A Fully-Automated Computer-Assisted Language Learning Courseware for Beginner-Level Korean Language Learners

Su-Jin Cho* and Seongsoo Lee**

* Korean Language Education Center, Sogang University, Seoul, Korea
** School of Electronic Engineering, Soongsil University, Seoul, Korea

* eklafl@chol.com
** cupid@chol.com

Abstract: - Recently, the progress of information and communication technologies enables computer-assisted language learning. Conventional computer-assisted language learning courseware hardly improves learner’s speaking skill, since it is difficult for the courseware to act as a speaking partner. In the very beginning level language learning, the number of vocabularies, grammars, idioms, and phrases are limited, and we propose a fully-automated computer-assisted language learning courseware to improve speaking skills. The proposed courseware act as a virtual human partner and learner can have conversion with it. It has pre-programmed conversation scenarios and speech database. During conversation, it recognizes learner’s speech, picks up proper response, and synthesizes artificial speech to the learner. Furthermore, the conversation scenario is based on VoiceXML programming language, and the instructor can easily make or modify conversation scenarios. It can be implemented as standalone CD-ROM title, web-based training, computer-telephony integration, and other various systems.

Key-Words: - Computer-Assisted Language Learning, Korean Language Education, Automated Courseware

1 Introduction

Recently, the progress of information and communication technologies (ICT) enables computer-assisted language learning (CALL) where computer systems support language learning without time and space limitation. Especially, CALL is known as an effective tool for foreign language learning, since it provides personalized instruction. Conventional foreign language learning is usually group-based learning in the classroom. Consequently, tutors cannot fully cope with different language understanding capability of each learner. Moreover, active learner-tutor interaction and appropriate feedback for each learner is essential in foreign language learning, but it is quite difficult for tutors to do so [1]. CALL can be an effective solution to solve these problems.

In CALL, extensive researches have been carried out to develop language learning courseware for foreign language learners. These researches successfully assist foreign language learners to improve listening, reading, and writing skills. Unfortunately, however, they fail to effectively support to improve speaking skill [2]. Most speaking coursewares just provide turn-taking materials of conventional textbook to learners so that they mostly practice pronunciation only. They act as automated cassette tapes, and lack of interactions with learners. These coursewares are just computerized supporting tools, and they don’t exploit full advantages of computer-assisted language learning. In fact, in the viewpoint of language learning, they hardly improve learners’ speaking skills.

Recently, the advent of speech recognition and natural language processing enable us to process, handle, and synthesize spoken language. Based on these technologies, we propose a fully-automated computer-assisted language learning courseware to improve speaking skills. The proposed courseware act as a virtual human partner and learner can have conversion with it. It has pre-programmed conversation scenarios and speech database. During conversation, it recognizes learner’s speech, picks up proper response, and synthesizes artificial speech to the learner. Furthermore, the conversation scenario is based on VoiceXML programming language [3], and the instructor can easily make or modify conversation scenarios.
scenarios. It can be implemented as standalone CD-ROM title, web-based training, computer-telephony integration, and other various systems.

2 Fully-Automated CALL Courseware

In the CALL courseware to practice conversation, it should “communicate” with learner, i.e. the courseware should “listen” and “understand” the learner’s speaking, and “reply” to the learner. It requires artificial intelligence (AI), but it is impossible to develop human-like AI even with state-of-the-art technologies. Instead, most CALL courseware has some pre-programmed scenarios.

In general, pre-programmed scenarios cannot fully cope with real-world conversation due to complex and abstract features of natural languages. However, for beginner-level foreign language learners, the required vocabularies, grammars, idioms, and phrases are limited to small predefined sets [3]. In Korean Proficiency Test (KPT), a nationally certified Korean language test, level 1 (beginner level) is confined within 4-word sentences and 1,000 words. This reduces the complexity of the scenario.

In addition, it is easy to make conversation scenarios when conversion situation is given such as ordering meals in the restaurant or making an appointment with friends. When conversation is confined to a specific situation, it falls to a small number of sentences and vocabularies. This further reduces the complexity of the scenario, which makes it possible to develop fully-automated computer-assisted language learning courseware [3].

Fig. 1 illustrates the architecture of the proposed CALL courseware. It consists of scenario editor, conversation scenario, speech corpus, speech recognition module, speech synthesis module, and VoiceXML (VXML) interpreter. The pre-programmed conversation scenario was written in VXML language, and can be easily modified through scenario editor. Speech corpus is a voice sample database of Korean language. Speech recognition module recognizes learner’s speaking and converts it to text words. Speech synthesis module converts text outputs of VXML interpreter into speech. Actually, it is just a text-to-speech converter, and finds the proper voice sample in the speech corpus.

3 Problem Solution

The conversation scenario is designed so that the scenario leads the conversation and the learner follows the given situation. Also it is designed to cover various types of learner’s responses. For example, when the question "내일 시간 있으면 만날래?" ("Do you have time with me tomorrow?") is given to the learner, various responses such as "내일은 시간이 없는데.", "내일은 어려운데.", and "내일은 안되는데." (both have the same meaning "I'm busy tomorrow.") are recognized as the same response. It also covers conditional responses. For example, when the question "내일 시간 있으면 만날래?" ("Do you have time with me tomorrow?") is given to the learner, positive response "응." ("Okay.") and negative response "내일은 시간이 없는데." ("I'm busy tomorrow.") are covered in the scenario.

Fig. 2 shows an example of conversation scenario. It is a pre-programmed scenario with a situation of making an appointment with a friend. When a learner speaks wrong or unexpected sentences (① in Fig. 2), the courseware suggests proper sentences and gives short explanation of grammar or phrases. When several similar expressions exist for a given situation (② in Fig. 2), the courseware expects a learner to speak one of these candidate sentences, and proceeds next stage when the learner pronounces correctly.

To develop effective scenarios for the courseware, we made some experiments for the Korean language learners in the beginner class, where all learners are foreigners. From the experimental results, we assumed 20 conversation situations including ordering meal at restaurant, receiving medical treatment at emergency room, buying necessities at drug store, and making a bank account.

We made other experiments, where above 20 situations are given to the learners and they are asked to make proper conversation with given situation. In the experiments, we analyzed their conversations, and collects frequently used vocabularies, sentences, and expressions. From the experimental results, we limited the number of vocabularies to 400 words, which is known as moderate number of vocabularies for beginner-level Korean language learner. As for the sentences, we restricted learner’s response to 3-word sentences.
Fig. 1. Courseware architecture.

Fig. 2. Example of conversation scenario.
VoiceXML Coding

VoiceXML is a markup language extended from XML. It is designed for creating audio dialogs that features synthesized speech, digitized audio, recognition of spoken and DTMF key input, recording of spoken input, telephony, and mixed initiative conversations [3]. VoiceXML minimizes client/server interactions by specifying multiple interactions per document. It shields application authors from low-level, and platform-specific details and separates user interaction code (in VoiceXML) from service logic (e.g. CGI scripts). VoiceXML is a common language for content providers, tool providers, and platform providers, and it is easy to use for simple interactions, and yet provides language features to support complex dialogs [3].

<?xml version="1.0" encoding="UTF-8"?>
<vxml version="2.0" xmlns="http://www.w3.org/2001/vxml"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.w3.org/2001/vxml
http://www.w3.org/TR/voicexml20/vxml.xsd">
<form id="Making an appointment">

<field name="step1">
<prompt>&lt;prompt&gt;여보세요.&lt;/prompt&gt;</item>
<grammar type="application/srgs+xml" root="r1" version="1.0">
<rule id="r1" scope="public">
<item>여보세요.</item>
<item>other sentences.</item>
</grammar>
<if cond="step1" == "">
&lt;grammar&gt;
<if cond="step1" == "">
여보세요라고 말해야 해. &lt;-- you should say Hello. --&gt;
</field>

<field name="step2">
<prompt>&lt;prompt&gt;전 성수인데요, 수진이 있어요?</prompt>&lt;-- This is Seongoo. Can I speak to Sujin? --&gt;
<grammar type="application/srgs+xml" root="r2" version="1.0">
<rule id="r2" scope="public">
<item>응, 나야.</item>
<item>나야, 수진이.</item>
<item>응, 난데.</item>
<item>other sentences.</item>
</grammar>
<if cond="step2" == ">
&lt;grammar&gt;
<if cond="step2" == "><item>응, 나야라고 말해야 해. &lt;-- you should say Yes, speaking. --&gt;
</field>
</field>

<field name="step3">
<prompt>&lt;prompt&gt;내일 시간 있으면 만날래</prompt>&lt;-- Do you have time with me tomorrow? --&gt;
<grammar type="application/srgs+xml" root="r3" version="1.0">
<rule id="r3" scope="public">
<item>응.</item>
<item>좋아.</item>
<item>그래.</item>
<item>내일은 시간이 없는데.</item>
<item>내일은 어려운데.</item>
<item>내일은 안되는데.</item>
<item>other sentences.</item>
</grammar>
<if cond="step2" == ">
&lt;grammar&gt;
<if cond="step2" == "><item>응 또는 내일은 안되는데라고 말해야 해. &lt;-- you should say okay or I'm busy tomorrow. --&gt;
</field>
</field>

<field name="step4">
<prompt>&lt;prompt&gt;�&lt;prompt&gt;안녕이라고 말해야 해. &lt;-- you should say Bye. --&gt;
</field>
</form>
</vxml>

Fig. 3. VoiceXML example of conversation scenario.
VoiceXML provides keyword extraction that is useful for conversation practice. For example, when the learner says "응, 내일은 바쁘지 않아." ("Okay, I’m free tomorrow."), VoiceXML recognizes this response as "응." ("Okay.") The key extraction grammar is defined and described in the `<grammar>` ...
`</grammar>` expression. Fig. 3 is an example of VoiceXML coding that corresponds to the conversation scenario example in Fig. 2. `<field>` defines input variables and its processing. `<prompt>` sends audio clip, waits for response, and gets proper input variables. `<grammar>` and `<rule>` check the keywords of input sentences. `<item>` defines proper keywords of input sentences. `<if>` and `<elseif>` are used for conditional control flow.

5 Current Status and Future Work
We finished the experiments to collect conversations of beginner-level learners, and made 2 conversation situations with 50 words vocabularies. We also finished the scenarios written in VoiceXML. The courseware exploits only several functions of VoiceXML. Scenario editor with VoiceXML grammar checking was also implemented. Now, we are implementing very simplified VoiceXML interpreter with limited functionalities. For the speech recognition module and the speech synthesis module, our collaborative research group in the field of speech processing is modifying their speech processing software to fit to the courseware. We are expecting to implement a simple demo program within a year.

References: