

# Anaphora Resolution using an Extended Centering Algorithm in a Multi-modal Dialogue System

Harksoo Kim, Jeong-Mi Cho, Jungyun Seo  
Department of Computer Science, Sogang University  
Seoul, 121-742, Korea

hskim@nlpzodiac.sogang.ac.kr, jmcho@nlprep.sogang.ac.kr, seojoy@ccs.sogang.ac.kr

## Abstract

Anaphora in multi-modal dialogues have different aspects compared to the anaphora in language-only dialogues. They often refer to the items signified by a gesture or by visual means. In this paper, we define two kinds of anaphora: screen anaphora and referring anaphora, and propose two general methods to resolve these anaphora. One is a simple mapping algorithm that can find items referred with/without pointing gestures on a screen. The other is the centering algorithm with a dual cache model, which Walker's centering algorithm is extended to for a multi-modal dialogue system. The extended algorithm is appropriate to resolve various anaphora in a multi-modal dialogue because it keeps utterances, visual information and screen switching-time. In the experiments, the system correctly resolved 384 anaphora out of 402 anaphora in 40 dialogues (0.54 anaphora per utterance) showing 95.5% correctness.

## Introduction

Human face-to-face communication is an ideal model for human-computer interface. One of the major features of face-to-face communication is its multiplicity of communication channels that acts on multiple modalities. By providing a number of channels through which information may pass between a user and a computer, a multi-modal dialogue system gives the user a more convenient and natural interface than a language-only dialogue system. In the system, a user often uses a variety of anaphoric expressions like *this*, *the red item*, *it*, etc. User's

intention is passed to the system through multiple channels, e.g., the auditory channel (carrying speech) and the visual channel (carrying gestures and/or facial expressions). For example, a user can say utterance (4) in Figure 1<sup>1</sup> while touching an item on the screen. The user may also say utterance (8) without touching the screen when there is only one red item displayed on the screen. Moreover, the user can use anaphoric expression to refer to an entity in previous utterances as in utterance (10).

- 
- (1) S: May I help you?
  - (2) U: I want to see some desks.
  - (3) S: (displaying *model 200* and *model 250*)  
We have *these models*.
  - (4) U: (pointing to the *model 200*)  
How much is *this*?
  - (5) S: *It* is 150,000 Won.
  - (6) U: I'd like to see some chairs, too.
  - (7) S: (displaying *model 100* and *model 150*)  
We have *these models*.
  - (8) U: How much is *the red item*?
  - (9) S: *It* is 80,000 Won.
  - (10) U: (pointing to the *model 100*)  
I'd like to buy *this* and *the previous selection*.
- 

Figure 1: Motivational example 1

Previous research on a multi-modal dialogue system was focused on finding the relationship between a pointing gesture and a deictic expression (Bolt (1980), Neal et al. (1988), Salisbury et al. (1990), Shimazu et al. (1994), Shimazu and Takashima (1996)) and on mapping a predefined symbol to a simple

---

<sup>1</sup> S means a multi-modal dialogue system and U means a user. Our goal is developing a multi-modal dialogue system (Kim and Seo (1997)) of which domain is home shopping and in which a user purchases furniture using Korean utterances with pointing gestures on a touch screen.

command (Johnston et al. (1997)). None of them, however, suggest methods of resolving deictic expressions with which pointing gestures are omitted; e.g., *the red item* in utterance (8). These approaches do not consider resolving an anaphoric expression that refers an object mentioned in previous utterances or displayed on previous screens. It, however, is important also for a multi-modal dialogue system to resolve all of these anaphora so that the system should correctly catch his/her intention. In this paper, we propose general methods to resolve a variety of anaphoric expressions that are found in a multi-modal dialogue. We classify anaphora into two types: deictic expression with/without a pointing gesture and referring expression, and propose methods to resolve them.

To resolve deictic expression like *this* in utterance (4) which co-occurs with a pointing gesture and *the red item* in utterance (8) which is uttered with no pointing gestures, the system counts the number of pointing gestures and the number of anaphoric noun phrases included in a user's utterance, and compares them. Then, the system maps the noun phrases to pointed items.

To resolve referring expression, one of the well known methods is *centering* theory developed by Grosz, Joshi, and Weinstein (Grosz et al. (1983)). The centering algorithm was further developed by Brennan, Friedman and Pollard for pronoun resolution (Brennan et al. (1987)) and was improved by Walker (Walker (1998)). However, those centering algorithms are not applicable to resolve anaphora in a multi-modal dialogue because the algorithm excludes the gestures and facial expression of a dialogue partner, which are important clues to understand his/her utterances. And, the algorithm cannot resolve complex anaphora like *the previous selection* in utterance (10) because it does not keep the time when the previous screen is switched to the current screen. To resolve such anaphora, we extend Walker's centering algorithm to the one with a dual cache model, which keeps the information displayed on a screen with screen switching-time.

The rest of this paper begins with describing our approach in section 1. After showing two methods to resolve anaphora in a multi-modal dialogue system in section 2, we report experimental results on these methods in section 3. Finally, we draw some conclusions.

## 1 Our approach

In this paper, we define two types of anaphora: screen anaphora and referring anaphora. Screen anaphora is an anaphoric noun phrase that refers to an entity on the present screen by a pointing gesture or through a visual channel. For example, *this* in utterance (4) in Figure 1 is the screen anaphora referred by a pointing gesture, and *the red item* in utterance (8) is the one referred through a visual channel. Referring anaphora is an anaphoric noun phrase that refers to an entity in previous utterances or on previous screens. For example, we call *It* in utterance (9) referring anaphora because the referred entity is *the red item* in the previous utterance (8). We also call *the previous selection* in utterance (10) referring anaphora because the referent is the *model 200* shown on the previous screen.

The screen anaphora resolution algorithm counts the number of pointing gestures and the number of anaphoric noun phrases included in the user's utterance and compares them. If the numbers are equal, the system maps the gestures to the phrases. Otherwise, the system uses some heuristics to map the gestures according to the priority of the phrases.

The referring anaphora resolution algorithm is based on the Walker's centering algorithm with a cache model. Centering is formulated as a theory that relates focus of attention, choice of referring expressions and perceived coherence of utterances, within a discourse segment. The centering algorithm (Grosz et al. (1983), Brennan et al. (1987), Walker et al. (1990)) consists of three main structures. *Forward-looking Centers* are entities which form a set of entities associated with each utterance. *Forward-looking Centers* are ranked according to their relative salience. The highest ranked entity is called the *Preferred Center*. *Backward-looking Center* is a special member of this set. It is the highest ranked member of *Forward-looking Centers* of the previous utterance, which is also realized in the current utterance. The algorithm defines a set of constraints, rules, and transition states between a pair of utterances by the use of these structures. It incorporates these rules and the other linguistic constraints to resolve anaphoric expressions. In the past, it was integrated with a stack model because most researchers believed that the centers should exist within a discourse segment (Grosz and Sidner

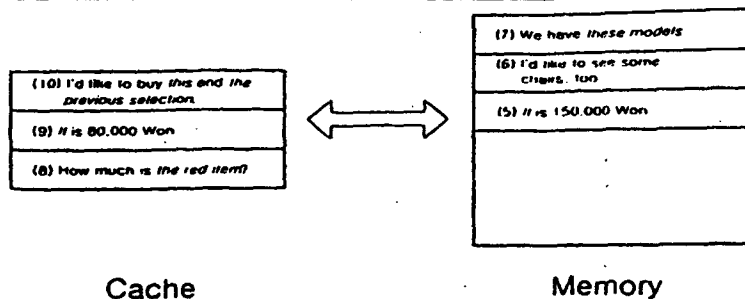


Figure 2: Walker's cache model

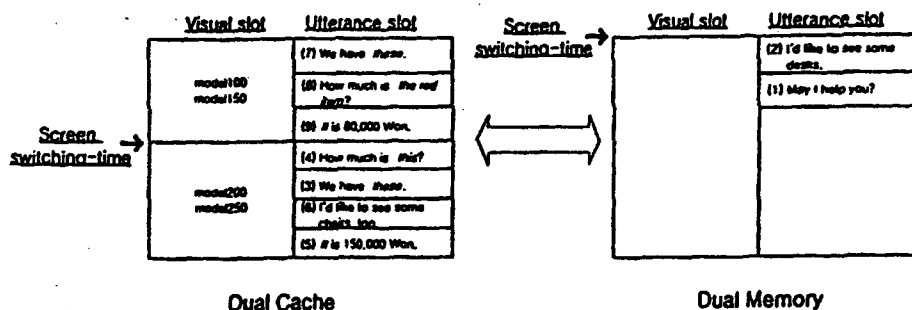


Figure 3: Dual cache model

(1986), Brennan et al. (1987), Walker (1989)). Walker replaced the stack model with a cache model because some objects were often referred in other discourse segments (Walker (1998)). The fundamental idea of the cache model is that the function of the cache when processing discourse is analogous to that of a cache when executing a program on a computer. In the cache model, the centers of an utterance are stored in the cache till the cache is full. When it is full, the least recently accessed centers in the cache are replaced to main memory. Figure 2 shows a state of the cache and memory when we apply the cache model to Figure 1. In Figure 2, we have replaced the centers by the user's utterances for simplifying the illustration. In Figure 2 and 3, the utterance with the highly ranked entities placed above the utterance with low ranked entities.

To resolve the anaphora like *the previous selection* shown in utterance (10) in Figure 1, the system should refer to previous utterances. If we employ Walker's cache model to find its referent, the system cannot get the correct result which is *model 200* in utterance (4), but *the red item* in utterance (8). The reason is that the model does not have the time base of the modifier, *previous*. In other words, it cannot decide the exact time of

*previous*. In this paper, we propose an extended centering algorithm with a dual cache model for a multi-modal dialogue system. To refer previous utterances and screens, the extended model keeps the centers of user's utterances, visual information, and screen switching-time. The visual information means the characteristics of the items on a screen, e.g., model number, color, size, shape and so on. The screen switching-time means the time when the system changes the previous screen to the current screen to show new items. Figure 3 shows a state of the dual cache and memory when we apply the new model to Figure 1. To decide the priority of the utterances, we present a priority rule; the more kinds of information an utterance carries, the higher priority the utterance has. One of the heuristic rules is that utterances occurring with pointing gestures have the higher possibility to be focused in a future dialogue than those without pointing gestures. If utterances carry the same kinds of information, the recent utterance have higher priority than the earlier utterance as in the stack model. For example, utterance (8) in Figure 1 were spoken earlier than utterance (9). It has, however, higher rank than utterance (9) as shown in Figure 3 because it occurred with visual information. Utterance (7) has higher rank

than utterance (8) because it occurred with gesture information such as the user's pointing gesture or the system's blinking gesture for the highlighted items as well as visual information. Now, if the dual cache model is applied to the system to find the referred entity of *the previous selection* shown in utterance (10) in Figure 1, the system can get the correct referent, *model 200* in utterance (4), because the model recognizes the time base of *previous* and searches centers in the utterance slot associated with the previous visual slot which includes *model 200* and *model 250* as shown in Figure 3.

## 2 Anaphora resolution algorithms

### 2.1 The screen anaphora resolution algorithm

The screen anaphora resolution algorithm replaces screen anaphora in an utterance with the items referred with/without pointing gestures. For example, if a user says "I'd like to buy *this* and *the red chair*." when pointing to an item on the screen, *this* in the utterance means the item that he/she points to and *the red chair* means the chair on the screen that is red. According to the number of anaphora and the number of gestures co-occurred with an utterance, we divide the algorithm into three cases.

**Case 1: The number of gestures and the number of anaphora are equal.**

Case 1 occurs most frequently in a multi-modal dialogue. In this case, the algorithm replaces the anaphora with the pointed items according to the order of occurrence. For example, *this* in the utterance (4) in Figure 1 can be resolved by this simple mapping.

**Case 2: The number of gestures is less than the number of anaphora.**

Case 2 occurs when a user omits a pointing gesture because he/she can uniquely select an item on the screen with the anaphoric expression or when there are referring anaphora as well as screen anaphora as in utterance (10) in Figure 1. In the former case, the algorithm can easily resolve the anaphora because it can uniquely decide the referred entity on the screen by visual information of the item. However, the algorithm cannot resolve the referring anaphora in this step because it needs to look at the previous utterances. The algorithm passes them to the referring anaphora resolution algorithm. We will

show the referring anaphora resolution algorithm in section 3.2.

**Case 3: The number of gestures is greater than the number of anaphora.**

Case 3 occurs when a user omits all or part of the utterance. The omission consists of two types: partial omission and total omission. To process the former, the algorithm first checks for missing essential cases in the case-frame. If the algorithm finds the omitted one, it fills the omitted one up and generates the supplemented result of the semantic analysis. For example, if a user says "How much?" when he/she points to *model 100* on the screen, the algorithm assumes that he/she omitted the theme *it* and generates the new semantic result like "How much is it?". Then, it processes the result according to the same method as Case 1. In the latter case, the algorithm assumes that the user uttered either "이것이요.(This, please.)" or "이것들이요.(These, please.)" because he/she just pointed to an item/items without an utterance. After restoring the omitted utterance, it can easily resolve the anaphora as Case 1.

### 2.2 The referring anaphora resolution algorithm

The referring anaphora resolution algorithm finds referents by using previous utterances and visual information. In utterance (10) in Figure 1, the user says, "I'd like to buy *this* and *the previous selection*." while he/she points to *model 100*. The system can resolve *this* using the screen anaphora resolution algorithm. However, the system cannot find the referred entity of *the previous selection*. The referring anaphora resolution algorithm resolves these kinds of anaphora. The algorithm is based on an extended centering algorithm with the dual cache model. The dual cache model consists of two slots and time points: visual slot, utterance slot, and screen switching-times as shown in Figure 4. The visual slot contains the visual information, *i.e.*, items displayed on a screen. The utterance slot contains the centers of the user's utterances. The screen switching-time, which is illustrated by an arrow, keeps the time when the previous screen is switched to the current screen. In Figure 4,  $V_k$  is the  $k$ th visual slot, which includes visual information of the  $k$ th screen.  $UP_{kj}$  means the utterance that has the  $j$ th priority at the  $k$ th visual slot.  $Cf$  is a list of

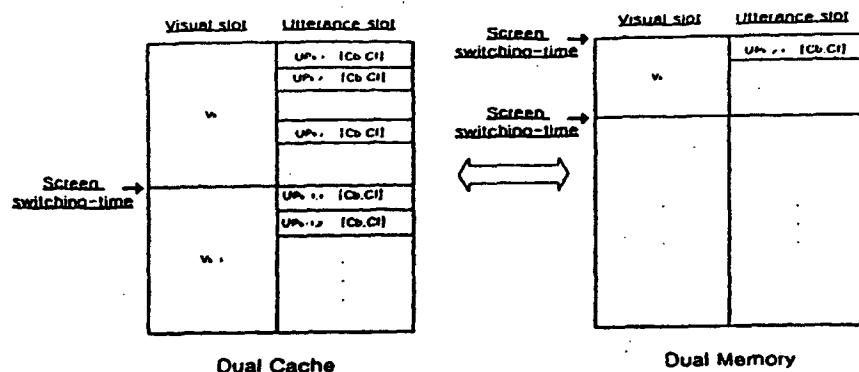


Figure 4: The structure of the dual cache model.

⊙ Priority Rule

The more kinds of information an utterance carries, the higher priority the utterance has. So, the priority between anchors is:

utterance + a pointing gesture > utterance + visual information > utterance with no other information

Figure 5: The priority of utterances

*Forward-looking Centers*, and *Cb* is the *Backward-looking Center*. A pair of *Cb* and *Cf* is called an anchor.

In the case of a language-only dialogue system, anchors stack up according to LIFO mechanism because the system processes only utterance without co-occurring information like a gesture. However, anchors in the dual cache model should not follow the mechanism because each utterance contains different kinds of information occurring with it. For example, utterance (4) in Figure 1 occurs with a gesture, and utterance (9) contains visual information. To decide the priorities among anchors, we propose a priority rule as shown in Figure 5. The rule means that utterances occurring with pointing gestures have the higher possibility to be focused in a future dialogue than those without pointing gestures. If anchors occur with the same kinds of information, they follow the LIFO mechanism. The reason why the algorithm should keep *Vk* in the dual cache model is that there are some anaphora referring to items which a user does not utter on the previous screen. As shown in utterance (5) in Figure 6, if the system does not keep the visual information, it cannot resolve *the previous red item* because the user saw the color of *model 200* through the visual channel but never uttered about the red item until utterance (5).

- (1) U: I'd like to see some desks.
- (2) S: (displaying *model 200* and *model 250*)  
We have these.
- (3) U: I'd like to see some chairs, too.
- (4) S: (displaying *model 100* and *model 150*)  
We have these.
- (5) U: (pointing to *model 100*)  
I'd like to buy *this* and *the previous red item*.

Figure 6: Motivational example 2

In this paper, the ranking of the items in *Cf* also follows Figure 5. If the items have the same priority, the algorithm ranks them by the obliqueness of grammatical relation of the subcategorized functions of the main verb: that is, first the subject, object, and objects<sub>2</sub>, followed by other subcategorized functions, and finally, adjuncts (Grosz and Sidner (1986), Brennan et al. (1987)).

The centering algorithm is based on constraints and rules as well as *Cbs* and *Cfs*. In this paper, we propose extended constraints and rules as shown in Figure 7 because the structure of the cache model and the priority of the utterances are changed according to Figure 4 and 5.

⊙ Constraints

1. There is exactly one  $Cb$  for each utterance  $U_i$ .
2. Every element of  $Cf(U_i)$  must be realized in  $U_i$ .
3.  $Cb(U_i)$  is highest-ranked element of  $Cf(UP_{k,j})$  that is realized in  $U_i$ .
4. The visual information and the gestures occurring with an utterance are not related to the previous utterances.

⊙ Rules

1. If some elements of  $Cf(UP_{k,j})$  are realized as a pronoun in  $U_i$ , then so is  $Cb(U_i)$ .
2. The priority of transitions from one utterance to the next is:

CONTINUE > RETAIN > SMOOTH SHIFT > ROUGH SHIFT

Figure 7: Extended constraints and rules

Figure 7 is similar to those of (Brennan et al. (1987)) except that  $Cf(U_{i-1})$  is replaced to  $Cf(UP_{k,j})$ , and the 4th constraint is added. The replacement means that  $Cb(U_i)$  may not be realized in the previous utterance,  $U_{i-1}$ . For example, the referent of *the previous selection* in utterance (10) in Figure 1 should be focused among the entities displayed in the previous screen. In such case,  $Cb(U_i)$  should be realized in the utterance  $UP_{k,j}$ , where  $U_k$  is the current visual slot, and  $j$  is decided by the priority rule in Figure 5. In other words, when the system detects a time base modifier such as *the previous*, first, it must decide the correct visual slot and then apply centering heuristics to decide  $Cb(U_i)$ . The 4th constraint shows that the user's current gesture must not be related to the previous utterances. We already adopted this constraint in the screen anaphora resolution algorithm in section 3.1. This constraint should also be applied to filter out unlike candidates. The transition types from one utterance to the next are extended as shown in Figure 8 by the same reason for the constraints and rules in Figure 7.

	$Cb(U_i) = Cb(UP_{k,j})$	$Cb(U_i) \neq Cb(UP_{k,j})$
$Cb(U_i) = Cp(U_i)$	CONTINUE	SMOOTH SHIFT
$Cb(U_i) \neq Cp(U_i)$	RETAIN	ROUGH SHIFT

Figure 8: Extended Transition states

The referring anaphora resolution algorithm that is based on these changes is the following: First, selects an anchor,  $Cb(UP_{k,j})$  and  $Cf(UP_{k,j})$ , and constructs all potential anchors in the present utterance  $U_i$ . In order to find the anchor in  $UP_{k,j}$ , we choose the  $k$ th visual information slot in the dual cache and memory and search anchors in utterance slots associated with the visual slot. That is, if the modifier expresses the time base like *previous* in Figure 1, the algorithm will search utterance slots associated with the previous visual information slot. During this process, it checks whether  $Cf(U_i)$  and  $Cf(UP_{k,j})$  are satisfied with the agreements, the grammatical functions and selectional restrictions (Brennan et al. (1987), Walker (1998)). In other words, the potential anchors are generated for each referring expression in an utterance and are specified for the agreements, the grammatical functions and the selectional restrictions. Then, it filters off unsuitable anchors using the extended filters in Figure 9, which are based on the constraints and rules in Figure 7. If an anchor remains, it is regarded as a pair of  $Cb$  and  $Cf$  in the present utterance.

For each anchor in the current list of anchors, apply the following filters derived from the centering constraints and rules. The first anchor that passes each filter is used to update the context. If more than one anchor at the same ranking passes all the filters, then the algorithm predicts that the utterance is ambiguous.

- ① FILTER 1: If the proposed  $Cb$  of the anchor does not equal the first element of this constructed list, then eliminate this anchor. This corresponds to constraint 3.
- ② FILTER 2: If none of the entities realized as anaphora in the proposed  $Cf$  equals the proposed  $Cb$ , then eliminate this anchor. If there are no anaphora in the proposed  $Cf$  then the anchor passes this filter. This corresponds to rule 1. However, anaphora that are resolved by a gesture and visual information must not be filtered. This corresponds to constraint 4.

Figure 9: Extended filters

Table 1: The component ratio of the experimental data and the experimental result

	Screen Anaphora	Referring Anaphora	Total
Frequency (the component ratio)	328 (81.6%)	74 (18.4%)	402 (100%)
# of detected anaphora (recall)	317 (96.6%)	71 (95.9%)	388 (96.5%)
# of resolved anaphora (precision)	316 (99.7%)	68 (94.4%)	384 (98.7%)
(# of resolved anaphora / Frequency)*100	96.3%	91.9%	95.5%

### 3 Evaluation and analysis of the experiments

#### 3.1 The experimental data

In order to experiment the proposed algorithms, we collected multi-modal dialogues which were simulated by 10 graduate students. They consist of 40 dialogues with 754 utterances (18.85 utterances per dialogue). The subject of the dialogue is furniture home shopping using a touch screen monitor. The data contains the user's utterances, pointing gestures and various visual information. In the data, we found 402 anaphoric noun phrases (10.5 anaphora per dialogue, and 0.54 anaphora per utterance). It means that anaphora resolution is very important for the multi-modal dialogue system. The screen anaphora appeared 4 times (81.6%) as much as the referring anaphora as shown in Table 1. It shows that a user usually points to the item when he/she wants to select an item on the screen.

#### 3.2 The analysis of experimental result

The two proposed algorithms detected 388 anaphora correctly from 402 anaphora and detected 1 anaphora incorrectly. The recall rate is 96.5% as shown in Table 1. The algorithms resolved 384 anaphora from the detected anaphora (389 anaphora), and the precision is 98.7% as shown in Table 1. Most of the failures were caused by preprocessing modules in our multi-modal dialogue system that generate an input for proposed anaphora resolution algorithms. The failure patterns are the following.

- The system failed to detect the anaphora modified by a subordinate clause like “*아까 선택했던 빨간 의자로 하죠.* (I'd like to buy the red chair that I selected before.)” because we restricted anaphora as noun phrases modified by noun or adjective.
- The screen anaphora resolution algorithm failed to know whether *두 모델(two models)* in “*두 모델의 가격 차이가 많이 있나요?* (What is the price difference of two models?)” was anaphoric noun phrase because a user did not explicitly use a definite article, *the*, in Korean. In English, we can easily know that *두 모델(two models)* is a anaphoric noun phrase because a user normally uses *the*. In Korean, however, it is difficult to find this kind of anaphora because the use of a definite article is a weak grammar rule.
- The referring anaphora resolution algorithm failed to resolve the anaphora like *아까 의자 다*(all of the previous chair) because the anaphora is expressed as a singular expression in Korean. Usually Koreans are not strict in number agreement. If the preprocessing modules can recognize the singular expression as the plural expression by looking at the meaning of *다(all)*, the algorithm can resolve these kinds of failure patterns.

As we can see in these failure cases, most failures are due to some special characteristics of Korean dialogues. We believe the proposed algorithms work much better in English multi-modal dialogues.

#### Conclusion

Unlike a language-only dialogue system, the multi-modal dialogue system has to resolve a

variety of anaphora because the system has various input channels. We proposed general algorithms to resolve such various anaphora in the multi-modal dialogue system. We defined two kinds of multi-modal anaphora, screen anaphora and referring anaphora. To resolve the screen anaphora, we proposed simple mapping algorithm. We proposed an extended centering algorithm integrated with the dual cache model to resolve referring anaphora. In the experiments, among 402 anaphora in 40 dialogues (0.54 anaphora per utterance), 384 anaphora were resolved. The result reflects the fact that the proposed algorithms work fairly well in resolving a variety of anaphora in multi-modal dialogues.

In a future work, we will test the system by using the Cf ranking method (Walker et al. (1990), Walker et al. (1994)) that Walker uses for Japanese. Since Korean is similar in structure to Japanese (i.e. it is a free word order, head-final language, with morphemic marking for grammatical function and topic), it would be interesting to see if the Cf ranking method can enhance our system's performance in multi-modal dialogue environment.

### Acknowledgements

Authors are grateful to the anonymous reviewers for their valuable comments on this paper. This work was supported in part by the Ministry of Information and Communication under the title of "A Research on Multimodal Dialogue Interface".

### References

- Bolt R. (1980) Put-That-There: Voice and gesture at the graphics interface. *Computer Graphics*, 14(3):262-270.
- Brennan S., Friedman M. and Pollard C. (1987) A Centering Approach to Pronoun. In *Proceedings of 25th ACL*, pp. 155-162.
- Grosz B. J., Joshi A. K., and Weinstein S. (1983) Providing a unified account of definite noun phrases in discourse. In *Proceedings of the 21st Annual Meeting of the ACL*, pp. 44-50.
- Grosz B. J. and Sidner C. L. (1986) Attentions, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175-204.
- Johnston M., Cohen P., McGee D., Oviatt S., Pittman J. and Smith I. (1997) Unification-based

Multimodal Integration. In *Proceedings of 8th EACL*, pp. 281-288.

- Kim H. and Seo J. (1997) A Multi-Modal User Interface System in Home Shopping Domain. (in Korean) In *'97 Spring Proceedings of Conference on Korea Cognitive Science Society*, pp. 74-80.
- Neal J., Dobes Z., Bettinger K. and Byoun J. (1988) Multi-Modal References in Human-Computer Dialogue. In *Proceedings of AAAI-88*, pp. 819-823.
- Salisbury M. W., Hendrickson J. H. and Lammers T. L., Fu C. and Moody S. A. (1990) Talk and draw: bundling speech and graphics. *IEEE Computer*, 23(8):59-65.
- Shimazu H., Arita S. and Takashima Y. (1994) Multi-Modal Definite Clause Grammar. In *Proceedings of COLING'94*, pp. 832-836.
- Shimazu H. and Takashima Y. (1996) Multi-Modal-Method: A Design Method for Building Multi-Modal Systems. In *Proceedings of COLING'96*, pp. 925-930.
- Walker M. (1989) Evaluating discourse processing algorithms. In *Proceedings of the 27th Annual Meeting of the ACL*, pp. 251-261.
- Walker M., Iida M. and Cote S. (1990) Centering in Japanese discourse. In *Proceedings of the 13th International Conference on Computational Linguistics*, pp. 1-7.
- Walker M., Iida M. and Cote S. (1994) Japanese discourse and the process of centering. *Computational Linguistics*, 20(2):193-233.
- Walker M. (1998) Centering Anaphora Resolution, and Discourse Structure. In: M. A. Walker, A. K. Joshi & E. F. Prince(Eds.), *Centering Theory in Discourse*. Oxford/UK: Oxford U.P., pp. 401-435.