ISCA Archive
http://www.isca-speech.org/archive

INTERSPEECH 2004 - ICSLP
8th International Conference on Spoken Language Processing
ICC Jeju, Jeju Island, Korea
October 4-8, 2004

Dynamic Time Windows for Multimodal Input Fusion

Anurag Gupta, Tasos Anastasakos

Motorola Labs
1301 E. Algonquin Road, Schaumburg, IL 61096, USA
{Anurag.Gupta, Tasos.Anastasakos}@motorola.com

Abstract

Natural interaction in multimodal dialogue systems demands quick system response after the end of a user turn. The prediction of the end of user input at each multimodal dialog turn is complicated as users can interact through modalities in any order, and convey a variety of different messages to the system within the turn. Several multimodal interaction frameworks have used fixed-duration time windows to address this problem. We conducted a user study to evaluate the use of fixed-duration time windows and motivate further improvements. This paper describes a probabilistic method for computing an adaptive time window for multimodal input fusion. The goal is to adjust the time window dynamically depending on the user, task, and the number of multimodal inputs for each turn. Experimental results show that the resulting system has superior performance when compared to a system with fixed-duration time windows.

1. Introduction

Multimodal systems collect, interpret, and combine multimodal inputs to understand the user's intention and then perform the requested task or generate a suitable system response. The interaction consists of alternating turns of user inputs and system responses. The Multimodal Input Fusion Manager (MMIFM) is a component of a complete multimodal system that collects and integrates multimodal inputs between successive system responses. After receiving an input, the MMIFM determines if the user has completed their turn before passing the combined multimodal inputs to a dialogue manager (DM) to generate a system response. Note that the MMIFM may not be implemented as a separate module in particular system architecture. The following requirements need to be considered for determining the end of a turn:

1) The system should provide unconstrained interaction capabilities (e.g., users must be able to speak anytime after making a gesture), and collect all their multimodal inputs for a turn before generating a response. The MMIFM must determine the end of turn without missing any input from the user in any modality and without imposing any interaction constraints on the user.

2) The system should respond promptly after the last user input for a turn. Thus the MMIFM should determine the end of turn as soon as a user finishes providing the last input.

Several multimodal interaction frameworks have proposed the use of fixed-duration time windows to determine the end of turn [5] [2]. A shortcoming of this approach is due to the delay between capturing the raw signal (such as voice input, gesture) and producing an interpretation (such as text transcription from voice or pen input) that varies considerably among the various modalities. In addition, user inputs may not follow strict timing patterns. Further exacerbating this problem, some of the modality interpreters may be distributed adding network latency to the delay. As a result, a fixed time window may expire before the MMIFM receives the interpretations from all inputs associated with a user turn.

This work proposes a probabilistic method for computing an adaptive time window for the MMIFM. The goal is to adjust the time window dynamically depending on the user, task, and the number of multimodal inputs for each turn. In the rest of the paper we describe the framework for MMIFM, the results of a user study that evaluates the use of fixed-duration time windows, and the proposed enhancements in the form of Dynamic Time Windows.

2. Multimodal Input Fusion Manager

The MMIFM is based on a 5-layer model [3] for interpreting multimodal inputs as shown in Figure 1. The model assumes turn-based interaction, i.e. a user provides an input and then the system responds (a user can interrupt a system's turn and start a new turn). The *Collection* layer collects raw signals from input modalities and passes them to the *Interpretation* layer which generates modality independent semantic interpretations of individual raw inputs in the form of typed feature structures (TFS) [1]. Modality recognizers and interpreters, e.g., a speech recognizer and natural language parser, perform the processes in these two layers. The MMIFM functionality is encapsulated in the *Classification* and *Integration* layers that receive TFSs from the *Interpretation* layer and determine the completion of a user turn. At the completion of a user turn, TFSs that are semantically compatible are combined into a single joint interpretation using an extended unification algorithm [4]. The joint interpretation is sent to a DM that manages the execution flow, can perform reasoning and provide with suitable system responses.

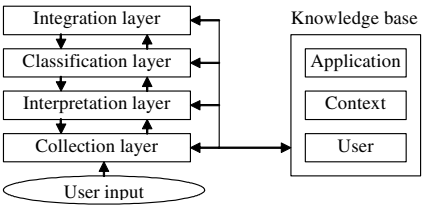

Figure 1: Model for multimodal input interpretation

The input collection strategy of the MMIFM using a fixed-duration time window is as follows: the MMIFM receives an input and if the combined interpretation (of the multimodal inputs collected so far in the turn) has missing feature values, it waits for a fixed duration to receive further inputs. The time window is restarted each time the MMIFM decides to wait for additional inputs. If no additional inputs

10.21437/Interspeech.2004-359

are received and the waiting period expires, the turn is considered complete. The MMIFM then combines the collected inputs to create a multimodal interpretation, transmits the multimodal interpretation to the DM, and starts a new cycle waiting for input from the user. We conducted an evaluation of the fixed-duration time window method and tested the effect of the duration variable on multimodal interaction. Based on previous empirical studies [8], we used two durations, 2 seconds and 4 seconds. The longer time duration was chosen to give users the perception that the system may not have collected their input, thus allowing observations about user behavior in such situations.

3. Use case study

The user study captures interactions with a multimodal in-car navigation system [4], developed using the MMIFM and a frame-based DM [10]. Users interact with a map-based graphical interface to get information on various locations and request driving directions. User input is entered through speech, handwriting, touch and gesture, either simultaneously or sequentially. In simultaneous multimodal interaction two or more modalities can be used to provide input at the same time. In sequential multimodal interaction although multiple modalities are available, they can only be used in succession, one modality at a time. Two versions of the system were running separately on two 650MHz computers, each with 256MB of RAM and a touch screen. One version of the navigation system used 2 seconds time window and the other version used 4 seconds time window.

An important statistic we calculated during this study was the time between the user providing the last input in a turn and the MMIFM determining the end of turn. We also calculated "under-collection" and "over-collection" errors generated by the MMIFM. An "under-collection" error occurs if a user turn, as collected by the MMIFM, does not have all the inputs provided by the user in that turn. For example, a user says, "I want to go there," and waits a few seconds before gesturing the location. If the time window expires before collecting the gesture input, the system makes an "under-collection" error. An "over-collection" error occurs when a user tries to re-enter their input within the same turn in cases that they feel the system is not responding, e.g., a user repeats, "I want to go there", after two seconds, when he or she feels the system did not capture his or her previous spoken input.

3.1. Subjects and Task

Sixteen subjects were enrolled for this study. The subjects, both male and female in the age group of 25-35 years, were working in technical fields and had daily interaction with computer-based systems. Before using the system, they were briefed about the tasks they needed to perform and given a demonstration of the system. Some representative tasks were:

- Have dialogue with the system to specify a few different destinations, e.g., a gas station, a hotel, an address, etc. This included specifying a destination going via a waypoint.
- Issue commands to control the map display, e.g., zoom to a certain area on the map.

These tasks could be completed either unimodally or multimodally. Some of them required multiple inputs from the same modality, e.g., providing a destination and a waypoint using touch. The users were free to choose their preferred mode of interaction for a particular task. The behavior of each user was observed during the interaction. Each of them was asked if they felt the system response time was acceptable and their response was recorded on a scale of 1 to 10.

3.2. Results

We observed 112 turns of interaction with the navigation system. 83% of those turns had two or more multimodal inputs. Considering the turns with multimodal inputs, 75% had only two inputs, and the rest had more than two inputs. In majority, or 95%, of the turns with multimodal inputs, users used only two modalities. These observations mean that users can use a modality more than once during a turn. Speech and touch/gesture were used 80%, handwriting and gesture were used 15%, and speech and handwriting were used 5% of the time. During multimodal interaction, 55% of the turns had simultaneous inputs, and the remaining 45% had sequential inputs. The average duration of a multimodal turn (not including the time to determine the end of turn) was 2.5 seconds. Unimodal commands required 30% longer time to issue than multimodal commands implying that multimodal interaction is faster. For the navigation system that used the 2-second time window, the over-collection error rate was 4% and the under-collection error rate was 6%. The average delay in determining the end of user turn after user provides their last input was 1.8 seconds. For the system with the 4-second time window, the over-collection error rate was 9% and the under-collection error rate was 2%. The average delay in determining the end of user turn was 2.9 seconds. The users gave an average rating of 6 out of 10 for the delay using the 2-second time window and 4 out of 10 for the delay using the 4-second time window.

3.3. Observations

We made the following observations during this study:

1. Sometimes it took a long time for some modalities to produce interpretations after capturing raw signals (e.g., when there was a long spoken input). The MMIFM received those interpretations a long time interval after previous inputs, and did not integrate them with previous inputs.
2. When users issued a multimodal command along with all the required parameters, they expected the system to respond without a delay. A system delay gave them the impression that the system did not capture their inputs. However, during multimodal dialogue, when users were unsure of all the information they needed to provide, it was acceptable to have a short delay.
3. Users tried to synchronize their multimodal inputs by closely following cross-modal references with the referred object. Each user preferred either to speak first and then touch or vice versa almost consistently, implying a preferred interaction style.
4. The task influenced the choice of modalities and the order in which they were used. The relationship between the timing of inputs in different modalities depended on the task being performed by the user, the duration of the current input, and the time that had elapsed since the

beginning of the user turn. The use of a particular modality influenced the input that can be provided in that modality. The modality used to provide the next input within a turn depended on the number of inputs already provided in the turn, the modality used to provide the current input, the duration of the current input compared to the average duration of an input in the current modality.

5. We observed that as the duration of time window increases, the under-collection error decreases but the over-collection error increases. The increase in the waiting time ensures all the inputs are collected. However, due to the increased waiting time, users assume that the system did not capture their input and retry. We also observed that the under-collection and over-collection errors are functions of the number of inputs within a multimodal turn, the duration of input in a given modality and the interaction style of each user. In general, as the numbers of inputs within a multimodal turn increased the over-collection error decreased and the under-collection error increased. Also, as the duration of a multimodal input increased, the under-collection error increased and the over-collection error decreased.

4. Dynamic Time Windows

Based on the observations from the user study, we implemented an adaptive wait period called *Dynamic Time Windows* (DTW) that would change depending on the user, the task, number of multimodal inputs received, etc. The DTW is computed using a Bayesian network (BN) [6].

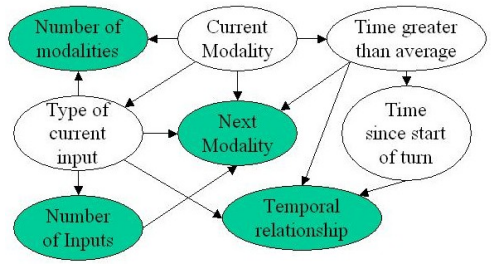

Figure 2: Bayesian network for predicting DTW

Figure 2 shows the BN that represents the joint probability distribution of eight variables (four inputs, and four outputs). These are:

1) *Type of current input* - an input variable that signifies the task or the object represented by the current input, e.g., to create a route.

2) *Current modality* – an input variable representing the name of the modality generating the current input.

3) *Time greater than average* – an input variable that is true if the duration of the current input is larger than the average duration of input in that modality.

4) *Time since start of turn* – an input variable representing the time elapsed since beginning of the turn. The duration is partitioned into three possible values - between 0 and 2 seconds, between 2 and 4 seconds, and 4 or more seconds.

5) *Number of modalities* – an output variable representing the number modalities that should be used within the turn.

6) *Number of inputs* – an output variable representing the number of interpretations that should be received in the turn.

7) *Next modality* – an output variable that represents the modality that should be used next. Its value can be either the name of a modality available in the system or a symbol "end of turn" predicting the end of turn.

8) *Temporal relationship* – an output variable representing the predicted temporal relationship between the current input and the next input based on the time ontology defined in OpenCyc [7]. Its value can be one out of the following five definitions – co-temporal, temporally subsumes, temporally intersects, temporally disjoint, and contiguous.

The causal and conditional dependencies represented within the BN were chosen based on the observations and analysis of multimodal data collected in the user study. For example, it can be stated that *Type of current input* is dependent on *Current modality*, and *Next modality* is dependent on *Number of inputs*, *Type of current input*, *Current modality*, and *Time greater than average* variables. The BN is trained using the Gradient Ascent training rule [9] that maximizes the conditional probability at each node by following the gradient of the natural log of the conditional probability with respect to the conditions (i.e., variables) it is dependent upon.

After receiving an input from a modality, the values for the following four variables are calculated: *Type of current input*, *Current modality*, *Time greater than average*, and *Time since start of turn*. The values for remaining variables are derived from the BN so that their conditional probability is maximized [9]. First, the value for *Number of inputs* variable is predicted and then values for *Next modality*, *Temporal relationship*, and *Number of modalities* variables are determined. The value with the maximum conditional probability is the predicted value for the respective variable.

4.1. Calculation of Dynamic Time Windows

The value of a DTW after receiving an input in a turn is calculated using Equation 1. *i* represents the modality that generated the last received input and *j* represents the modality that is predicted to be used next using the BN. $AvgDur_i$ represents the statistical average of the duration of inputs received from modality *i* during previous turns. $AvgTimeDiff_{ij}$ represents the statistical average of the time difference between the end times of successive inputs received from modalities *i* and *j,* respectively, in previous turns. The coefficients (c_i and c_{ij}) are based on the modalities *i* and *j*, the values of *Number of modalities* and *Temporal relationship* variables predicted using the BN, and the time that has been elapsed since the beginning of the turn.

$$DTW = c_i AvgDur_i + c_{ij} AvgTimeDiff_{ij} \quad (1)$$

5. Results

We collected additional multimodal data using the multimodal in-car navigation system to evaluate the efficiency of the adaptive time window and the performance of the BN. In total, 200 turns of interaction were recorded. From the collected multimodal data, 150 turns were used in the testing set, which provided 495 training instances for the network. The remaining 50 turns were used as the validation set to verify the training and measure the performance of the network. The 50 turns provided 116 testing instances in the validation set. The BN (Figure 2) is used to predict the value for four variables – *Next modality*, *Temporal relationship*, *Number of modalities*, and *Number of inputs*. We measured the accuracy of the predicted values against the observed values in the validation set. The value of the *Next modality* variable is 'end of turn' when the BN predicts that the user turn has ended. The accuracy in predicting the end of turn was also measured.

Figure 3 shows the measured accuracy for three variables and the end of the turn as a function of the number of epochs used to train the network. At low training periods, the accuracy of the predicted value for all the variables is very low (between 45 to 65%). At longer training periods, the accuracy improves to above 80% for the *Next modality* and *Number of modalities* variables and the *end of turn*. The prediction of the *Temporal relationship* variable had lower accuracy than other variables.

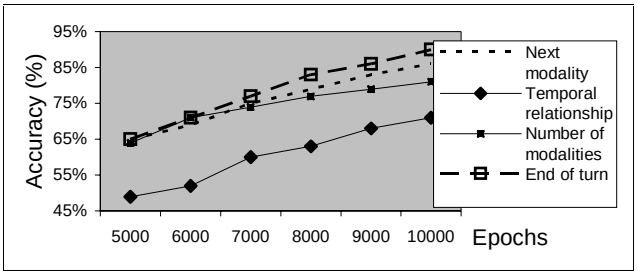

Figure 3: Accuracy of the predicted values of variables in the Bayesian network

The MMIFM using DTW was incorporated into the multimodal in-car navigation system. We simulated the provision of each input within the 112 turns of multimodal data, collected during the user study, with their exact timing to the MMIFM. Figure 4 shows the histogram of predicted time windows for those 112 turns. The average delay in determining the end of turn reduced to 1.3 seconds. This represents a 40% improvement on the user study results.

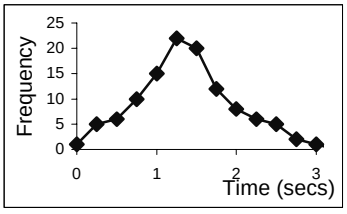

Figure 4: Histogram of dynamic time windows

We also conducted online experiments with the same users and tasks (as during the user study). The over-collection error rate reduced to 2% and under-collection errors reduced to 3% (compared to 4% and 6% respectively for 2 seconds time window). The improvement is partly due to the reduced delay in determining the end of user turn and also due to prediction of the preferred interaction style. DTW adjusts the time window to reduce the chances of each error. These results validate the effectiveness of the approach.

6. Conclusions and Future work

Future work should look into online learning to allow DTW to adapt to the current user, and using the beliefs of the DM regarding the user's intentions and the expected inputs to satisfy those intentions to determine the end of turn. We plan to enhance the interface between the MMIFM and input modalities so that the MMIFM can query the state of input modalities. It can then poll modalities that take a long time to process inputs to ensure that their input is not missed in a turn. This will help in reducing under-collection errors.

In conclusion, collecting and integrating multimodal input requires knowledge of the user's interaction style, the task at hand and a number of other factors. The DTW technique predicts the end of a user turn based on previous interaction history, current task being completed, number of inputs within the turn, etc. This technique significantly enhanced the usability of a multimodal system by reducing integration errors and improving response time.

7. References

[1] Carpenter, R. "*The logic of typed feature structures*", Cambridge University Press, Cambridge, 1992.

[2] Cohen, P. R., Johnston, M., McGee, D., Oviatt, S., Pittman, J., Smith, I., Chen, L., and Clow, J. "QuickSet: Multimodal interaction for distributed applications", *Proc. of the 5th International Multimedia Conference*, *ACM Press, New York, USA, 31-40, 1997.*

[3] Gupta, A. "A Reference Model for Multimodal Input Interpretation". *Extended Abstracts of CHI' 03, Ft. Lauderdale, FL, 2003.*

[4] Gupta, A., Lee, R., and Choi, E. "Multi-modal Dialogues As Natural User Interface For Automobile Environment" *Proc. of Australian Speech Science and Technology Conference, Melbourne, Australia*, *2002.*

[5] Johnston, M., Bangalore, S., Vasireddy, G., Stent, A., Ehlen, P., Walker, M., Whittaker, S., and Maloor, P. "MATCH: An Architecture for Multimodal Dialogue Systems". *Proc. of ACL'02, 376-383, 2002.*

[6] Mitchell, T. M. "*Machine Learning*", McGraw Hill, 1997.

[7] OpenCyc documentation at http://www.opencyc.org

[8] Oviatt, S.L., DeAngeli, A. and Kuhn, K. "Integration and synchronization of input modes during multimodal human-computer interaction". *Proc. of CHI '97, ACM Press, New York, USA, 415–422, 1997.*

[9] Russell, S., Binder, J., Koller, D., & Kanazawa, K. "Local learning in probabilistic networks with hidden variables", *Proc. of IJCAI'95, San Francisco, USA, 1995.*

[10] Thompson, W. and Bliss, H. "A Declarative Framework for building Compositional Dialog Modules". *Proc. of ICSLP'00, Beijing, China, 640 – 643, 2000.*