for learning Mandarin Chinese. The game automatically generates a list of non-contradictory statements from lesson templates, together with a list of questions based on these statements. The student listens to the question, and answers by speech. The system is able to automatically judge the correctness of the answer, and assess the overall performance of the student. Evaluations on our approaches to generating questions and judging answers were performed, and the results showed that our methods were effective.

In the future, we plan to tie the game with video clips for language learning. The student watches a short video clip and then answers questions posed by our system which are based on the content of the video. This would be more attractive to the student than plain text. We would also like to carry out a carefully defined user study where before and after tests can quantify learning gains.

6. Acknowledgements

This research is supported in part by the Delta Electronics Environmental and Educational Foundation.

7. References

[1] J. H. Wolfe, "Automatic Question Generation From Text - An Aid to Independent Study," *ACM SIGCUE Outlook* , vol. 10, no. SI, pp. 104-112, Feb. 1976.

[2] H. Kunichika, A. Takeuchi, and S. Otsuki, "A Multimedia Language Learning Environment with Intelligent Tutor," in *International Conference on Computers in Education*, Taiwan, 1993.

[3] H. Kunichika, T. Katayama, T. Hirashima, and A. Takeuchi, "Automated Question Generation Methods for Intelligent English Learning Systems and its Evaluation," in *ICCE2004*, 2003.

[4] S. Seneff, "TINA: A Natural Language System for Spoken Language Applications," *Computational Linguistics*, vol. 18, no. 1, pp. 61-86, Mar. 1992.

[5] L. Baptist and S. Seneff, "Genesis-II: A Versatile System for Language Generation in Conversational System Applications," in *ICSLP*, Beijing, China, 2000, pp. 271-274.

[6] Y. Xu and S. Seneff, "Mandarin Learning Using Speech and Language Technologies: A Translation Game in the Travel Domain," in *ISCSLP*, Kunming, China, 2008, pp. 29-32.

[7] J. Glass, "A probabilistic framework for segment-based speech recognition," *Computer Speech and Language*, vol. 17, no. 2-3, pp. 137-152, Apr. 2003.

[8] Y. Xu, "Combining Linguistics and Statistics for High-Quality Limited Domain English-Chinese Machine Translation," Master's Thesis, MIT, Cambridge, Massachusetts, 2008.