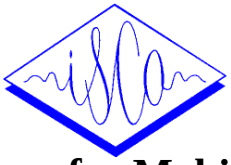

International Symposium on Chinese Spoken Language Processing (ISCSLP 2002)
Taipei, Taiwan
August 23-24, 2002

Multi-Speaker Dialogue for Mobile Information Retrieval

Hsien-Chang WANG

Department of Computer Science and Information Engineering
National Cheng-Kung University, Tainan
wangsj@csie.ncku.edu.tw

Chieh-Yi HUANG

Department of Computer Science and Information Engineering
National Cheng-Kung University, Tainan
chiehyi@ms12.hinet.net

Chung-Hsien YANG

Department of Electrical Engineering
National Cheng-Kung University, Tainan
n2888142@ccmail.ncku.edu.tw

Jhing-Fa WANG

Department of Electrical Engineering
National Cheng-Kung University, Tainan
wangjf@csie.ncku.edu.tw

ABSTRACT

Currently, most Spoken Dialogue Systems (SDS) only deal with the interaction between the system and one speaker. In some situations, interaction may occur between several speakers and the system. This paper proposes methods for the Multi-Speaker Dialogue System (MSDS) which allows user to retrieve useful information such as navigation guide, weather forecast, etc., in the car environment.

The differences between traditional SDS and MSDS are addressed first. The interaction between speakers and the MSDS are classified into three types i.e., independent, cooperative, and conflict. Then, we focus on two major research topics of the MSDS, i.e., speaker source identification which determines the active speaker and multi-speaker dialogue management which interpreters the speaker intention and maintains the dialogue histories to keep the interaction goes smoothly. .

Twenty-four testers attended the experiment for active speaker detection and multi-speaker dialogue system. The experiments showed an encouraged result that the proposed approach works properly, and it provides user-friendlier interface for multi-speaker interaction in the car environment.

1. INTRODUCTION

It has been several decades since the first development and release of ***Spoken Dialogue System*** (SDS). Currently, most SDSs only deal with the interaction between the system and one speaker. In some situations, interaction may occur between several speakers and the system. Speech signal processing and the dialogue management should be reconsidered to deal with multiple-speaker interaction. This motivates us to study the development of ***Multi-Speaker Dialogue System*** (MSDS).

In human-to-human multi-speaker dialogue, speakers may cooperate to accomplish a common goal, or negotiate to solve conflict opinions to achieve the same goal. The same cases may occur and should be considered in MSDS. We defined two types of goal in MSDS, i.e., the individual and global goal. The individual goal is that one speaker wants to achieve. Since individual goals may conflict with each other, system should maintain a global goal to integrate the individual goals. The following examples demonstrate different cases where individual goals do and do not conflict with each other. We classify the interaction between speakers and system into three types, i.e., ***independent, cooperative***, and ***conflict interaction***. The examples of multi-speaker dialogues are shown below:

(Independent, speakers have independent goals)
User1: What's the weather in Taipei?
User2: Where is the Tainan Station?

In the first example, the goal of two speakers are different and independent, system can handle this kind of queries by simply responding the result respectively.

(Cooperative, speakers have the same goal)
User1: Find a place for me to eat something.
User2: I want to have hamburger and coke.

In this example, the individual goal of User1 is to find a restaurant; and User2 is to eat hamburger. The dialogue manager should detect and integrate these individual goals to form the global goal, that is, a place where hamburger is supplied. Another example, in which the individual goals are conflicted, is given below:

(Conflict, speakers have conflict goals)
User1: Is there any Chinese-Petroleum gas station?
User2: I think the Formosa gas station is better.

In the last example, the global goal should be adjusted when speaker User2 has an individual goal differing from that of User1.

Previous study [1] of MSDS could deal with one goal only; speakers were not allowed to make cross-domain queries. To deal with multi-speaker dialogue interaction, techniques such as multiple microphone recording, source identification, noise canceling, multi-speaker voice activity detection and separation, and multi-speaker dialogue management should be studied first. Figure 1 is the flowchart of a multi-speaker dialogue system (four speakers in this figure). This paper focuses on the major component of the MSDS, i.e., speaker source identification and multi-speaker dialogue management.

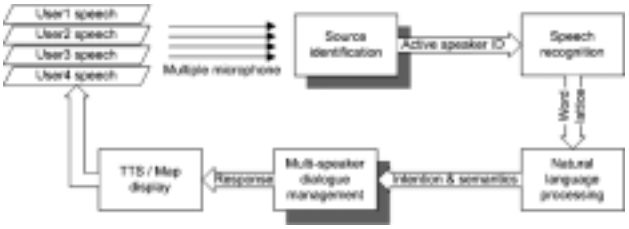

Figure 1. The flowchart of a multi-speaker dialogue system.

This paper is organized as follows. Section 2 describes how to identify the active speaker, which includes the issues about the choice and placement of multiple microphones in the car environment. Section 3 illustrates the algorithm of multi-speaker dialogue manager, together with several examples. Section 4 is the experimental results. The concluding remarks are given in final Section .

2. SPEECH SOURCE IDENTIFICATION

2.1. Multiple microphone placement

In a noisy environment, multi-microphone is usually used to enhance the input signal. We had studied many kinds of multi-microphone placement to determine the best one for multi-speaker purpose. Table 1 is the brief comparison of various kinds of microphone placement.

Microphone type	# of mic.	Advantage	Disadvantage	Suitable situation/ environment
Omni-directional microphone	Single	Low cost, easy to implement	Tend to contain noise	NULL
Unidirectional microphone	Single	Easy to implement, Low cost, Anti-noise	Inconvenient	Single user
Microphone array	Multiple	Source separation, Noise canceling	Complex to implement	Multi-speaker
Unidirectional microphones	Multiple	Easy to implement, Anti-noise	High cost	Multi-speaker in car

Table 1. Comparison of various types of microphone placement.

Concluded from our tests and surveys, currently, unidirectional microphones are the suitable for experiments in car environment. In our experiments, we place a unidirectional microphone in front of each speaker. Totally four microphones are used in our study.

2.2. Speech source identification

The block diagram of the multi-microphone processing is shown in Figure 2. We use a matched filter to identify the speech source (i.e., to identify the active speaker) in the multi-microphone placement; and an adaptive filter to estimate the enhanced (noise-free) signal.

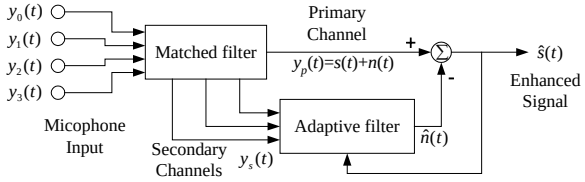

Figure 2. Block diagram of microphone processing

The matched filter is to detect speech in noisy environment, while user speaks to a microphone [2]. The output of matched filter from each microphone is compared with a threshold to decide from which microphone the speech comes, i.e., the primary channel. The impulse response $h_i(t)$ of the matched filter in the i-th microphone for signal $y_i(t)$ is given by

$$h_i(t)=y_i(N-1-t) \qquad 0\le t\le N-1 \tag{1}$$

where N is the length of the signal and i= 0, 1, 2, 3. The output of the matched filter is given by

$$z_i(t)=\sum_{t=0}^{N-1} h_i(t-k)y_i(t) \tag{2}$$

The decision of speech detection is made as

$$D_i(t)=\begin{cases}1 & if\ z_i(t)\ge Threshold\\ 0 & otherwise\end{cases} \tag{3}$$

We mark a microphone as primary channel if its $D_i(t)$ equals one, and the others are then classified as secondary channels. These secondary channels are used to estimate the noise $\hat{n}(t)$. The enhanced signal is obtained by subtracting $\hat{n}(t)$ from the primary channel $y_p(t)$. We make a common assumption that the signal and noise are uncorrelated. For the secondary channels $y_s(t)$, the minimum mean square error (MMSE) filter $h(n)$ is used to estimate the noise $\hat{n}(t)$. Applying the orthogonality principle to secondary channels [3], we have the following equation:

$$E\left\{\left(y_p(t)-\sum_{j=0}^{N-1}h(j)y_s(t-j)\right)y_s(t-k)\right\}=0 \tag{4}$$

Let $r_{y_s y_s}$ denotes the auto-correlation of y_s and $r_{y_s y_p}$ denote the cross-correlation between y_s and y_p. We can rewrite Eq. (4) into the following form

$$r_{y_p y_s}(k)-\sum_{j=0}^{N-1}h(j)r_{y_s y_s}(k-j)=0 \tag{5}$$

where k= 0,…,*N-1*. For simplicity, the matrix form of Eq. (5) is given by

$$\boldsymbol{r}_{y_p y_s} - \boldsymbol{R}_{y_s}\boldsymbol{h} = 0 \quad (6)$$

The gradient method is applied to solve Eq. (6), and we have:

$$\varepsilon(\boldsymbol{h}) = E\left\{\left(y_p(t) - \sum_{j=0}^{N-1} h(i)y_s(t-j)\right)^2\right\} \quad (7)$$

$$\frac{\partial\varepsilon(\boldsymbol{h})}{\partial\boldsymbol{h}} = 2(\boldsymbol{r}_{y_p y_s} - \boldsymbol{R}_{y_s}\boldsymbol{h}) \quad (8)$$

The Eq.(8) can be used to derive the coefficients of the filter $h(n)$, and thus we have the enhanced speech signal for the active speaker.

3. MULTI-SPEAKER DIALOGUE MANAGEMENT

Once the system is able to determine the active speaker, the nest important task is to maintain and keep smooth the interaction between system and multiple speakers. In an MSDS, each speaker may have his own "goal" for information retrieval, which is defined as individual goal in this paper. Contrasted to the individual goal, the global goal is the integration of each individual goal. The management of multi-speaker dialogue is to interpret intentions and semantics of individual speaker, to detect if there was conflict between speakers, to integrate individual goals into global goals, to determine whether a specific goal is completed, and to generate the response. In the next sections, we illustrate how the management of MSDS works by giving an algorithm and some examples.

3.1. Algorithm of multi-speaker dialogue management

The algorithm of multi-speaker dialogue management is shown in Algorithm 1. Each time the system receive the input of a speaker (the recognized result, word graph *WGi*), natural language processing technique [4] is applied to understand the intention and semantics of this speaker. We use a data structure, semantic vector, to record this information. Then, this semantic vector is combined with previous ones, to form the dialogue histories. Both individual and global histories are recorded in our method. Then, the integrated semantic vectors are used to determine if there was goal completed. The determination is based on whether essential information needed for a specific query is enough. For example, if the speaker is querying the weather condition, the essential information may be the location (ex. city name), weather type (ex. temperature), and date (ex. this afternoon). Once a goal is complete, system may perform database query and generate proper response to the speaker.

3.2. Examples of the interaction and management of an multi-speaker dialogue system

In the following examples, we illustrate the cases for 1) the speakers have independent individual goals, which can be solved easily; 2) the speakers having conflict individual goals, the system must resolve this problem before further information can be responded to the speakers; 3) the speakers have a common goal, which requires speakers cooperatively provide necessary information for the system.

Algorithm 1: Management of multi-speaker dialogue

Input: word graph of speech recognition for each speaker, $WG_1, WG_2, \ldots, WG_n$. where n is the number of total speakers.
Output: response to speakers.
Step 1: Initialization
Initialize the semantic vectors SV_i to be NULL.
$SV_i = (d^i_D, d^i_{PA}, d^i_{SA1}, d^i_{SA2}, \ldots)$, $i=1\sim n$
each element d^i_x is an integer. For speaker i,
d^i_D represents the domain that speaker mentioned;
d^i_{PA} is the primary attribute for this domain; and
d^i_{SAj} is the secondary attributes, where j varies with domain.
Initialize the dialogue history lists, H_i, for each speaker and H_S for system to be NULL.
Step 2: Determine semantics vector of each speaker
apply NLP techniques to WG_i to determine the corresponding semantic vector SV_i.
Step 3: Determine accomplished goal(s)
Semantic vector *SVi* for this turn is copied to the history *Hi*. *SVi* and *Hi* are integrated to check if a goal was completed.
Step 4: Decision.
If any goal is completed, go to Step 5; else go to Step 6.
Note that, under what condition a goal was complete is definable for each domain.
Step 5: Response.
Perform database query and generate response to the user according to the goal(s) found in Step 3. Go to Step 6.
Step 6: Iteration
Accept next input (WG_{i+1}) and go to Step 2.

Example 1. Speakers have different individual goals.

Time index	Action	Content
1	User1 input	"I want to go to the city hall".
2	User2 input	"Tell me the weather of Taipei"
3	Generate SVi	SV_1=("GUIDE", "DESTINATION", "city hall", Null…) SV_2=("WEATHER", "LOCATION", "Taipei", Null…)
4	Check goal completeness	User1=TRUE User2=TRUE
5	Check if conflict	NO
6	Generate response	"The weather in Taipei is rainy." "The city hall is about 450 meters away, please follow my instruction"

Example 2. Speakers have conflict individual goals.

Time index	Action	Content
1	User1 input	"Find me a Chinese food restaurant".
2	User2 input	"No, I want Italian specialties"
3	Generate SVi	SV_1=("GUIDE", "RESTAURANT", "Chinese food", Null…) SV_2=("GUIDE", "RESTAURANT", "Italian specialties", Null…)
4	Check goal completeness	User1=TRUE User2=TRUE
5	Check if conflict	YES
6	Generate response	"Please specify again, do you want Chinese food or Italian specialties"

Example 3. Speakers have common goal.

Time index	Action	Content
1	User1 input	"I want to know the route to …".
2	Generate Svi	SV_1=("GUIDE", "DESTINATION", Null…)
3	Check goal completeness	User1=FALSE, (no DESTINATION)
4	Generate response	"Please specify the destination."
5	User1 input User2 input	"To the nearest gas station" "And, how far is it"
6	Combine new SVi with old ones	SV_1=("GUIDE", "DESTINATION", "gas station", "nearest" Null…) SV_2=("DISTANCE", "DESTINATION", "gas station", "nearest", Null…)
7	Check goal completeness	User1=YES User2=YES
8	Check if conflict	NO
9	Generate response	"A gas station is 1.5 kilometers ahead, please go straight."

4. EXPERIMENTS

We set up an experimental automobile which contains 4 unidirectional microphones in front of each seat for the active speaker identification. The recording device is a notebook computer together with a PCMCIA multi-channel recording card. Totally 24 testers, grouped into eight groups, attended our experiments. Testers were informed with the system capability briefly. The domains, i.e., route guide, weather forecast and stock prices etc., which the system may provide information for, were also informed to the testers. Two types of experiments were carried out in our study, i.e., identifying the speech source and completing a dialogue.

4.1. Speech source identification

Several stable speeds of the vehicles are carefully maintained to control the decibel of the noise. Attendants of this experiment talked to the system, the recorded data are used for the speech source identification. The experimental result is shown in Table 2, which showed that the adopted approach achieves good performance for active speaker detection.

Correct rate / Situation	Speed (km/hr)		
	60	80	100
Silent	93 %	91 %	90 %
Radio on	91 %	90 %	87 %
Windows opened	91 %	89 %	88 %

Table 2. Experimental result of active speaker identification.

4.2. Dialogue completeness success rate

The evaluation of MSDS is done by checking the dialogue completeness rate for different situations. Experiments include single speaker, cooperative multi-speaker, and conflict multi-speaker. The results show only the part that speech recognition output the correct result in top_3 candidates. Table 3 shows the experimental results. Some notations used are described below.

S_C : number of correct turns for single speaker.

G_C : correct turns for cooperative speakers.

C_C : correct turns for conflict speakers

S_a : total number of turns for single speaker dialogues

G_a : total number of turns for cooperative dialogues

C_a : total number of turns for conflict dialogues

Type of multi-speaker dialogue	Correct rate
Single speaker (Sc/Sa)	91.2 %
Cooperative speakers (Gc/Ga)	89.5 %
Conflict speakers (Cc/Ca)	86.4 %

Table 3. Evaluation of the MSDS.

5. CONCLUSIONS

In this paper, we have addressed important issues for the development of multi-speaker dialogue system, especially in the car environment where every passenger may want to interact with the system. Deciding microphone placement, locating speaker position, and managing multi-speaker dialogue context are essential topic and should be further studied.

In fact, two kinds of interaction may occur in the car environment, the interaction between speaker and system (***interaction***), and between speaker and speaker (***intra-action***). This paper discussed the former only. How to model both the interaction and intra-action in an MSDS is a more difficult task and requires more studies. Research of multi-speaker spoken dialogue system (MSDS) is in the initial stage, we hope this paper can give rise to the research in the techniques of MSDS.

6. REFERENCES

[1] Young S.R., "Discourse Structure For Multi-Speaker Spontaneous Spoken Dialogs: Incorporating Heuristics Into Stochastic RTNS", *proceeding of ICASSP'95*, pp. 177~180, 1995.

[2] VASEGHI S.V., Advanced Digital Signal Processing and Noise Reduction. John Wiley & Sons, 2000.

[3] Haykin S., *Modern Filters*. Macmillan Publishing Company, 1989.

[4] Zue V, and Seneff S et al., "JUPITER: A Telephone-Based Conversation Interface for Weather Information", *IEEE Transaction on Speech and Audio Processing*, Vol.8, No. 1, pp.85~96, 2000.