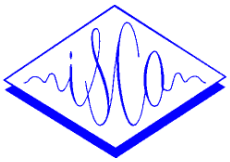


6th International Conference on Spoken Language Processing (ICSLP 2000)
Beijing, China
October 16-20, 2000

Natural Language Call Steering for Service Applications

Wu Chou[1], Qiru Zhou[1], Hong-Kwang Jeff Kuo[1], Antoine Saad[1],

David Attwater[2], Peter Durston[2], Mark Farrell[2], Frank Scahill[2]

1. Bell Labs. Lucent Technologies, 600 Mt. Ave., Murray Hill, NJ 07974, U.S.A.
2. BT Advanced Communications Technology Center, Adastral Park, Martlesham Heath, Ipswich, IP5 3RE, England, U.K.

{wuchou, qzhou, kuo, saad}@research.bell-labs.com,
{david.attwater, peter.durston, mark.farrell, frank.scahill}@bt.com

ABSTRACT

In this paper, a dialogue system for natural language based call steering is described and studied. The system is based on natural language speech recognition and understanding within a mixed initiative dialogue. The system is implemented on Bell Labs. Speech Technology Integration Platform (BLSTIP) using dialogue and natural language understanding components from BT laboratories. A prototype system in the operator service domain [2] is described. In order to improve the acoustic and language modeling for natural language based dialogue applications, various approaches are described and studied. The structure of the dialogue manager is also presented in which mixed-initiative dialogue can be supported with efficiency. Call classification and steering experiments were performed. The results confirm the efficacy of the proposed approach.

1. INTRODUCTION

Natural language dialogue between human and machine is a challenge. In order to make a natural language based dialogue system successful, various efforts are made to improve the accuracy, flexibility and robustness of the system component technologies, such as speech recognition, speech understanding, dialogue generation and dialogue manager, text-to-speech synthesis, etc. Such a complex dialogue application imposes stringent requirements on the flexibility of the system platform.

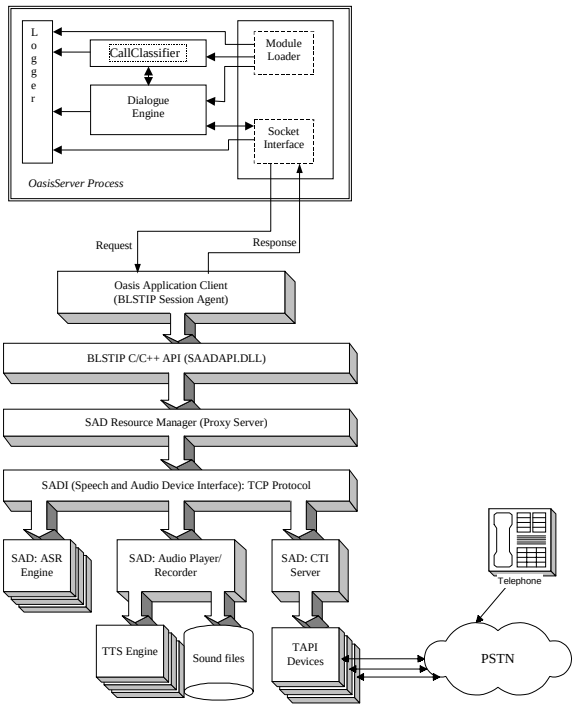
One of the drawbacks in systems deployed in the past is the limitation imposed by the finite state grammar on the language that a user can use to communicate with the machine. Although such constraint alleviates the complexity and problem in recognizing human speech, it becomes an obstacle to support more powerful, user friendly and flexible dialogue systems for mixed-initiative dialogues.

In this paper, we study issues encountered in designing and implementing a natural language based call steering application for telephone service calls. This is a complicated application, and it performs a detailed diagnostic dialogue to identify the service problem, such as a troubled telephone line and etc., that the user is experiencing. It provides the desired service after receiving user's consent and confirmation [2]. In the prototype system studied in this paper, the dialogue can go deep through many turns. The natural language based request and query from the user is recognized through natural language based automatic speech recognition. There is no constraint on the way that the user should communicate to the system. It allows the user to make direct requests as well as provide a description of the problem where the final action will be identified as the outcome of the dialogue. A call classifier provides natural language understanding based on the word string from the speech recognition output. The dialogue manager uses this understanding to determine the next appropriate system action.

The organization of this paper is as follows. In Section 2, the dialogue system architecture and design are presented which support natural language based mixed-initiative dialogue applications such as call steering, movie locator, etc. Section 3 is devoted to natural language based speech recognition and statistical language modeling for dialogue applications. Section 4 is concentrated on the dialogue manager design and automatic query generation. Call classification and steering are studied in Section 5 and results are given based on a case study in a telephone service application.

2. BLSTIP: ARCHITECTURE AND API

The natural language call steering (NLCS) system is built on top of Bell Labs Speech Technology Integration Platform (BLSTIP) [1], which provides a convenient mechanism for evaluating advanced natural language call steering applications. In order to support complicated dialogue application, BLSTIP provides a flexible API interface to core dialogue system platform functionalities such as telephony control, prompt play-out, speech recognition and text to speech synthesis (TTS).



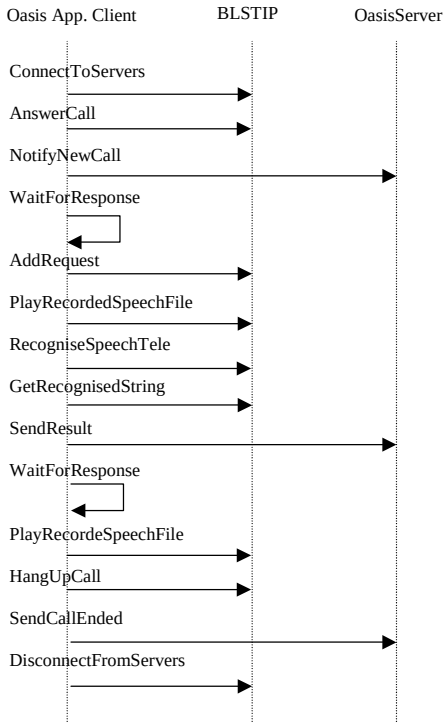


Figure 1. NLCS Trial System Architecture

In order to reduce application start up delay, we modified BLSTIP generic application architecture. Instead of requiring a user's call in to invoke the whole application dialogue, we run call classifier (CC), dialogue engine (DE) as active components under the NLCS dialogue manager (DM). A small, application session agent (client) is written using BLSTIP API to answer incoming calls from users, and initiate the application service logic by sending the initiation message to application dialogue manager (DM), then waiting for further instructions from the DM (forming an event loop). The DM takes control from this point on, until the session is finished, or the user hangs up. The DM maintains the service logic (the application session state machine) and provides speech/telephony interface service by sending service requests to its session agent. The agent calls BLSTIP API functions to execute the request and send received messages/events back to the DM. The NLCS application platform is shown in Figure 2.

3. ACOUSTIC MODELING AND LANGUAGE MODELING

Natural language based automatic speech recognition was applied in the dialogue system for service calls steering application. The reason to adopt a natural language based approach is that there is no control over the way that a user might use to describe the symptom of the problem and the system will be dynamically driven by both initiatives from user and machine as the dialogue proceeds.

The acoustic modeling for natural language based speech recognition was based on decision tree based state tying. A decision tree was built based on the data samples in the training data. It was obtained by a two-level robust clustering process, in which generalized context clusters were formed by relaxing the context constraints on those rarely seen contexts before performing the second level decision tree clustering [3]. The acoustic model is context dependent including within and crossword context dependent units. The crossword context dependent model units were obtained using a generalized tagging scheme to avoid the depletion of the available training data [3]. It is typical in new applications that there is only a very limited amount of in-domain training data and no available seed model to start with. In order to achieve fast deployment from one language to another, the UK English speech recognition system in the call steering task was developed from a North American English system without using any UK English specific seed model. This is made possible by an approach of pronunciation mapping and acoustic model adaptation. The system uses the same phone set as in American English, but is acoustically adapted to UK English.

Due to limited in-domain acoustic training data, data from other applications were used to increase triphone coverage. In our application, the portion of in-domain data in the model training set is less then 10% and more than 90% data are out-of-domain data. This disparity

introduces a new problem in acoustic modeling. Although out-of-domain data helps to improve the coverage of the acoustic model, it leads to an overgrown decision tree in which most tree nodes are based on out-of-domain data and cannot be reliably generalized to in-domain application. Therefore, control the structure and size of the acoustic model became a critical issue. The approach we adopted was based on the penalized Bayesian Information criterion [3]. It was applied in our application, and the model size was reduced by 50% without the loss of recognition accuracy. In order to adapt the model further into in-domain application, a new approach in model adaptation, extended maximum a posterior linear regression (EMAPLR) [6] was applied using the in-domain data. The limited in-domain data was applied twice in model building, one in decision tree state tying based model building and one in in-domain data adaptation. The performance improvements from the additional adaptation using the in-domain data ranges from 10-12% comparing to the approach of only using in-domain data for decision tree based model building in which in-domain data is mixed with out-of-domain data.

A trigram language model was trained for call steering application. Of the 9K transcriptions and recordings from the human-human dialogues, 8K was used for training and 1K used for language model testing. The training set contains about 180K words and the test set is about 25K words. The training corpus has a vocabulary size of about 4.5K words, but we restrict the language mode vocabulary to those words, which appear at least twice in the training corpus. This restriction brings the vocabulary size down to about 2.5K.

The language model was augmented by adding phrases to the original lexicon. Salient phrases, which are used by the classifier for routing the calls, are added to the lexicon for the language model building. A total of 38 phrases were added that include greeting words like “hello-there”, descriptions of the desired service like “wake-up-call,” times like “one-o’clock,” and special telephone numbers of specific services like “1-5-4”.

In addition to these phrases, which are manually selected, other phrases are automatically selected from the training data based on the maximum likelihood criterion [4]. In this study, phrases are added iteratively if they improve the unigram likelihood of the language model with respect to the training corpus. Adding phrases can improve recognition results by capturing a longer context length for the language model. Examples of phrases added by this algorithm include “could-you,” “I-need-to,” and “constantly-engaged”.

The perplexity results, as a function of the number of phrases added to the language model, are with respect to the held-out test set. The baseline trigram perplexity was 43.2 and bigram perplexity was 54.4. Adding about 200 phrases reduced the normalized trigram perplexity by 2% and reduced the bigram perplexity by 20%. A relative improvement of 3% in recognition accuracy was observed with phrase based language model. This language model has 2.4K unigram, 39K bigram and 92k trigram.

In addition to the language model that is used to handle the caller’s first utterance, a second language model is also used that handles confirmations like “yes” and “no, I don’t want a wake-up call, I want…,” However, in live trials, we found that the user behavior was different from what we expected in the way that how they made their service confirmation. More studies are required to determine what the user may say as follow up.

4. NLCS DIALOGUE MANAGER DESIGN

The prototype system uses a state-based dialogue engine to decide on the most appropriate response to the caller’s input. The dialogue engine contains a model for what the caller has asked for (or what the system believes they have asked for) on a blackboard. All inter-state transitions are conditional on the blackboard content, so as the call progresses a sequence of states is mapped out. The engine supports mixed-initiative dialogue-allowing the user efficiently change topic in case of an error. If the dialogue engine detects a problem in the interaction (normally signified by lack of progress through the states), the system exits to a human agent.

Although TTS is available on the platform, the prototype system uses high-quality recordings for audio output. The selection of the correct recording wording depends on the blackboard contents. In particular, exact wordings used by the caller are echoed in the system response, giving a more natural interaction.

5. CALL CLASSIFICATION AND STEERING

The Call Classifier (CC) component of the natural language dialogue system was designed to meet a set of criteria that would enable it to be portable and generic. These were:

It should employ a data-driven approach, as opposed to being handcrafted for a specific task.

It should be robust enough to cope with errors introduced by the NL speech recognition.

It should return a ranked list, by probability, of possible classification outcomes.

This section will discuss how these criteria influenced the design of the system. In addition, where appropriate, objective figures will be used to illustrate how well (or not) these were achieved.

During classifier training the same orthographic transcriptions used to train the n-gram language model, supplemented by data relating each transcription to one of 16 call-classes and one of 4 call-types, are used to identify salient phrases and keywords relating to the call-classes

and types. The result is a list of phrases and corresponding 'score' for each class.

When we classify a new example we generate a set of phrases which we lookup in our list of trained phrases. By summing the 'score' for each class we arrive at our raw classification result.

This approach entirely satisfies our first criteria, to employ a data-driven approach, and also significantly adds to the robustness of the system to recognition errors.

For example, given an example of an orthographic transcription and the error prone recognition output this method can still make a sensible classification decision – both cases being classified as an 'alarm call' request. The most salient phrases are indicated in **bold** type.

Transcription: *Hi I'd **like-to-book** an **alarm-call** please*

Recognition: *Hi I'd **like-to-book** a **call-please***

Objectively we can measure the effectiveness of our employed technique. Recall testing (training and testing on the same orthographic data) gives a classification accuracy of ~95%. Unseen testing (training and testing on exclusive sets of orthographic data) gives a classification accuracy of ~75%. When we test using the output of a real-time natural language speech recognition system the classification accuracy is ~55%.

Our final criteria are met by mapping the 'score' assigned to each class to a probability of that class being the correct class (confidence). We are helped in this task by a number of simple assumptions that hold true for our data:

Only one of the N classes can be correct.

The probability of the N-th class being correct if one of the previous N-1 results is correct is zero.

By analysis of our test data we can use these assumptions to build a model that accurately maps the 'score' assigned to a class to its confidence. Table 1 demonstrates that if we choose to operate within a confidence range the accuracy of the classifier is assured. In the design of natural dialogues, for example, this would allow a dialogue manager making decisions based on the confidence of a classification result.

Confidence Range (%)	50-59	60-69	70-79	80-89	90+
Proportion of Data (%)	8.8	14.6	4.8	10.4	42.9
Classifier Accuracy (%)	58.6	66.1	72.9	85.34	96.03

Table 1 : Classifier Accuracy for Confidence Ranges (orthographic transcriptions)

6. SUMMARY

In this paper, a dialogue system for natural language based call steering was described and studied. The system was implemented on Bell Labs Speech Technology Integration Platform (BLSTIP) that supports various dialogue applications. The natural language call steering system was designed to perform mixed-initiative dialogues, and it was based on natural language speech recognition and understanding. In order to improve acoustic and language modeling for natural language based dialogue applications, various approaches were described, including the approach of robust decision tree clustering, penalized Bayesian information criterion, EMAPLR adaptation, phrase based language modeling, and language model for confirmation, etc. The structure of the dialogue manager was also described in which mixed-initiative dialogue could be supported with efficiency. Call classification and steering experiments were performed, and results confirm the efficacy of the proposed approach.

Confidence Range (%)	0-9	10-19	20-29	30-39	40-49
Proportion of Data (%)	0.0	0.0	0.5	5.6	12.4
Classifier Accuracy (%)	-	-	26.83	34.89	44.51

7. REFERENCES

1. Q. Zhou, C.-H. Lee, W. Chou, A. Pargellis; "Speech Technology Integration and Research Platform: A System Study"; *5th European Conference on Speech Communication and Technology, TMC.2*, Rhodes, Greece; 22-25 Sept 1997.
2. M. Edgington, D. J. Attwater and P. J. Durston, "OASIS - a framework for Spoken Language Call Steering", Proc Europspeech'99
3. W. Chou, "Decision Tree Tying Based on Penalized Bayesian Information Criterion", Proc. ICASSP'99.
4. H.-K. J. Kuo and W. Reichl, "Phrase Based Language Models for Speech Recognition", Proc. EuroSpeech'99.
5. W. Reichl and W. Chou, "Robust Decision Tree State Tying for Continuous Speech Recognition", to appear in IEEE Trans. on Speech and Audio Processing.
6. W. Chou, O. Siohan, T. Andr'e Myrvoll and C.-H. Lee, " Extended Maximum A Posterior Linear Regression (EMAPLR) Model Adaptation for Speech Recognition", Proc. ICSLP'2000, Beijing.