Optimization in Multimodal Interpretation

Joyce Y. Chai[*] **Pengyu Hong**[+] **Michelle X. Zhou**[‡] **Zahar Prasov**[*]

[*]Computer Science and Engineering
Michigan State University
East Lansing, MI 48824
{jchai@cse.msu.edu,
prasovz@cse.msu.edu}

[+]Department of Statistics
Harvard University
Cambridge, MA 02138
hong@stat.harvard.edu

[‡]Intelligent Multimedia Interaction
IBM T. J. Watson Research Ctr.
Hawthorne, NY 10532
mzhou@us.ibm.com

Abstract

In a multimodal conversation, the way users communicate with a system depends on the available interaction channels and the situated context (e.g., conversation focus, visual feedback). These dependencies form a rich set of constraints from various perspectives such as temporal alignments between different modalities, coherence of conversation, and the domain semantics. There is strong evidence that competition and ranking of these constraints is important to achieve an optimal interpretation. Thus, we have developed an optimization approach for multimodal interpretation, particularly for interpreting multimodal references. A preliminary evaluation indicates the effectiveness of this approach, especially for complex user inputs that involve multiple referring expressions in a speech utterance and multiple gestures.

1 Introduction

Multimodal systems provide a natural and effective way for users to interact with computers through multiple modalities such as speech, gesture, and gaze (Oviatt 1996). Since the first appearance of "Put-That-There" system (Bolt 1980), a variety of multimodal systems have emerged, from early systems that combine speech, pointing (Neal et al., 1991), and gaze (Koons et al, 1993), to systems that integrate speech with pen inputs (e.g., drawn graphics) (Cohen et al., 1996; Wahlster 1998; Wu et al., 1999), and systems that engage users in intelligent conversation (Cassell et al., 1999; Stent et al., 1999; Gustafson et al., 2000; Chai et al., 2002; Johnston et al., 2002).

One important aspect of building multimodal systems is multimodal interpretation, which is a process that identifies the meanings of user inputs. In a multimodal conversation, the way users communicate with a system depends on the available interaction channels and the situated context (e.g., conversation focus, visual feedback). These dependencies form a rich set of constraints from various aspects (e.g., semantic, temporal, and contextual). A correct interpretation can only be attained by simultaneously considering these constraints. In this process, two issues are important: first, a mechanism to combine information from various sources to form an overall interpretation given a set of constraints; and second, a mechanism that achieves the best interpretation among all the possible alternatives given a set of constraints. The first issue focuses on the fusion aspect, which has been well studied in earlier work, for example, through unification-based approaches (Johnston 1998) or finite state approaches (Johnston and Bangalore, 2000). This paper focuses on the second issue of optimization.

As in natural language interpretation, there is strong evidence that competition and ranking of constraints is important to achieve an optimal interpretation for multimodal language processing. We have developed a graph-based optimization approach for interpreting multimodal references. This approach achieves an optimal interpretation by simultaneously applying semantic, temporal, and contextual constraints. A preliminary evaluation indicates the effectiveness of this approach, particularly for complex user inputs that involve multiple referring expressions in a speech utterance and multiple gestures. In this paper, we first describe the necessities for optimization in multimodal interpretation, then present our graph-based optimization approach and discuss how our approach addresses key principles in Optimality Theory used for natural language interpretation (Prince and Smolensky 1993).

2 Necessities for Optimization in Multimodal Interpretation

In a multimodal conversation, the way a user interacts with a system is dependent not only on the available input channels (e.g., speech and gesture), but also upon his/her conversation goals, the state of the conversation, and the multimedia feedback from the system. In other words, there is a rich context that involves dependencies from many different aspects established during the interaction. Interpreting user inputs can only be situated in this rich context. For example, the temporal relations between speech and gesture are important criteria that determine how the information from these two modalities can be combined. The focus of attention from the prior conversation shapes how users refer to those objects, and thus, influences the interpretation of referring expressions. Therefore, we need to simultaneously consider the temporal relations between the referring expressions and the gestures, the semantic constraints specified by the referring expressions, and the contextual constraints from the prior conversation. It is important to have a mechanism that supports competition and ranking among these constraints to achieve an optimal interpretation, in particular, a mechanism to allow constraint violation and support soft constraints.

We use temporal constraints as an example to illustrate this viewpoint[1]. The temporal constraints specify whether multiple modalities can be combined based on their temporal alignment. In earlier work, the temporal constraints are empirically determined based on user studies (Oviatt 1996). For example, in the unification-based approach (Johnston 1998), one temporal constraint indicates that speech and gesture can be combined only when the speech either overlaps with gesture or follows the gesture within a certain time frame. This is a hard constraint that has to be satisfied in order for the unification to take place. If a given input does not satisfy these hard constraints, the unification fails.

In our user studies, we found that, although the majority of user temporal alignment behavior may satisfy pre-defined temporal constraints, there are some exceptions. Table 1 shows the percentage of different temporal relations collected from our user studies. The rows indicate whether there is an overlap between speech referring expressions and their accompanied gestures. The columns indicate whether the speech (more precisely, the referring expressions) or the gesture occurred first. Consistent with the previous findings (Oviatt et al, 1997), in most cases (85% of time), gestures occurred before the referring expressions were uttered. However, in 15% of the cases the speech referring expressions were uttered before the gesture occurred. Among those cases, 8% had an overlap between the referring expressions and the gesture and 7% had no overlap.

	Speech First	Gesture First	Total
Non-overlap	7%	45%	52%
Overlap	8%	40%	48%
Total	15%	85%	100%

Table 1: Overall temporal relations between speech and gesture

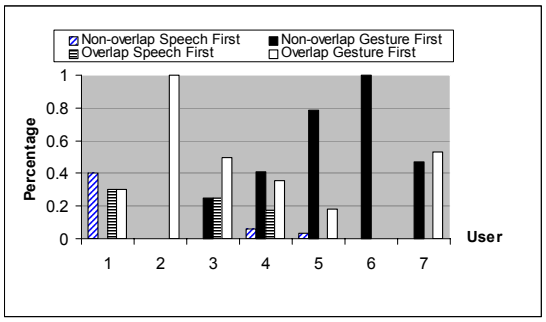

Figure 1: Temporal relations between speech and gesture for individual users

Furthermore, as shown in (Oviatt et al., 2003), although multimodal behaviors such as sequential (i.e., non-overlap) or simultaneous (e.g., overlap) integration are quite consistent during the course of interaction, there are still some exceptions. Figure 1 shows the temporal alignments from seven individual users in our study. User 2 and User 6 maintained a consistent behavior in that User 2's speech referring expressions always overlapped with gestures and User 6's gesture always occurred ahead of the speech expressions. The other five users exhibited varied temporal alignment between speech and gesture during the interaction. It will be difficult for a system using pre-defined temporal constraints to anticipate and accommodate all these different behaviors. Therefore, it is desirable to have a mechanism that

[1] We implemented a system using real estate as an application domain. The user can interact with a map using both speech and gestures to retrieve information. All the user studies mentioned in this paper were conducted using this system.

allows violation of these constraints and support soft or graded constraints.

3 A Graph-based Optimization Approach

To address the necessities described above, we developed an optimization approach for interpreting multimodal references using graph matching. The graph representation captures both salient entities and their inter-relations. The graph matching is an optimization process that finds the best matching between two graphs based on constraints modeled as links or nodes in these graphs. This type of structure and process is especially useful for interpreting multimodal references. One graph can represent all the referring expressions and their inter-relations, and the other graph can represent all the potential referents. The question is how to match them together to achieve a maximum compatibility given a particular context.

3.1 Overview

Graph-based Representation

Attribute Relation Graph (ARG) (Tsai and Fu, 1979) is used to represent information in our approach. An ARG consists of a set of nodes that are connected by a set of edges. Each node represents an entity, which in our case is either a referring expression to be resolved or a potential referent.

Each node encodes the properties of the corresponding entity including:

- *Semantic information* that indicates the semantic type, the number of potential referents, and the specific attributes related to the corresponding entity (e.g., extracted from the referring expressions).
- *Temporal information* that indicates the time when the corresponding entity is introduced into the discourse (e.g., uttered or gestured).

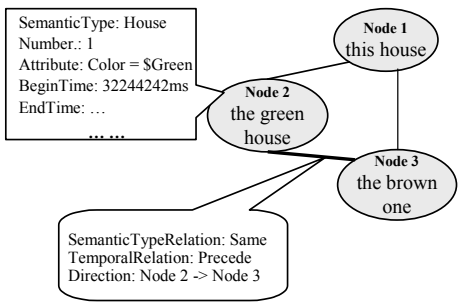

Figure 2: An example of a referring graph

Each edge represents a set of relations between two entities. Currently we capture *temporal relations* and *semantic type relations*. A temporal relation indicates the temporal order between two related entities during an interaction, which may have one of the following values:

- *Precede*: Node A precedes Node B if the entity represented by Node A is introduced into the discourse before the entity represented by Node B.
- *Concurrent*: Node A is concurrent with Node B if the entities represented by them are referred to or mentioned simultaneously.
- *Non-concurrent*: Node A is non-concurrent with Node B if their corresponding objects/references cannot be referred/mentioned simultaneously.
- *Unknown*: The temporal order between two entities is unknown. It may take the value of any of the above.

A semantic type relation indicates whether two related entities share the same semantic type. It currently takes the following discrete values: *Same*, *Different*, and *Unknown*. It could be beneficial in the future to consider a continuous function measuring the rate of compatibility instead.

Specially, two graphs are generated. One graph, called the *referring graph,* captures referring expressions from speech utterances. For example, suppose a user says Compare this house, the green house, and the brown one. Figure 2 show a referring graph that represents three referring expressions from this speech input. Each node captures the semantic information such as the semantic type (i.e., Semantic Type), the attribute (Color), the number (Number) of the potential referents, as well as the temporal information about when this referring expression is uttered (BeginTime and EndTime). Each edge captures the semantic (e.g., SemanticTypeRelation) and temporal relations (e.g., TemporalRelation) between the referring expressions. In this case, since the green house is uttered before the brown one, there is a temporal Precede relationship between these two expressions. Furthermore, according to our heuristic that objects-to-be-compared should share the same semantic type, therefore, the SemanticTypeRelation between two nodes is set to Same.

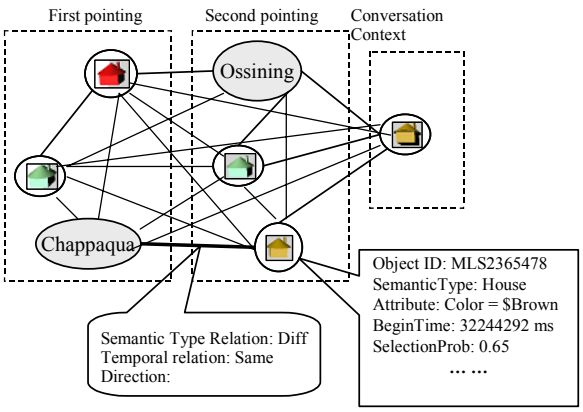

Figure 3: An example of referent graph

Similarly, the second graph, called the *referent graph*, represents all potential referents from multiple sources (e.g., from the last conversation, gestured by the user, etc). Each node captures the semantic and temporal information about a potential referent (e.g., the time when the potential referent is selected by a gesture). Each edge captures the semantic and temporal relations between two potential referents. For instance, suppose the user points to one position and then points to another position. The corresponding referent graph is shown in Figure 3. The objects inside the first dashed rectangle correspond to the potential referents selected by the first pointing gesture and those inside the second dashed rectangle correspond to the second pointing gesture. Each node also contains a probability that indicates the likelihood of its corresponding object being selected by the gesture. Furthermore, the salient objects from the prior conversation are also included in the referent graph since they could also be the potential referents (e.g., the rightmost dashed rectangle in Figure 3[2]).

To create these graphs, we apply a grammar-based natural language parser to process speech inputs and a gesture recognition component to process gestures. The details are described in (Chai et al. 2004a).

[2] Each node from the conversation context is linked to every node corresponding to the first pointing and the second pointing.

Graph-matching Process

Given these graph representations, interpreting multimodal references becomes a graph-matching problem. The goal is to find the best match between a referring graph (G_s) and a referent graph (G_r). Suppose

- A referring graph $G_s = \langle \{\alpha_m\}, \{\gamma_{mn}\} \rangle$, where $\{\alpha_m\}$ are nodes and $\{\gamma_{mn}\}$ are edges connecting nodes α_m and α_n. Nodes in G_s are named *referring nodes*.
- A referent graph $G_r = \langle \{a_x\}, \{r_{xy}\} \rangle$, where $\{a_x\}$ are nodes and $\{r_{xy}\}$ are edges connecting nodes a_x and a_y. Nodes in G_r are named *referent nodes*.

The following equation finds a match that achieves the maximum compatibility between G_r and G_s:

$$Q(G_r, G_s) = \sum_x \sum_m P(a_x, \alpha_m) NodeSim(a_x, \alpha_m) + \sum_x \sum_y \sum_m \sum_n P(a_x, \alpha_m) P(a_y, \alpha_n) EdgeSim(r_{xy}, \gamma_{mn}) \quad (1)$$

In Equation (1), $Q(G_r, G_s)$ measures the degree of the overall match between the referent graph and the referring graph. $P(a_x, \alpha_m)$ is the matching probability between a node a_x in the referent graph and a node α_m in the referring graph. The overall compatibility depends on the similarities between nodes (*NodeSim*) and the similarities between edges (*EdgeSim*). The function $NodeSim(a_x, \alpha_m)$ measures the similarity between a referent node a_x and a referring node α_m by combining semantic constraints and temporal constraints. The function $EdgeSim(r_{xy}, \gamma_{mn})$ measures the similarity between r_{xy} and γ_{mn}, which depends on the semantic and temporal constraints of the corresponding edges. These functions are described in detail in the next section.

We use the graduated assignment algorithm (Gold and Rangarajan, 1996) to maximize $Q(G_r, G_s)$ in Equation (1). The algorithm first initializes $P(a_x, \alpha_m)$ and then iteratively updates the values of $P(a_x, \alpha_m)$ until it converges. When the algorithm converges, $P(a_x, \alpha_m)$ gives the matching probabilities between the referent node a_x and the referring node α_m that maximizes the overall compatibility function. Given this probability matrix, the system is able to assign the most probable referent(s) to each referring expression.

3.2 Similarity Functions

As shown in Equation (1), the overall compatibility between a referring graph and a referent graph depends on the node similarity

function and the edge similarity function. Next we give a detailed account of how we defined these functions. Our focus here is not on the actual definitions of those functions (since they may vary for different applications), but rather a mechanism that leads to competition and ranking of constraints.

Node Similarity Function

Given a referring expression (represented as α_m in the referring graph) and a potential referent (represented as a_x in the referent graph), the node similarity function is defined based on the semantic and temporal information captured in a_x and α_m through a set of individual compatibility functions:

$$NodeSim(a_x,\alpha_m) = Id(a_x,\alpha_m)\ SemType(a_x,\alpha_m)$$
$$\Pi_k Attr_k(a_x,\alpha_m)\ Temp(a_x,\alpha_m)$$

Currently, in our system, the specific return values for these functions are empirically determined through iterative regression tests.

***Id*(a_x,α_m)** captures the constraint of the compatibilities between identifiers specified in a_x and α_m. It indicates that the identifier of the potential referent, as expressed in a referring expression, should match the identifier of the true referent. This is particularly useful for resolving proper nouns. For example, if the referring expression is *house number eight*, then the correct referent should have the identifier *number eight*. We currently define this constraint as follows: $Id(a_x,\alpha_m) = 0$ if the object identities of a_x and α_m are different. $Id(a_x,\alpha_m) = 100$ if they are the same. $Id(a_x,\alpha_m) = 1$ if at least one of the identities of a_x and α_m is unknown. The different return values enforce that a large reward is given to the case where the identifiers from the referring expressions match the identifiers from the potential referents.

***SemType*(a_x,α_m)** captures the constraint of semantic type compatibility between a_x and α_m. It indicates that the semantic type of a potential referent as expressed in the referring expression should match the semantic type of the correct referent. We define the following: $SemType(a_x,\alpha_m) = 0$ if the semantic types of a_x and α_m are different. $SemType(a_x,\alpha_m) = 1$ if they are the same. $SemType(a_x,\alpha_m) = 0.5$ if at least one of the semantic types of a_x and α_m is unknown. Note that the return value given to the case where semantic types are the same (i.e., "1") is much lower than that given to the case where identifiers are the same (i.e., "100"). This was designed to support constraint ranking. Our assumption is that the constraint on identifiers is more important than the constraint on semantic types. Because identifiers are usually unique, the corresponding constraint is a greater indicator of node matching if the identifier expressed from a referring expression matches the identifier of a potential referent.

***Attr*$_k$(a_x,α_m)** captures the domain specific constraint concerning a particular semantic feature (indicated by the subscription k). This constraint indicates that the expected features of a potential referent as expressed in a referring expression should be compatible with features associated with the true referent. For example, in the referring expression *the Victorian house*, the style feature is *Victorian*. Therefore, an object can only be a possible referent if the style of that object is *Victorian*. Thus, we define the following: $A_k(a_x,\alpha_m) = 1$ if both a_x and α_m share the kth feature with the same value. $A_k(a_x,\alpha_m) = 0$ if both a_x and α_m have the feature k and the values of the feature k are not equal. Otherwise, when the kth feature is not present in either a_x or α_m, then $A_k\ (a_x,\alpha_m) = 0.1$. Note that these feature constraints are dependent on the specific domain model for a particular application.

***Temp*(a_x,α_m)** captures the temporal constraint between a referring expression α_m and a potential referent a_x. As discussed in Section 2, a hard constraint concerning temporal relations between referring expressions and gestures will be incapable of handling the flexibility of user temporal alignment behavior. Thus the temporal constraint in our approach is a graded constraint, which is defined as follows:

$$Temp(a_x,\alpha_m) = \exp(-\frac{|\ BeginTime(a_x) - BeginTime(\alpha_m)\ |}{2000})$$

This constraint indicates that the closer a referring expression and a potential referent in terms of their temporal alignment (regardless of the absolute precedence relationship), the more compatible they are.

Edge Similarity Function

The edge similarity function measures the compatibility of relations held between referring expressions (i.e., an edge γ_{mn} in the referring graph)

and relations between the potential referents (i.e., an edge r_{xy} in the referent graph). It is defined by two individual compatibility functions as follows:

$$EdgeSim(r_{xy}, \gamma_{mn}) = SemType(r_{xy}, \gamma_{mn})\ Temp(r_{xy}, \gamma_{mn})$$

***SemType*(r_{xy}, γ_{mn})** encodes the semantic type compatibility between an edge in the referring graph and an edge in the referent graph. It is defined in Table 2. This constraint indicates that the relation held between referring expressions should be compatible with the relation held between two correct referents. For example, consider the utterance How much is this green house and this blue house. This utterance indicates that the referent to the first expression this green house should share the same semantic type as the referent to the second expression this blue house. As shown in Table 2, if the semantic type relations of r_{xy} and γ_{mn} are the same, $SemType(r_{xy}, \gamma_{mn})$ returns 1. If they are different, $SemType(r_{xy}, \gamma_{mn})$ returns zero. If either r_{xy} or γ_{mn} is unknown, then it returns 0.5.

***Temp*(r_{xy}, γ_{mn})** captures the temporal compatibility between an edge in the referring graph and an edge in the referent graph. It is defined in Table 3. This constraint indicates that the temporal relationship between two referring expressions (in one utterance) should be compatible with the relations of their corresponding referents as they are introduced into the context (e.g., through gesture). The temporal relation between referring expressions (i.e., γ_{mn}) is either Precede or Concurrent. If the temporal relations of r_{xy} and γ_{mn} are the same, then $Temp(r_{xy}, \gamma_{mn})$ returns 1. Because potential references could come from prior conversation, even if r_{xy} and γ_{mn} are not the same, the function does not return zero when γ_{mn} is Precede.

Next, we discuss how these definitions and the process of graph matching address optimization, in particular, with respect to key principles of Optimality Theory for natural language interpretation.

3.3 Optimality Theory

Optimality Theory (OT) is a theory of language and grammar, developed by Alan Prince and Paul Smolensky (Prince and Smolensky, 1993). In Optimality Theory, a grammar consists of a set of well-formed constraints. These constraints are applied simultaneously to identify linguistic

$SemType(r_{xy}, \gamma_{mn})$		r_{xy}		
		Same	Different	Unknown
γ_{mn}	Same	1	0	0.5
	Different	0	1	0.5
	Unknown	0.5	0.5	0.5

Table 2: Definition of $SemType(r_{xy}, \gamma_{mn})$

$Temp(r_{xy}, \gamma_{mn})$		r_{xy}			
		Preceding	Concurrent	Non-concurrent	Unknown
γ_{mn}	Precede	1	0.5	0.7	0.5
	Concurrent	0	1	0	0.5

Table 3: Definition of $Temp(r_{xy}, \gamma_{mn})$

structures. Optimality Theory does not restrict the content of the constraints (Eisner 1997). An innovation of Optimality Theory is the conception of these constraints as soft, which means violable and conflicting. The interpretation that arises for an utterance within a certain context maximizes the degree of constraint satisfaction and is consequently the best alternative (hence, optimal interpretation) among the set of possible interpretations.

The key principles or components of Optimality Theory can be summarized as the following three components (Blutner 1998): 1) Given a set of input, Generator creates a set of possible outputs for each input. 2) From the set of candidate output, Evaluator selects the optimal output for that input. 3) There is a strict dominance in term of the ranking of constraints. Constraints are absolute and the ranking of the constraints is strict in the sense that outputs that have at least one violation of a higher ranked constraint outrank outputs that have arbitrarily many violations of lower ranked constraints. Although Optimality Theory is a grammar-based framework for natural language processing, its key principles can be applied to other representations. At a surface level, our approach addresses these main principles.

First, in our approach, the matching matrix $P(a_x, \alpha_m)$ captures the probabilities of all the possible matches between a referring node α_m and a referent node a_x. The matching process updates these probabilities iteratively. This process corresponds to the Generator component in Optimality Theory.

Second, in our approach, the satisfaction or violation of constraints is implemented via return values of compatibility functions. These

	G1: No Gesture	**G2**: One Gesture	**G3**: Multi-Gestures	Total Num
S1:No referring expression	1(1), 0(0)	3(1), 0(0)	0	4(2), 0(0)
S2: One referring expression	6(4), 5(2)	96(89), 58(21)	8(7), 11(2)	110(90), 74(25)
S3: Multiple referring expressions	0(0), 1(0)	3(1), 7(1)	12(8), 8(0)	15(9), 16(1)
Total Num	7(5), 6(2)	102(91), 65(22)	20(15), 19(2)	129(111) 90(26)

Table 4: Evaluation Results. In each entry form "a(b), c(d)", "a" indicates the number of inputs in which the referring expressions were correctly recognized by the speech recognizer; "b" indicates the number of inputs in which the referring expressions were correctly recognized and were correctly resolved; "c" indicates the number of inputs in which the referring expressions were not correctly recognized; "d" indicates the number of inputs in which the referring expressions also were not correctly recognized, but were correctly resolved. The sum of "a" and "c" gives the total number of inputs with a particular combination of speech and gesture.

constraints can be violated during the matching process. For example, functions $Id(a_x,\alpha_m)$, $SemType(a_x,\alpha_m)$, and $Attr_k(a_x,\alpha_m)$ return zero if the corresponding intended constraints are violated. In this case, the overall similarity function will return zero. However, because of the iterative updating nature of the matching algorithm, the system will still find the most optimal match as a result of the matching process even some constraints are violated. Furthermore, A function that never returns zero such as $Temp(a_x,\alpha_m)$ in the node similarity function implements a gradient constraint in Optimality Theory. Given these compatibility functions, the graph-matching algorithm provides an optimization process to find the best match between two graphs. This process corresponds to the Evaluator component of Optimality Theory.

Third, in our approach, different compatibility functions return different values to address the Constraint Ranking component in Optimality Theory. For example, as discussed earlier, once a_x and α_m share the same identifier, $Id(a_x,\alpha_m)$ returns 100. If a_x and α_m share the same semantic type, $SemType(a_x,\alpha_m)$ returns 1. Here, we consider the compatibility between identifiers is more important than the compatibility between semantic types. However, currently we have not yet addressed the strict dominance aspect of Optimality Theory.

3.4 Evaluation

We conducted several user studies to evaluate the performance of this approach. Users could interact with our system using both speech and deictic gestures. Each subject was asked to complete five tasks. For example, one task was to find the cheapest house in the most populated town. Data from eleven subjects was collected and analyzed.

Table 4 shows the evaluation results of 219 inputs. These inputs were categorized in terms of the number of referring expressions in the speech input and the number of gestures in the gesture inputs. Out of the total 219 inputs, 137 inputs had their referents correctly interpreted. For the remaining 82 inputs in which the referents were not correctly identified, the problem did not come from the approach itself, but rather from other sources such as speech recognition and language understanding errors. These were two major error sources, which were accounted for 55% and 20% of total errors respectively (Chai et al. 2004b).

In our studies, the majority of user references were simple in that they involved only one referring expression and one gesture as in earlier findings (Kehler 2000). It is trivial for our approach to handle these simple inputs since the size of the graph is usually very small and there is only one node in the referring graph. However, we did find 23% complex inputs (the row S3 and the column G3 in Table 4), which involved multiple referring expressions from speech utterances and/or multiple gestures. Our optimization approach is particularly effective to interpret these complex inputs by simultaneously considering semantic, temporal, and contextual constraints.

4 Conclusion

As in natural language interpretation addressed by Optimality Theory, the idea of optimizing constraints is beneficial and there is evidence in favor of competition and constraint ranking in multimodal language interpretation. We developed a graph-based approach to address optimization for multimodal interpretation; in particular, interpreting multimodal references. Our approach simultaneously applies temporal, semantic, and contextual constraints together and achieves the best interpretation among all alternatives. Although currently the referent graph corresponds to gesture

input and conversation context, it can be easily extended to incorporate other modalities such as gaze inputs.

We have only taken an initial step to investigate optimization for multimodal language processing. Although preliminary studies have shown the effectiveness of the optimization approach based on graph matching, this approach also has its limitations. The graph-matching problem is a NP complete problem and it can become intractable once the size of the graph is increased. However, we have not experienced the delay of system responses during real-time user studies. This is because most user inputs were relatively concise (they contained no more than four referring expressions). This brevity limited the size of the graphs and thus provided an opportunity for such an approach to be effective. Our future work will address how to extend this approach to optimize the overall interpretation of user multimodal inputs.

Acknowledgements

This work was partially supported by grant IIS-0347548 from the National Science Foundation and grant IRGP-03-42111 from Michigan State University. The authors would like to thank John Hale and anonymous reviewers for their helpful comments and suggestions.

References

Bolt, R.A. 1980. Put that there: Voice and Gesture at the Graphics Interface. *Computer Graphics*, 14(3): 262-270.

Blutner, R., 1998. Some Aspects of Optimality In Natural Language Interpretation. *Journal of Semantics*, 17, 189-216.

Cassell, J., Bickmore, T., Billinghurst, M., Campbell, L., Chang, K., Vilhjalmsson, H. and Yan, H. 1999. Embodiment in Conversational Interfaces: Rea. In *Proceedings of the CHI'99 Conference*, 520-527.

Chai, J., Prasov, Z, and Hong, P. 2004b. Performance Evaluation and Error Analysis for Multimodal Reference Resolution in a Conversational System. *Proceedings of HLT-NAACL 2004 (Companion Volumn)*.

Chai, J. Y., Hong, P., and Zhou, M. X. 2004a. A Probabilistic Approach to Reference Resolution in Multimodal User Interfaces, *Proceedings of 9th International Conference on Intelligent User Interfaces (IUI)*: 70-77.

Chai, J., Pan, S., Zhou, M., and Houck, K. 2002. Context-based Multimodal Interpretation in Conversational Systems. *Fourth International Conference on Multimodal Interfaces*.

Cohen, P., Johnston, M., McGee, D., Oviatt, S., Pittman, J., Smith, I., Chen, L., and Clow, J. 1996. Quickset: Multimodal Interaction for Distributed Applications. *Proceedings of ACM Multimedia.*

Eisner, Jason. 1997. Efficient Generation in Primitive Optimality Theory. *Proceedings of ACL'97*.

Gold, S. and Rangarajan, A. 1996. A Graduated Assignment Algorithm for Graph-matching. *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 4.

Gustafson, J., Bell, L., Beskow, J., Boye J., Carlson, R., Edlund, J., Granstrom, B., House D., and Wiren, M. 2000. AdApt – a Multimodal Conversational Dialogue System in an Apartment Domain. *Proceedings of 6th International Conference on Spoken Language Processing (ICSLP)*.

Johnston, M, Cohen, P., McGee, D., Oviatt, S., Pittman, J. and Smith, I. 1997. Unification-based Multimodal Integration, *Proceedings of ACL'97*.

Johnston, M. 1998. Unification-based Multimodal Parsing, *Proceedings of COLING-ACL'98*.

Johnston, M. and Bangalore, S. 2000. Finite-state Multimodal Parsing and Understanding. *Proceedings of COLING'00*.

Johnston, M., Bangalore, S., Visireddy G., Stent, A., Ehlen, P., Walker, M., Whittaker, S., and Maloor, P. 2002. MATCH: An Architecture for Multimodal Dialog Systems, *Proceedings of ACL'02*, Philadelphia, 376-383.

Kehler, A. 2000. Cognitive Status and Form of Reference in Multimodal Human-Computer Interaction, *Proceedings of AAAI'01*, *685-689*.

Koons, D. B., Sparrell, C. J. and Thorisson, K. R. 1993. Integrating Simultaneous Input from Speech, Gaze, and Hand Gestures. In *Intelligent Multimedia Interfaces*, M. Maybury, Ed. MIT Press: Menlo Park, CA.

Neal, J. G., and Shapiro, S. C. 1991. Intelligent Multimedia Interface Technology. In *Intelligent User Interfaces*, J. Sullivan & S. Tyler, Eds. ACM: New York.

Oviatt, S. L. 1996. Multimodal Interfaces for Dynamic Interactive Maps. In *Proceedings of Conference on Human Factors in Computing Systems: CHI '96*, 95-102.

Oviatt, S., DeAngeli, A., and Kuhn, K., 1997. Integration and Synchronization of Input Modes during Multimodal Human-Computer Interaction, In *Proceedings of Conference on Human Factors in Computing Systems: CHI '97*.

Oviatt, S., Coulston, R., Tomko, S., Xiao, B., Bunsford, R. Wesson, M., and Carmichael, L. 2003. Toward a Theory of Organized Multimodal Integration Patterns during Human-Computer Interaction. In *Proceedings of Fifth International Conference on Multimodal Interfaces*, 44-51.

Prince, A. and Smolensky, P. 1993. Optimality Theory. Constraint Interaction in Generative Grammar. ROA 537. http://roa.rutgers.edu/view.php3?id=845.

Stent, A., J. Dowding, J. M. Gawron, E. O. Bratt, and R. Moore. 1999. The Commandtalk Spoken Dialog System. *Proceedings of ACL'99*, 183–190.

Tsai, W.H. and Fu, K.S. 1979. Error-correcting Isomorphism of Attributed Relational Graphs for Pattern Analysis. *IEEE Transactions on Systems, Man and Cybernetics.*, vol. 9.

Wahlster, W., 1998. User and Discourse Models for Multimodal Communication. *Intelligent User Interfaces, M. Maybury and W. Wahlster (eds.)*, 359-370.

Wu, L., Oviatt, S., and Cohen, P. 1999. Multimodal Integration – A Statistical View, *IEEE Transactions on Multimedia*, Vol. 1, No. 4, 334-341.