Example-based Dialog Modeling for English Conversation Tutoring

Sungjin Lee **Cheongjae Lee** **Gary Geunbae Lee**

Department of Computer Science and Engineering
Pohang University of Science and Technology,
South Korea
junion@postech.ac.kr lcj80@postech.ac.kr gblee@postech.ac.kr

Abstract

In this paper, we present an Example-based Dialogue System for English conversation tutoring. It aims to provide intelligent one-to-one English conversation tutoring instead of old fashioned language education with static multimedia materials. This system can understand poor expressions of students and it enables green hands to engage in a dialogue in spite of their poor linguistic ability, which gives students interesting motivation to learn a foreign language. And this system also has educational functionalities to improve the linguistic ability. To achieve these goals, we have developed a statistical natural language understanding module for understanding poor expressions and an example-based dialogue manager with high domain scalability and several effective tutoring methods.

1 Introduction

In today's age of globalization, international exchanges and businesses are increasing rapidly and it makes English more important as a world language. In the past, language learning was just memorizing words and grammar rules but it failed to enhance the more important ability of conversation. Therefore, for last two decades, pedagogical devices using multimedia have been developed to improve the conversational ability. But it was also turned out to be less effective than being expected. Now, it is widely agreed among educators that the best way to learn to speak a foreign language is to engage in natural conversation with a native speaker of the language (Kim et. al., 2000). Yet this is also one of the most costly ways to teach a language, due to the inherent high demand of one-to-one student-teacher interaction that it implies. Recent research on the spoken dialogue system shows the possibility for a computer to have a conversation with human.

So far, most of spoken dialogue systems for language learning have been focusing on pronunciation practice (Mayfield et. al., 2000). In those systems, students should say a sequence of fixed sentences. This limitation comes mainly from the difficulties of understanding various user utterances and dialogue modeling of diverse scenarios. In addition, it's more difficult to understand utterances of students because of its vocabulary errors and grammar errors. However, if the system halts whenever a user utterance contains an error, students would be stressed out. So, we consider the following three desired requirements that a spoken dialogue system for English learning will have: First, the system is able to understand student's poor and non-native expressions. Second, the system allows dialogue modeling of high domain scalability to support various practical scenarios such as immigrant, transportation, restaurants and so on. Finally, the system should provide educational functionalities which help students improve their linguistic ability and successfully complete the dialogue.

2 System Architecture

An overview of the English tutoring dialog system is shown in Fig 1. It generally follows a common spoken dialogue system design (Seneff et. al., 1998). The ASR (automatic speech recognition) module is operated first, which recognizes the user utterance. The recognized result is used for the input of the SLU (spoken language understanding). The SLU module extracts semantic concepts from the user utterance and constructs the predefined semantic frame. The DM (dialog manager) generates system responses with the semantic frame and the discourse history. The discourse history is a set of semantic frames in one dialogue session. The result of the dialogue manager is represented by a system action tag. The NLG (natural language generation) produces literal system utterances corresponding to a given

system action tag. Finally, the TTS (text-to-speech) synthesizes the spoken utterance from the literal utterance.

The DM module is designed by adopting an object-oriented approach for cross-domain and mixed initiative DM (I. O'Neil et. al., 2005). Generic dialogue behavior is separated from domain specific behavior. Each specific expert inherits generic dialogue behavior and possibly refines it and adds its own domain specific behavior. In our system, there exist a number of domain experts for various themes to learn (e.g., ImmigrantExpert, TransportExpert and so on) and a supervisor expert which manages overall language learning process.

When user gets started, DM activates SupervisorExpert first. SupervisorExpert validates the user's identity to load the user profile from the user database and illustrates about available subjects to practice. After a subject is chosen, DM deactivates SupervisorExpert and activates the selected subject expert. While users have a conversation with the selected expert, the expert collects various assessment data from the on-going dialogue. When the dialogue ends, domain expert sends the assessment data to SupervisorExpert and DM activates SupervisorExpert again. Finally, SupervisorExpert gives feedback to users based on the assessment data.

3 Robust Spoken Language Understanding

The goal of spoken language understanding is to extract meanings from natural language speech and infer the speaker's intention. To understand the human language, most SLU systems define a semantic frame, i.e., a formal structure of predicted meanings. The task of SLU is to map natural language speech into a semantic frame.

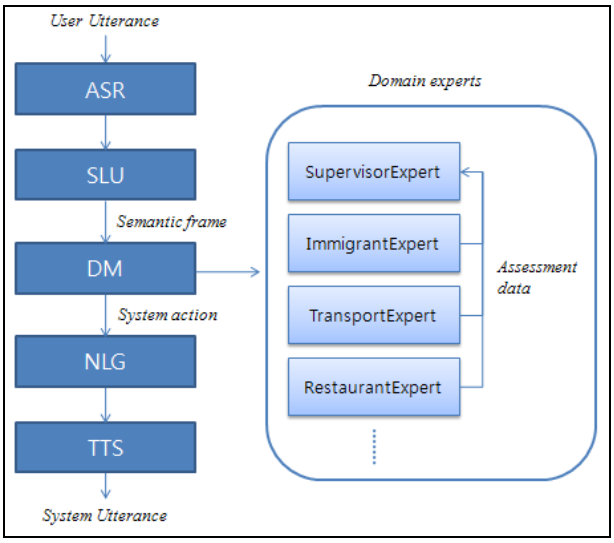

Fig. 1 Overview of the system architecture

The methods of SLU can be generally divided into rule-based methods and statistical methods. The rule-based methods are better for applications which have well-formed input utterances and provide a small number of service domains so that accurate and plentiful linguistic results are obtained. However, statistical methods are good for systems that have to handle some input errors and support many number of different domains. Since a spoken dialogue system for English learning needs to handle poor expressions of students and provide various diverse themes to learn, statistical methods are more appropriate. Furthermore, since our system is for training a conversation not for learning words and grammar rules, error-robustness from statistical methods is more preferable than rich syntactic information from rule-based systems.

A brief model of our statistical SLU is shown in Fig. 2. Our approach attempts to understand spoken language by extracting essential factors for pre-defined slots. This approach addresses the problem of SLU as three levels of understanding problems: Dialog Act, Main Action, and Component Slot. A Dialog Act and Main Action present the meaning of an utterance at the discourse level, and it is approximately the equivalent of intent or subject slot in a practical dialog system. The Dialog Act is a domain independent and surface-level concept, but the Main Action is a domain-specific and functional-level concept. A Component Slot is a generalized identifier of a named entity such as a person, location, organization, or time. In the SLU problem, we itemize the Component Slots as the domain-specific semantic meanings of words. Table 1 shows an example of our representation scheme for Immigrant domain. We solve the tasks on Dialog Act and Main Action slots as classification tasks. The value of a Dialog Act slot is assigned from one of the classes which designate the surface-level speech acts, such as *yn_question*, *wh_question*, *request*, *statement* and so on for each sentence. Similar to the value assignment of the Dialog Act slot, the value of the Main Action slot is assigned from one of the classes of the main application actions in a specific domain, for instance, in the Immigrant domain, such as *Tell_Origin*, *Tell_Period*, *Tell_Job*, *Tell_Purpose* and so on. The tasks for the Component Slots, such as, *name*, *period*, and *job* are solved as named entity recognition tasks.

I will stay for three weeks
Dialog Act = Statement
Main Action = Tell_Period
PERIOD = three weeks

Table 1 An example of a semantic frame

When a user utterance enters, CRF (Conditional Random Field, sequence labeling model) (Lafferty et. al.,

2001) extracts component slots first and then MaxEnt (Maximum Entropy classifier) (Ratnaparkhi, 1998) classifies the Dialog Act/Main Action from the utterance. The detail probabilistic models were designed following the previous works (Lee et. al., 2007)

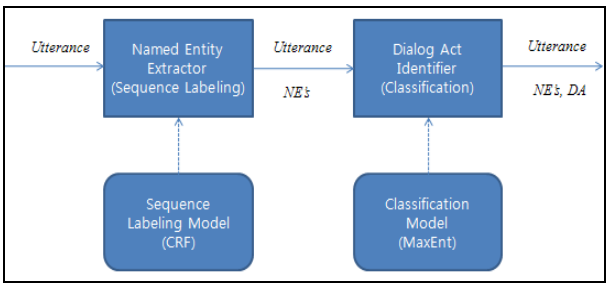

Fig. 2 Block diagram of SLU

4 Example-based Dialogue Manager

Dialogue Manager is a central component in a spoken dialogue system and controls a dialogue flow. It generates system utterances corresponding to the user's utterances and it accesses to external knowledge sources to provide relevant information. To develop a new spoken dialogue system, developers should decide what kind of dialog modeling methodology to apply based on their application. Generally, methodologies of a dialogue management are classified as system-initiative, user-initiative or mixed-initiative with respect to who directs the dialogue. Alongside these distinctions, we can distinguish between methods for representing and implementing the flow of the dialogue as Finite State Network-based, Frame-based or Agent-based with respect to dialogue control strategy (McTear, 2004).

For our system in which dialogue scenarios are known in advance, it is appropriate to model these dialogues as system-initiative. Most system-initiative dialogue managements use finite state network and carefully design system prompts to restrict user's input utterances to simple words or phrases. For language learning, however, it is important for students to use fluent sentences not just words or phrases as input. Therefore system-initiative and natural user responses should be considered together. This kind of system is often referred to "limited mixed-initiative" (McTear,2004). And the best way of implementing the limited mixed-initiative dialogues is a framed-based method. Normally, frame-based systems use scripts or rules as a dialogue control algorithm, but in this system, we use an example-based dialogue modeling technique (Lee et. al., 2006) which provides high domain scalability. While in a rule-based system a set of rules should be crafted by hands for a new domain, the example-based system automatically builds up a dialogue model for a new domain from dialogue examples.

The brief process of the example-based system is drawn in Fig. 3. The dialogue example database (DB) is automatically constructed by tagging dialogue situations and system responses to the dialogue example corpus and by storing them in a relational database in which dialogue situation plays a role as index keys. We define the dialogue situation as semantic frames from SLU plus Discourse History Vector which represents what information has been known from users to current time as a binary bit vector.

In the execution time, DM combines a semantic frame from SLU and the discourse history into a search key (user query) of the dialogue example database. If there exist multiple retrieval results, we measure the utterance similarity to find a best match. Utterance similarity is defined as a linear interpolation of discourse history similarity and lexico-semantic similarity. Discourse history similarity is a cosine measure of the discourse history vector of the on-going dialogue and that of the retrieved example dialogue. Lexico-semantic similarity reflects how much the user utterance is similar to the retrieved example utterance. To compare these two utterances we should carefully handle the named entity parts. Even though named entities of the two utterances are different it should not lower the degree of similarity. So, we replace named entities with their meaning before measuring the lexical similarity. Sometimes it is possible to find no exact examples. When this happens, we try to perform a partial-match with a semantic frame only because the dialogue situation is more dependent on the user's intention contained in the semantic frame. Despite a partial-match, if there exists no example, we apply meta-rules as the last resort. The meta-rules handle various no example match cases (Lee et. al., 2006)

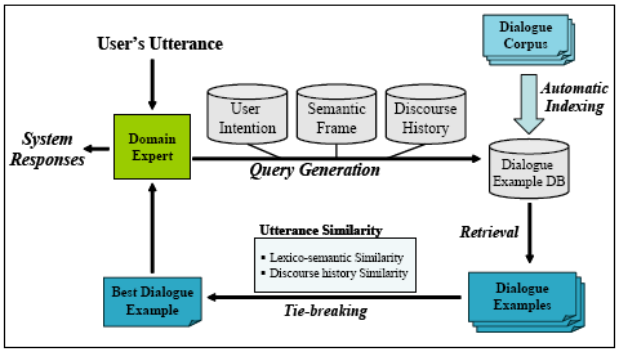

Fig. 3 Example-based Dialogue Modeling Strategy

5 Tutoring and Assessment Functionality

The robust SLU and flexible DM mentioned previous sections provide students with an experimental environment in which students have a goal-oriented conversation with a virtual native speaker. As a language tutor-

ing system, we add several educational functionalities such as implicit tutoring, explicit tutoring and assessment feedback. If the system helps students express what they want to say when they have a problem in a conversation, students will rapidly learn expressions proper to those situations.

Usually, students do not produce grammatically sound and complete sentences. But as long as their intentions are understandable, the system displays an example sentence on the screen as implicit tutoring. By implicit tutoring, users learn better expression without stopping a conversation because frequent halting makes them lose their interests. Fig. 4 shows the block diagram of the implicit tutoring. The user utterance is first checked whether it is appropriate to the current situation or not. Since empty retrieved result from dialogue example database means that the user utterance is not understandable, the system asks the user to repeat. Otherwise, it means at least the user intention is proper. Then the system queries the tutoring example database with user's intention to get better example sentences. If multiple example results are retrieved we determine the best example based on the lexical similarity.

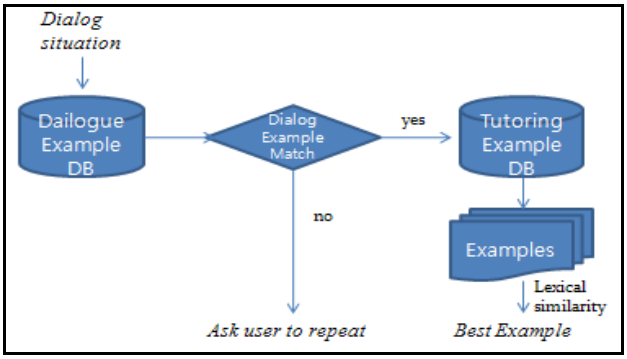

Fig. 4 Block Diagram of Implicit Tutoring

Explicit tutoring, as the name explains, helps students when they explicitly ask the system to help. Sometimes, students have no idea about what to say and they cannot continue the dialog. In such a case, students can ask the system explicitly to give a hint. Then the system searches the tutoring example database with the expected user action, which in turn can be obtained from dialogue example database, and shows an example sentence as a hint on the screen so that students can use it to keep a conversation. It's also possible to give a hint in the form of mother language so that students can practice translations in accordance with the user level. Fig. 5 shows the block diagram of the explicit tutoring. When a user utterance enters, the system examines whether it is an explicit help request or not. If it is an explicit help, the system searches the tutoring example database to find the appropriate user utterance to the previous system utterance. Among several retrieved examples, we select one of them through random selection. Students refer to the example and can continue the dialogue.

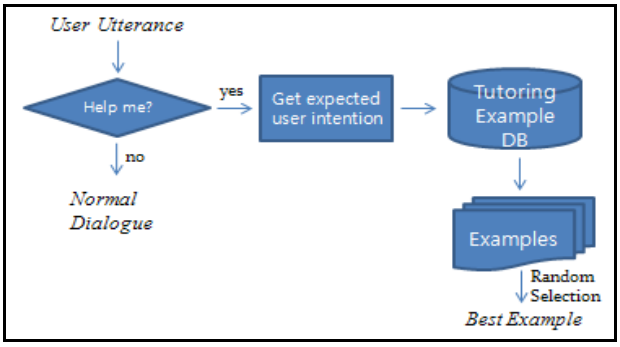

Fig. 5 Block diagram of Explicit Tutoring

At the end of the dialogue the system provides assessment feedback which includes total score and specific messages that inform students what part they need to improve. The total score is sum of pre-defined measurements such as mission completion level, elapsed time, utterance suitability, help frequency, and filler word frequency. These assessment data are gathered by domain experts during dialogues and lastly SupervisorExpert calculates the total score and provides feedback messages on a basis of the assessment data.

6 Implementation and Running Examples

The English tutoring conversational system was fully implemented with example-based dialog management techniques. The implementation was done using C++ on Windows XP with a 2.4 GHz dual core processor and 2.0 GB of main memory. The following scenario, and the corresponding inputs and outputs, gives a flavor of the system in real action. A brief commentary, indicating which main features of the DM are being called into play, is given at key points in the dialogue and is printed in italics.

The dialogs with the supervisor expert is conducted in mother tongue (Korean), but they are translated into English for readers in this paper.

S1: Hi. This is a dialogue system for English learning. May I ask your name for authentication?

DM activates SupervisorExpert (SE) which greets students and authenticates them

U1: My name is Hong Kil Dong

SE authenticates the user and loads the user's profile

S2: You are a new user. We have four themes such as immigrant, transportation, hotel, restaurant theme. Which theme do you want to learn?
U2: I'd like the immigrant theme
S3: The immigrant theme is selected. Let's get started

SE notifies DM that ImmigrantExpert (IE) is the one student selects. Then DM deactivates SE and activates IE.

S4: Hello

IE starts a dialogue with greeting.

U3: Hi
S5: Your passport please
U4: Here this
On screen: *[Example] Here you are*

[Example] is an example utterance provided by implicit tutoring. "Here this", the student's utterance, is understandable based on context but it is not a good expression. The system recommends a better expression so that the student becomes better in a next dialogue.

S6: What's your nationality?
U5: I am a Korean
On screen: *[Example] I come from Korea*

Even when the user utterance is correct the system shows another possible expression by implicit tutoring.

S7: What's the purpose of your visit?
U6: Travel
On screen: *[Example] I am here on business*

If a user input is simple words or a phrase, then the system shows a complete sentence.

S8: How long are you staying?
U7: I stay for three weeks
On screen: *[Example] I will stay for three weeks*

The user utterance is semantically understandable but it is grammatically wrong. The system shows a grammatically correct sentence on screen which is obtained by searching the tutoring example database using the semantic frame from SLU.

S9: Anything to declare?
U8: Pardon me?
S10: Anything to declare?
U9: Nothing
On screen: *[Example] Just my personal things*
S11: What kind of job?
U10: Help me
On screen: *[Hint] I am a student*

[Hint] is an example utterance provided by explicit tutoring. The system knows that it asked the user about a job in the previous question. And it also knows that it already got information about nationality, purpose of visit and period through the dialog history. Therefore by using such dialogue history the system can find out a proper example from the tutoring example database.

U11: I am a computer programmer
S12: Ok, no problem
U12: Thank you. Bye

At the end of the dialogue, IE sends assessment data to SE and notifies DM that the immigrant theme is over. DM deactivates IE and activates SE again.

S13: The immigrant theme has been done. Your total score is B. In next time, you better use more comprehensive and complete expressions, and try to say more quickly. Would you like to continue learning another theme?

SE provides assessment feedback which includes total score and specific messages that inform the student which part they need to improve.

U13: No.
S14: Study session ends.

7 Conclusion

We developed an intelligent and experimental dialogue system for English conversation tutoring by integrating an error-robust statistical SLU and an example-based dialogue manager with high domain scalability and diverse educational functionalities. It provides students with active, personal and interesting learning environment contrary to the previous passive and boring educational environment. Future development plans include more extension of robust non-native speaker SLU, which uses common errors of non-native speakers, such as deletion, shift, spelling, auxiliary, article, and preposition errors, as features of the statistical model.

Acknowledgement

This work was a part of the collaborative research project, KT VR Lab@POSTECH, between POSTECH and KT.

References

In-Seok Kim, Inn-Chull Choi, "A Study of Software Contents and Teaching Techniques for Developing an Intelligent Learning Program for English Listening and Speaking Skills," *Multimedia-Assisted Language Learning, 4(1),* 2000, pp. 167-223

J. Lafferty, A. McCallum, and F. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," *Proc. Int'l Conf. Machine Learning*, 2001

C. Lee, S. Jung, J. Eun, M. Jeong and G. Lee, "A Situation-based Dialogue Management Using Dialogue Examples" *in Proceedings of the international conference on acoustics, speech and signal processing 2006, vol. 1,* 2006, pp. 69-72

Changki Lee, Jihyun Eun, Minwoo Jeong, Gary Geunbae Lee, YiGyu Hwang, Myung-Gil Jang, "A multi-strategic concept spotting approach for robust spoken Korean understanding,". *ETRI journal, vol 29, no 2*. 2007, pp178-188

Michael F McTear, "Spoken dialogue technology: toward the conversational user interface," *Springer Verlag: London* 2004

L. Mayfield Tomokiyo, L. Wang, and M. Eskenazi. "An empirical study of the effectiveness of speech-recognition-based pronunciation training," *In Proceedings of the 6th ICSLP Volume 1,* 2000. pp. 677–680

I. O'Neil, P. Hanna, X. Liu, D. Greer, and M. McTear, "Implementing advanced spoken dialogue management in java," *Speech Communication*, vol. 54, no. 1, 2005, pp. 99-124

A. Ratnaparkhi, "*Maximum Entropy Models for Natural Language Ambiguity Resolution*," PhD thesis, University of Pennsylvania, 1998

S. Seneff, E. Hurley, R. Lau, C. Pao, P. Schmid, and V. Zue, "Galaxy-II: A reference architecture for conversational system development," In *The Proceedings of ICSLP*, 1998, pp. 931-934