2012 International Conference on Innovation and Information Management (ICIIM 2012)
IPCSIT vol. 36 (2012) © (2012) IACSIT Press, Singapore

Textual Conversation Agent for Enhancing Attraction in E-Learning

Xuetao Mao[1, a], Jean-Paul Sansonnet [2, b] and Lin Li [1,c 1 +]

[1] Department of Computer Science and Technology, Wuhan University of Technology, Wuhan, China
[2] LIMSI-CNRS, BP 133, F-91403 Orsay Cedex, Paris, France

Abstract. E-Learning is one of the most pervasive applications of computer technique. Usually, the application presents textual knowledge through web-browser, others are implemented by real-time video broadcasting. Two methods are either static or bandwidth consuming, and both are monodirectional. New tools are needed with two main requirements: interactivity and durability. In this context, Conversational Agents are a promising approach for improving the experience between the learners and the application. A Conversational Agent embedded E-Learning application will express knowledge rely on the rational reasoning over the learner's utterances in stead of straightly displaying predefined context. In this paper, we described a framework of the tutoring conversational agent. Under the framework, we provide a practical method to rapid create new tutoring applications embedded with Conversational Agents. It has proved that the interactive mode of E-Learning system is more attractive for learners.

Keywords: E-Learning, Agent, Conversational Agent, NLP

1. Introduction

Recently, E-Learning or distance education is widely introduced in all-level educations[1].One common difficulty of E-Learning is to improve the effectiveness of study. The traditional applications are designed to display predefined content to the learners continuously no matter in textual or multimedia form[2]. But the monodirectional E-Learning model is weak to help learners concentrating on the screen because of the lack of interaction. Responsive E-Learning environment then is valued. as example the PEACOCK(Performance based E-Learning Adaptive Cost-efficient Open Corpus framework), which facilitate an intellectual assortment and distant delivery of personalized the media content to learners in order to make it the closest to perfection of learning objectives and interests, desirable media content formatting and delivery costs[3]. Some other researches exploit IM tools like MSN Messenger and tutoring agent to organize the on-line study [4]. The IM tool is used as an user-application interface and the tutoring agent as a complement to provide help during the collaborative learning. A problem of this kind of system is that popular IM tools often distract learner's attention from their real aims. Therefore, a human instructor is often needed to organize the learning process. That is why the face-to-face teaching and learning cannot be substituted. However, a tutoring agent can work imitating a teacher in condition that: 1) the agent can express organized knowledge decided by teacher; 2) the agent can be responsive. In this paper, we introduce TCA(Textual Conversational Agent) as the tutoring agent into E-Learning applications. Firstly, we give a definition of TCA and consider how it combined with E-Learning applications. Secondly, we describe the framework of Conversational Agent in E-Learning. Technique details on building a Knowledge Base of TCA are given as well.

2. TCA Embedded E-Learning System

+ Corresponding author
E-mail address: [a] maoxuetao@whut.edu.cn, [b] jps@limsi.fr, [c]cathylilin@whut.edu.cn

TCA stands for Textual Conversational Agent which recognizes only text inputs and yield text outputs as well. Distinguish from others; TCA is one of the simplest forms of language involved agents.

2.1. Conversational Agent

Conversational Agent(CA) is a communication method that exploit computational linguistic technologies to engage learners in text-based information-seeking and task-oriented dialogs for a broad range of applications[5]. CAs are often embedded in web-based applications sometimes with a virtual avatar that are called by some researchers Embodied Conversational Agents(ECAs)[6]. Embodied Conversational Agents provide more naturalness than text-only ones for they employ speech, gesture, face expression etc. to assist in a conversational session. On the technique layer, there are two requirements of building a Conversational Agent :

2.2. NLP(Natural Language Processing) Technology

To imitate a face-to-face conversation, the conversational agent has to first receive learner's words, interpret them, then reasoning to the application logic for response contents such as searching in a database, and perform some actions carrying the response information. This is a DIVA NLP conversational chain[7].

2.3. Application Integration Technology

A conversational agent must meet the requirements of different applications. For instance, an e-commerce application should be reliable, secure and effective while an entertainment system should be expressive, friendly and interesting.For E-Learning application, Conversational Agents are expected to act as tutors that both guarantee learners concentrate in a class and conduct them to interact with the agent.

2.4. Agents in the E-Learning System

It's hard to go into NLP domain to deal with all kinds of natural language input for any CA system. But when integrated with E-Learning application, we can focus only on a particular teaching course. That is the size of the knowledge base(KB) of the CA is limited in a small world language model. We define in this framework three kinds of roles with a tutoring CA:

-STU stands for student. STUs are deliberate agents that always begin a conversation session.

-TEA xitongfor teacher. TEAs interact with STUs through an asynchronous mode by creating the KBs.

-CA stands for conversational agent. CAs are always responsive. When a CA pretends to deliberately begin a conversation session, it in fact reacts indirectly in response to certain other agent's requests.

Embed a CA in an E-Learning system brings at least two advances: 1) more attractive; 2) customizable. Figure 1 shows some examples executed in browser. One can chat to know the agent by input English words in the textbox:

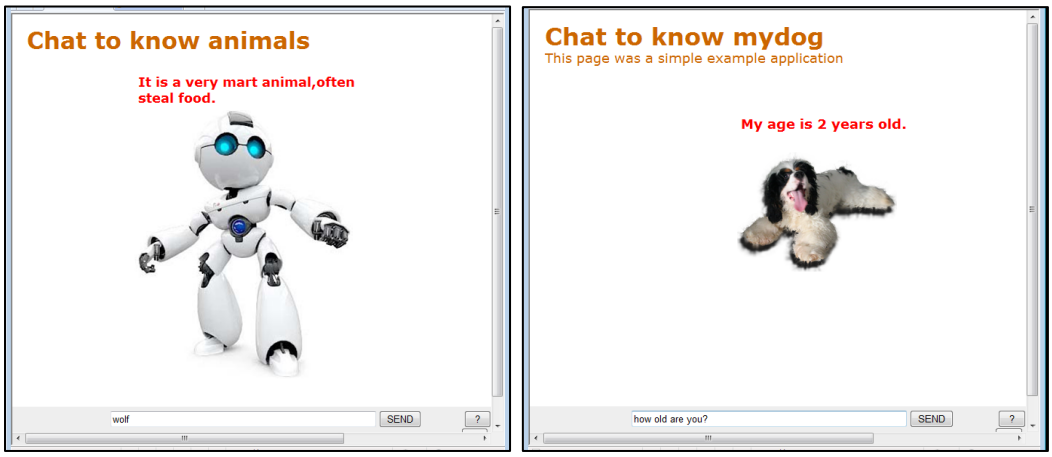

Fig.1 Agent embedded E-Learning Applications

3. Framework of Tutoring TCA

A framework of tutoring TCA includes 4 phases showing in Figure 2. Phase 1 is input analysis which made of formalization and interpretation modules； Phase 2 is knowledge base creation which made of formalization and representation modules; Phase 3 is semantic matching which made of a matching module and Phase 4 is a language generation module.

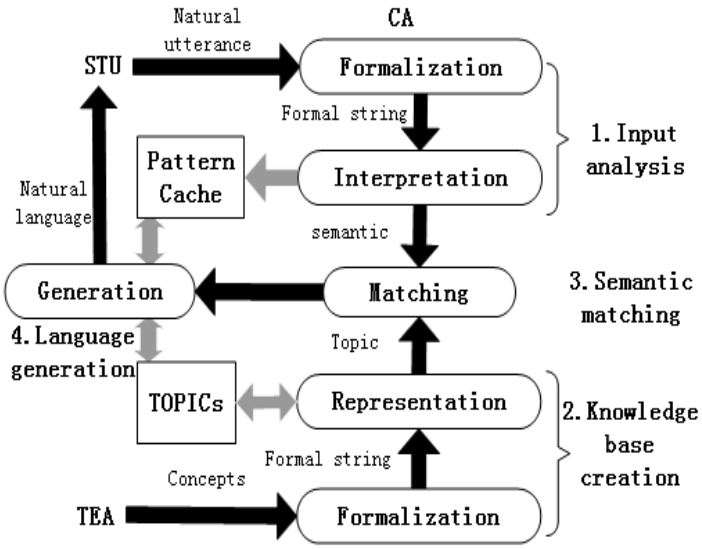

Fig.2 Framework of TCA

3.1. Formalization

Since human inputs are deliberate, what they would input are hard to predict. Before interpretation, we initially formalize the inputs to filter out some noises such as symbols and unify the capital and small letters. For example, characters as ", . / \ | [ ] { } ( ) : ; - _ ~ # $ ^" and consecutive spaces are replaced by a solo space; "&" is replaced by "and"; quotation marks are escaped by "#". All letters are then changed to lowercase. The replace protocol is generic but not strict that can be modified according to certain applications.

3.2. Interpretation

The first phase is to interpret natural requests into a formal request form (FRF) by through some filter rules. The formalized strings output by formalization module are sent to an Interpretation module in order to:

3.3. Extract the Semantic of the Strings

We argue that semantic of an learner input is associated with some key words divided into: generic semantic (GS) and specific semantic (SS), and then the semantic can be marked as SEM (GS, SS).

3.4. Cache the Structure Patterns of the Input Strings

The conversational agent will respond to the learner input with natural language utterance which consists of two parts: a) the acknowledgeable content and b) the expressive structure. Based on the result of the semantic extraction, the generic semantic can be seen as a structure pattern because it can generate the skeleton of the response expressions beforehand. For example, "(*QUEST WHAT THEMEANING) 1"* can generate "*The meaning of 1 is 2".* Fillers at position 1 and 2 are SSs that will be selected from the knowledge base at runtime.

The semantic extraction is implemented by a series of filter rules in order to separate the two kinds of key words. For example: "*what's the meaning of #ftp# ?*"→"*QUEST WHAT THEMEANING #ftp#*".

The uppercase part is generic semantic expression while the lowercase part is specific semantic expression which is related to the practical application.

3.5. Semantic Space

For our recent consideration, we cascade three layers of filter rules to construct the *Semantic Space[8]*.The three layers from top to bottom are:

1) Generic rules. The generic rules deal with general linguistic phenomena e.g.

<rule id="323" pat="<(.) (know|knew|known|experienced?|go BYMEANSOF|went BYMEANSOF|gone BYMEANSOF|lived? BYMEANSOF) (.*)>" go="NEXTRULE">*

<filter>

[1, "TOKNOW", 3]

</filter>

</rule>

In this example, strings like "know, knew, known, experienced"etc. are translated into English-like string "TOKNOW".

2) Special rules. The special rules deal with application-related ad hoc words, for example:

<rule id="love" pat=" (THEUSER TOLIKE THEAVATAR)|(QUEST (.)TOLIKE (THEUSER|THEAVATAR)) ">*

<do>

if (THETOPIC.coop>=10) {TALK_say(["yes, I love you!", "I love you very much."], 0, 4); } else if (THETOPIC.coop>=5) {TALK_say(["maybe I will love you too", "I may love you", "ok, I may consider that"], 0, 4); } .....

else {TALK_say(["I don't like you!", "I hate you!", "you are very disgusting"], 0, 4); }

</do></rule>

Special rules will be hard coded for special words. But we can divide this layer into more sub-layers to provide relatively generic rules before more particular ones. For example, to provide rule: *space.person* before rule: *space.person.dorian.* The latter is less generic then the former.

3) Reaction rules. The reaction rules generate generic reactions to the input semantic according to the result of the interpretation that described before. For example:

*<rule id="type" pat=" THETYPE|(WHAT .*TOBE THEAVATAR(?! .+)) ">*

*<say> I am a _THETOPIC.subtype_. *

* I am a kind of _THETOPIC.type_. *

* Everyone said that I am _THETOPIC.evaluation_. *

</say></rule>

The variable like _THETOPIC.subtype_ comes from the selection in the knowledge base.

3.6. Semantic Key Dictionary

The uppercase English-like strings (e.g. TOBE, THETYPE) in the above examples are called *Semantic Keys* (SKs). They are extract result from the translation of the word lemmas into their associated semantic classes defined by analysis on some corpora [9]. Each SK is represented by an unique keyword with its prefix. Table 1 shows some examples:

Table 1 Excerpt from the SK Dictionary

SK Example	Gloss
TOKNOW	Denotes the mental action of knowing something
TORESOLVE	Denotes the mental action of resolving a problem
THESPEECH	Denotes the activity supported by the action of speaking
THEAVATAR	Denotes the graphical/dialogical assisting character of the application
ISFEMALE	Denotes the quality of a person with gender:female
ISPATIENT	Denotes the quality of somebody who is patient

AKOCURRENCY Denotes a kind of: currency

AKOJOB Denotes a kind of: specific job

QUEST Denotes the grammatical relation:question

NEG Denotes the grammatical relation:negation

3.7. Representation

The function of Phase 2 is for teachers to rapid create the knowledge base of the conversational agent. Like in Input analysis phase, the inputs are formalized at first. Then in a *Representation* module, the knowledge segments are recorded with a uniform structure called *TOPIC*. The knowledge base of the conversational agent is made of many autonomous *TOPICs* that created based on predefined templates of which the ontology is defined at first.

1) TOPIC Ontology. Like the concept of Class in OOP (Object Oriented Programming), the concrete concept in the ontology of TOPIC template is the *TOPIC* i.e. the learning course, it contains some attributes such as type, subtype, name, language, history and future. These attributes are inherited by descendent concepts such as *THING*(the physical entity) and *STRUCTURE*(the abstract entity). Eath sub-concept contains attributes and their descendents.

2) Example of a TOPIC. A typical example of TOPIC is given below(simplified from the TOPIC of the author M.Xuetao):

<topic id="TOPICMXT">

*<!-- ***** 0. TOPIC ********** -->*

<type gloss="the leaves in this ontology = {OBJECT, PERSON, ANIMAL, INSTITUTION, SERVICE, APPLET}">

PERSON</type>...

*<!-- ********** 1. THING ********** -->*

<size gloss="abstract concept of the size = big, small, tiny, huge...">

tall</size>.....

*<!-- *******1.2 PERSON = Joher, Adam&Eve, AskAge ******-->*

<gender gloss="sex = male female">male</gender></topic>

This TOPIC is created based on the PERSON template and added some specific attributes like *greetings*. Each attributes are given a gloss for the creator. Based on a template, people can create formal TOPICs. In year 2008, we have created 7 templates include *person, animal, applet, artwork, institution, object* and *service* for experiment purpose.

3.8. Reasoning Functionality

Each round of the input and responsive output is based on the reasoning functionality which is in fact a matching process: Search the SS ingredient of the SEM(GS, SS) in the TOPIC. This will not be a problem because the TOPIC files are lightweight textual document. If there is a match with the attribute in the TOPIC, we assign its value to a functional variable marked as *TOPIC.x* in which *x* is the attribute name. Else if nothing matched in the TOPIC, the reasoning functionality will select and set the value of another attribute to the temporary e.g. *TOPIC.temp*. At the other hand, one layer of the semantic space we discussed in section 3.1.2 provides application-relevant keywords. If the SS ingredient also matches these keywords, the corresponding information will also reflect in the reactions. Thus the reasoning functionality include three parts: 1) reasoning the static knowledge, the TOPIC, 2) reasoning to the dynamic knowledge, the Specific Layer of the Semantic Space and 3) the Generic Layer of the Semantic Space based on common NLP result.

3.9. Generation

At last, the generation module is relatively simple. When the previous rules filtered and translated user's input into one of the formal request forms(FRF), rules in this layer handle the FRF, and then attach a answering set. The answers in the set are natural language sentences formed by combining the *TOPIC.x* or *TOPIC.temp* with the generic reaction sentence structure. For example, an answer like: "my name is _TOPIC.name_" will works.

4. Summary

To embed a Conversational Agent in an E-Learning system can be more attracting and customizable compare to traditional monodirectional mode. We don't want to dive into a universal NLP domain of dialog system because there is both generic and specific level in the interactive E-Learning. Based on corpus result from experiments, we propose a framework for both runtime language interpretation and develop-time knowledge representation. It is has shown effectively in our teaching work for undergraduates who use TCAs to remember about 10% more English words than those without TCAs. In recent, the generic layer of the semantic space is rather simple. If we can update it to a Formal Request Language (FRL) [10], the conversational agent will be seen more human-like in the future.

5. Acknowledgement

This paper is Supported by "The fundamental Research Funds for the Central Universities".

6. References

[1] T. Zahariadis and S. Voliotis: New Trends in Distance Learning Utilizing Next Generation Multimedia Networks[J]. *Kluwer Educational and Information Technologies*, 2003, Vol.8, No.1, P67-81.

[2] D. E. Douglas and G. Van Der Vyver,: Effectiveness of E-Learning course materials for learning database management systems:An experimental investigation[J].*Journal of Computer Information Systems*, 2004, 44(4), P41–48.

[3] J. H. Yousif, D. K. Saini, H. S. Uraibi: Artificial Intelligence in E-Leaning-Pedagogical and Cognitive Aspects:Proceedings of the World Congress on Engineering 2011 Vol II WCE 2011, July 6 - 8, 2011, London, U.K.

[4] S. C. Hsu, M. Y. Day, S. H. Wu: Wing-Kwong Wong and Wen-Lian Hsu: Designing a Tutoring Agent for Facilitating Collaborative Learning with Instant Messaging[M]. Intelligent Tutoring Systems - 8th International Conference, ITS 2006, Proceedings, P689-691.

[5] J. Lester, K. Branting and B. Mott: Conversational Agents[M]. The Practical Handbook of Internet Computing, Chapman & Hall, 2004.

[6] J. Cassell, J. Sullivan, S. Prevost and E. Chuchill, Embodied Conversational Agents[J]. MIT Press, Cambridge, MA (2000), P67-83.

[7] X.T. M., J-P. Sansonnet and F. Bouchet: A Corpus-Based NLP-Chain for a Web-Based Assisting Conversational Agent, Third Workshop on Animated Conversational Agents, WACA'2008, Paris, nov 2008

[8] X.T. M., F. Bouchet and J- P Sansonnet : An ACA-based Semantic Space for Processing Domain Knowledge in the Assistance Context. Third International Conference on New Trends in Information and Service Science, NISS 2009, June 30-July 2 2009, Beijing, China.

[9] X.T. M., F. Bouchetand J-P Sansonnet: Intelligent Assisting Conversational Agents Viewed through Novice Users' Rrequests, International Conference Interfaces and Human Computer Interaction, IHCI 2009, June 20-22 2009, Algarve, Portugal.

[10] F. Bouchet, J-P Sansonnet: Tree Kernel and Feature Vector Methods for Formal Semantic Requests Classification[M] Poster presentation, International Conference on Machine Learning and Data Mining MLDM´2009, july 23-25 2009, Leipzig, Germany.