YaoTalk: A Conversational System for the IIIS Domain *

Tianlin Shi †
Institute for Interdisciplinary Information Sciences
Tsinghua University, Beijing
stl501@gmail.com

Sheng Yang
Institute for Interdisciplinary Information Sciences
Tsinghua University, Beijing
yangsheng6810@gmail.com

ABSTRACT
Information about the Institute for Interdisciplinary Information Sciences (IIIS) is typically acquired through the web pages, which is limited in its static nature. In this course project, we develop a conversational system, named YaoTalk, to allow for voice-in/out information acquisition through natural dialogues. Unlike commercial form-based dialogue systems, we propose to exploit hierarchical knowledge structure by adopting a tag-based dialogue management module. This module is further coherently combined with an efficient database unit and a domain-specific ASR to meet our purpose. Through extensive testing, we evaluate the quality and efficiency of our developed system, and discuss its limitations. The results show our system is properly responsive, and leads to several potential directions of future work.

1. INTRODUCTION

The website of Institute for Interdisciplinary Information Sciences is an official platform for providing introduction and releasing information about Yao Class. Though enjoying great popularity among students in and out of the institute, its static nature limits the further possibility of flexible and user-friendly interaction.

In this course project, we have been developing a conversation based system, called YaoTalk in short, for this particular domain. The ultimate goal of this work is to develop a voice-in and voice-out dialogue system that can be deployed online along with the IIIS website. From a system development perspective, some of the challenges that we aim to resolve are (1) to extract information from the current IIIS webpages, (2) to exploit the knowledge structure in this domain so as to establish efficient representation and effective dialogue modeling, (3) to port existing speech recognition and synthesis frameworks by incorporating domain-specific information.

The following section describes various components of our system. Section 3 introduces in detail the data management deployed in our system. Section 4 discusses our experiment and some results. Section 5 concludes this work and discusses future work.

* A course project for the SCHM class taught by several dedicated professors from CUHK: Helen M. Meng, Lee Tan, Pak-Chung Ching.
† Website: www.tianlinshi.com

2. SYSTEM COMPONENTS

Figure 1 shows a block diagram of the YaoTalk system. The user communicates with the system through speech, which is recognized by an ASR adapted from CMU Sphinx
1
2
[2]. Keywords are extracted from the recognized text by tools including Jieba and Natural Language Toolkit (NLTK) [1]. The YaoTalk dialogue model is responsible for fusing information from the keywords as well as database and produce responses, while at the same time updating its information states. The responses are synthesized into natural language and converted into sound by Apple PlainTalk
2
2
. The basic components are mainly written in C++, except for the dialogue model, which is implemented using Python to allow for flexible modifications. The project utilizes Python Shelve for small data storages and MySQL for bigger, automatically generated data.

2.1 Dialogue Modeling

Through dialogue modeling, the YaoTalk system transforms information from the database into conversational interaction. Many commercial practices of dialogue systems, such as those in the Automatic Terminal Information Service (ATIS) domain [4] and in the automobile advertisement domain [3], typically adopt a form-based approach. In this...
M: Do you want to know something else, say the research fields?
M: Which specific aspect, research fields or courses?
M: How about a specific field? I would suggest quantum information.

Figure 3: Use tag structure to guide the dialogue.
(a). search for related topics. (b). narrow down the aspects. (c). specify the general topic.

Figure 2: Illustration of the tag-based dialogue modeling.

approach, the system aims to narrow down the search by filling out a form with a pre-determined structure.

Unlike these formed-based approaches, YaoTalk further exploits the hierarchical knowledge structure in the IIIS domain, and is capable of answering questions at varying levels of knowledge granularity. To be specific, our approach is tag-based and the dialogue state is modeled by a bag of tags. By extracting incoming keywords from user inquiries, the system discovers new tags that incrementally impose constraints so that the system could respond with more specific information. The basic idea is illustrated through Figure 2.

During conversation, the dialogue manager queries the database with the tags, so that it obtains a list of possible responses and potential follow-up questions. The responses and the questions can have the following type (1) a set of strings, which is passed to language synthesizer, (2) a general-purpose procedure, which is called with arguments being the tags, (3) NIL, which asks the model to make responses on its own.

Upon the third case, the dialogue system of YaoTalk is able to synthesize response based on the relationship of tags. In Figure 3, three relationships are shown. The YaoTalk dialogue system could base its follow-up questions on related topics, different aspects of some concept or specifications of the current topics.

Figure 4: Voice-In/Out components

2.2 Voice-In/Out

This essential Voice-I/O component for YaoTalk is implemented in C++ by combining the audio processing library PortAudio, the speech recognition toolkit CMU Sphinx and the speech synthesis module Apple PlainTalk. See Figure 4 for the component diagram.

For speech synthesis, the interface with PlainTalk is not hard as is given in OS X API; for speech recognition, we use PortAudio to capture 16 kHZ single-channel audio from the operating system, and activate a call-back function that processes 1024 frames at a time. The audio is then fed into the CMU Sphinx toolkit.

For a domain specific application, the challenge for speech recognition is to properly adapt the dictionary, the acoustic model and the language model by incorporating domain knowledge. Our dictionary is collected by crawling the IIIS website and is composed of words segmented from the crawled texts (Figure 5.a). Since acoustic data for IIIS domain is scarce and thus training a brand new model is impractical, we resort to an existing acoustic model trained from Chinese news broadcast programs. The phonemes in the original acoustic model is used to establish pronunciation
(a) Dictionary  (b) Acoustic model.

(c) Unigram in language model.

(d) Training bigram and trigram from sample queries.

Figure 5: How the YaoTalk system adapts domain knowledge to ASR.

Figure 6: The user interface to visualize the conversation.

Figure 7: Translate Table

We use MySQL (MariaDB) as the backup database, which stores the webpages crawled from IIIS website beforehand. Since the webpages are not well organized, they are not used directly by the system, but as a source of urls which can be provided when the user require further information. The pages are searched through link to this page (the text in `<a href=url>text</a>` html key), and the keyword in the webpage.

The main part of the database is in 'db', items of which share a common interface, as shown in Figure 8. The first three keys are for interface. The 'tag' key is the tag we used to search the database, all data item with any one of the queried keywords will be returned, with a similarity score calculated from 'tag' and the queried keywords.

The main database uses Python Shelve, which can store data in a python dictionary. The database contains two parts, 'translate' and 'db'. The former part is a translation table, which can map synonyms into one slot, as (Figure 7). Since the speech recognition cannot recognize English, we use this mapping to enable tagging conversion from Chinese to English.

The main part of the database is in 'db', items of which share a common interface, as shown in Figure 8. The first three keys are for interface. The 'tag' key is the tag we used to search the database, all data item with any one of the queried keywords will be returned, with a similarity score calculated from 'tag' and the queried keywords. The 'func' is a function, which takes the data item as a parameter, and returns a string for the front-end to speak. It is possible to introduce, for example, randomness in this function. The 'Q' is also a function taking the data item as

3An open Sphinx knowledge base tool for building language models.
parameter. It suggests a follow-up question that the front-end could ask after being given the queried information, and shows the front-end how to modify the next query (add or remove some keywords, or replace keywords) to go through the implicitly directed acyclic data graph (DAG). For example, as in Figure 9, if the next query is some kind of positive answer, then we remove all keywords, and replace them with the keyword in 'yes_keyword'. Note that if we only include

\[\text{Figure 8: Data Item}\]

\[
\text{def teacher_name_func(d):}
\text{\quad return u'(0)担任1的教授', format(d['\text{教师1}'], d['\text{课程1}'])}
\]

\[
\text{def teacher_q(d):}
\text{\quad return u'(0)你想进一步了解该课程吗?',}
\text{\quad u'yes_promp': u'd', u'yes_keyword': d['\text{课程1}'],}
\text{\quad u'no': None}}
\]

\[\text{Figure 9: Function Member}\]

'tag' and the data (key, value) pair, the data management would work just like a search engine. The functional design is more general in the sense that data could be processed and computed before being passed on to the dialogue management.

4. EXPERIMENT

In this section, we provide qualitative evaluation of the YaoTalk conversational system. A more quantitative evaluation is left as future work when the system is online, as it would require more test data from many users. Throughout the experiment, the conversation is visualized by a user interface shown in Figure 6.

Figure 10 and 11 shows the capabilities of our system. As we can see, when speaking with a cooperative speaker, the system performs very well by providing the key information needed by the user. This is because the tag-based representation restricts the search of knowledge and generates the right expectations for the dialogue.

However, when speaking with a non-cooperative speaker, the results in Figure 10 suggests the performance is not quite good. This is reasonable. Due to lack of knowledge outside the current domain, the system cannot make a plausible dialogue move, and generate user-friendly responses. This issue can probably be resolved as more data is added to our system, and if a self-learning mechanism could be developed (See section 5 for details).

5. CONCLUSION AND FUTURE WORK

In this work, we introduces the main idea that underlies YaoTalk, a conversational system for the IIS domain. By exploiting the tag-based representation, YaoTalk is able to answer questions and build dialogues with levels of knowledge granularity from coarse to fine, while the structure within these tags is further utilized to make dialogue moves. By fusing the domain specific knowledge with third-party...
tools like CMU Sphinx and PlainTalk, YaoTalk could deal with voice requests and produce voice responses, which as shown in the experiments is quite effective.

As future work, YaoTalk could be improved in the following ways. First, a quantitative evaluation method should be established, so that we could more objectively understand the performance of the system, in a way similar to the WHEELS system [3]. The performance evaluation might involve the accuracy of speech recognition, extent of user-friendliness, response time of the system, etc. Second, an automatic mechanism could be developed to extract information from online information in this domain, including but not limited to seminar notification and news report. Moreover, the dialogue system should have a learning mechanism by asking the expert user for issues not understood.

6. ACKNOWLEDGMENT

We would like to acknowledge the assistance and expertise of three dedicated professors from CUHK – Helen M. Meng, Lee Tan, Pak-Chung Ching – for teaching this course and for supervising the projects.

7. REFERENCES


