Multimodal Dialog Management

L.J.M. Rothkrantz, P. Wiggers, F. Flippo, D.Woei-A-Jin, R.J. van Vark

Data and Knowledge Systems Group
Delft University of Technology
Mekelweg 4, 2628 CD Delft, The Netherlands
l.j.m.rothkrantz@ewi.tudelft.nl
p.wiggers@ewi.tudelft.nl

Abstract. Unreliable speech recognition, especially in noisy environments and the need for more natural interaction between man and machine have motivated the development of multimodal systems using speech, pointing, gaze, and facial expressions. In this paper we present a new approach to fuse multimodal information streams using agents. A general framework based on this approach that allows for rapid application development is described. Since anaphora very often occur in natural discourse a special agent for anaphora resolution was developed within this framework.

1 Introduction

In the development of spoken dialog systems a gradual growth in complexity and naturalness of interaction can be witnessed. One of the dimensions along which these systems can be differentiated is the locus of control. In the simplest cases control either lies with the system or with the user. In more advanced schemes control is distributed among both parties, leading to mixed initiative systems. In this case, the dialog system tries to fill a frame with information provided by the user, displaying prompts that may vary from commands to gentle attempts to persuade the user to provide the necessary information, while possibly at the same time verifying the correctness of the information extracted from earlier inputs. Despite the questions asked by the system, the user can choose to provide a different piece of information. Mixed initiative dialogs end when a frame has been filled.

McTear [6] identifies another type of dialogue system: agent-based. These systems go beyond cooperating with the user to fill a frame; rather, they attempt to solve a problem together with the user. The user and the system exchange knowledge and reason about their own actions and beliefs, as well as each other's input.

Another thread of research involved with the dialog between human and machine focuses on multimodal interaction, of which natural language, be it spoken or written, may be part. The benefit of exploiting multiple modalities is twofold: mutual disambiguation and naturalness.

Mutual disambiguation is the act of using information from one modality to fill in or correct missing or ambiguous information in another modality: the

weakness of speech is compensated by the use of gesture, and vice versa. By using information from another source as well, such as images of the speaker's lip movements while speaking or gesture information on where the user was pointing on a map, the system can be more certain of what the speaker intended.

The reason multimodal interfaces are more natural than either traditional WIMP interfaces or even unimodal speech interfaces, is that humans communicate multimodally. Our brain is designed to process multiple streams of information to assess the state of the world. This is why we use our hands when we speak and reflect the semantics of what we are saying in our facial expressions. For the same reason we tend to pay more attention to someone's face when talking in noisy situations and we find understanding people on the telephone harder than in a face-to-face conversation. Similarly, people instinctively use the most appropriate modality or combination of modalities for a task and switch to another set of modalities when a command is not understood the first time around [7]. This self-correcting behavior results in better performance and less frustration compared to a situation in which users are constrained to using a single modality that may not be optimal for the task at hand.

However, despite the availability of multimodal devices, there are very few commercial multimodal applications available. One reason for this maybe the lack of a framework to support development of multimodal applications in reasonable time and with limited resources. In this paper we will describe an agent-based framework enabling rapid development of applications using a variety of modalities and methods for ambiguity resolution, featuring a novel approach to multimodal fusion. Furthermore, a module for anaphora resolution within this framework will be described.

2 Related Work

Multimodal interfaces have enjoyed a great deal of attention in recent years and several multimodal frameworks have been proposed. Perhaps the earliest work on multimodal interfaces is that of Bolt [2] in 1980. His system provided an interface in which shapes could be manipulated using a combination of speech and pointing, with commands like "put that to the left of the green triangle". Fusion and reference resolution was done at the parse level: every time an anaphor or deictic reference was recognized, the system would immediately establish where the user was pointing and resolve the reference. While performing fusion directly on recognition of a reference yields a straightforward implementation of fusion, it is hardly satisfactory, as gestures and speech are in general not synchronized.

Krahnstoever [4] describes a multimodal framework targeted specifically at fusing speech and gesture with output being done on large screen displays. Several applications are described that have been implemented using this framework. The fusion process is not described in great detail, but appears to be optimized for and limited to integration of speech and gesture, using inputs from cameras that track a user's head and hands.

The W3C has set up a multimodal framework specifically for the web [5]. Rather than an implementation of a multimodal framework, it proposes a set of properties and standards - specifically the Extensible Multimodal Annotation Markup Language (EMMA) - that a multimodal architecture should adhere to.

In the multimodal framework of the Smartkom project [11] the user interacts with a lifelike agent mainly through speech and gestures. The framework is knowledge based, modular and application independent. The main component for reference resolution and fusion of incoming information is the discourse memory which is a three-tiered representation model that allows multiple modalities to refer to the same object at the discourse level.

The QuickSet system [3], built by the Oregon Graduate Institute, integrates pen with speech to create a multimodal system. The system employs a Members-Teams-Committee technique very similar to the fusion technique described in this paper, using parallel agents to estimate a posteriori probabilities for various possible recognition results, and weighing them to come to a decision. However, our approach is more reusable as it separates the data and feature acquisition from recognition and supports a variety of simultaneous modalities beyond pen and speech.

3 Design of the Multimodal Dialog System

The general architecture of the multimodal system is presented in Fig. 1. As is clear from this figure, the approach is speech centric, or rather language centric as text can come from either a speech recognizer or be typed on a keyboard. Language is the main modality, and other modalities are used to resolve deictic references, pronouns, and other anaphora as well as ellipsis in the text. Currently, the speech recognizer provides the fist best hypothesis annotated with

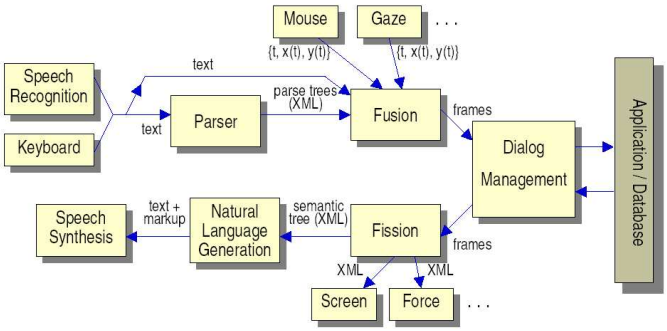

Fig. 1. Multimodal Dialog System

time boundary information to the parser, but given the modular structure of the

framework this is not a fundamental limitation of the system and more advanced recognizers providing word lattices may be used.

3.1 An Object-Oriented Framework

The scheme of Fig. 1 has been implemented as a framework [9], in other words it provides a core of common functionality that every multimodal dialog system needs, while the task specific parts of the system can be plugged in by a developer to produce a custom application. In this, the framework follows the object-oriented philosophy of inversion of control, or less respecful, of "old code calls new code". The framework core calls the plugged-in components and ensures proper communication between them. This allows for easy and rapid application development as the developer does not need to have knowledge of the frameworks internals, but only needs to implement the interfaces to the framework. Configuration of the implemented framework is largely declarative: the user specifies structure, the "what" knowledge, not procedure, the "how" knowledge.

3.2 Fusion

The framework features a new approach to fusion that is reusable across applications and modalities. The process is depicted in Fig. 2. The input to the fusion process is a semantic parse tree of concepts with time stamps as generated by the natural language parser component of the speech interface. This parse tree needs to be transformed into frames that the dialog manager can use to make calls to the application. To accomplish this, the natural language concepts in the parse tree need to be mapped to application concepts. In addition, ambiguity needs to be resolved. Ambiguity exists when the user uses pronouns or deictic references, for example "remove *that*", or "tell me more about *it*". Another case of ambiguity is ellipsis, in which words that are implied by context are omitted, such as "rotate this clockwise ... *and this too*".

Resolving agents operate on the parse tree to realize the aforementioned mapping of concepts and resolution of ambiguity. The framework does not specify the implementation details of resolving agents. All that is expected is that the agents take a fragment from the parse tree, perform some transformation on it, and use it to fill a slot in the semantic frame that is sent to the dialog manager. The agents can use data from a modality through an access object called *context provider*, to give them a context in which to perform their task. Context providers can provide data from an external sensor, such as a gaze tracker, but also from more abstract data sources such as dialog history or application state (e.g. which toolbox button is selected). For example, an agent performing pronoun resolution might have access to gaze or gesture input to resolve a pronoun to an object on the screen that the user pointed to or looked at. Any agent will typically have access just one such input. This keeps the design of the agents simple, as they do not need to be concerned with combining data from multiple sources. This

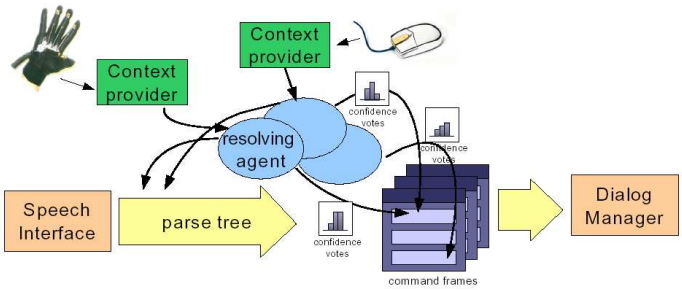

Fig. 2. Fusion

combination is done by the fusion manager. It is possible for resolving agents to share the same modality.

The agents themselves do not actually perform fusion. Their task is to perform an assessment of what they think the contents of a slot in the frame should be. Each agent will provide zero or more possible solutions with corresponding probability scores. The whole of the solutions provided by all agents will finally determine what the slot will contain. To make resolving agents reusable, the resolution process is separated from the acquisition of data from modalities. The resolution process is implemented in the resolving agents, while the acquisition of data is the responsibility of the context providers. Resolving agents merely specify the type of data they expect to receive from their context provider. In this way, an agent that requires (x,y)-data points to do its work can accept data from any context provider that provides (x,y)-data, such as a mouse, a gaze tracker, or a haptic glove. In a system with a mouse and a gaze tracker, for instance, two copies of the same pronoun resolution agent might be active, one using data from the mouse, and another using data from the gaze tracker. Each will give its resolutions along with corresponding probability scores, based on the data they have access to.

Thus, resolving agents operate *locally* with only the information they have access to, namely the fragment of the parse tree they use and the data they receive from their modality, if any. However, all agents together create a *global* result that takes into account all of the parse tree and all of the available modalities. Because each resolving agent works independently of the others, the agents can work in parallel, taking advantage of multiprocessor hardware to increase performance.

Context providers provide timestamps along with their data. These can be used by the resolvers so select data that are applicable to the parse tree fragment they are handling, using the timestamps that the natural language parser provides. For instance, the pronoun resolver agent mentioned before will look at data points that were generated around the time that the pronoun was spoken. Timestamps for speech data and context data ensure that the modality streams are properly synchronized.

The *fusion manager* controls the fusion process. It spawns resolving agents and passes them parse tree fragments to work with, takes the possible values for each slot from the agents and makes a decision based on the probability scores provided and the weights assigned to the resolving agents. Finally it merges frames from the conversation interface with method calls from the applications GUI, resolving ambiguities to create a frame with unambiguous meaning.

4 An Anaphora Resolution Agent

As an example of a resolving agent we will discuss an agent for anaphora resolution that was recently implemented. The application considered here is a multimodal interface for an electronic device, for example a multimodal interface for television program recording. It is possible to do all tasks hands-free with spoken input only, or control the device with pointing input in addition to speech. Visual feedback is used for displaying information and spoken output to guide the user through the dialogue.

Much research has been done on anaphora resolution in (computational) linguistics and natural language processing. Unfortunately, not all of the theories and models developed here are equally well-suited for automatic processing of spoken language. Reference resolution methods such as centering theory [1], often presuppose higher order information such as syntactic roles like subject and direct object or even semantic knowledge to infer how well a potential referee would fit in a sentence. For specific domains, modules could be plugged into the framework that provide this information, but this does not solve other more pervasive problems related to the very nature of speech itself. These problems occur in the shape of ungrammaticalities in spoken language, such as relative free constituent order, restarts, corrections and hesitations as well as in the form of recognition errors.

Such difficulties make syntactic and semantic analysis to the levels required for reference resolution a hard task and affect the performance of the anaphora resolution module itself as well. Therefore, in our current work [10], we used a statistical shallow semantic parser that does not provide any syntactic information but extracts phrases meaningful for the task at hand. The grammar rules specify the concepts used in the application and the possible ways they may be realized in an utterance as well as fillers that define word patterns not meaningful to the application.

To do robust anaphora resolution within this framework and in the presence of possible recognition errors we adopted the "Never look back" strategy of [8] for our agent. This model is based on the notion that the preference for referents can be determined by the attentional state of the hearer which in turn strongly correlates with the recency of entities in the discourse. Discourse entities are grouped into three categories: hearer-old discourse entities, mediated discourse entities and hearer-new discourse entities. Hearer-old discourse entities are entities already in the discourse model of the hearer, mediated discourse entities are linked to entities already in the discourse model, and hearer-new

discourse entities are not yet in the discourse model. The entities are ordered according to a preference relation [10].

In the original model of [8] entities are removed from focus if they are not used in the utterance under consideration. To tailor the approach more to the application at hand entities are not removed during the system turn, only during the user turn. This is done because the user can ignore the system output and refer back to what he said earlier.

Even though no syntactic information is necessary to determine the preferred referent, still some information about dependencies between several phrases is needed to determine which referents can or cannot be referred to considering the context of a sentence. To compensate for the lack of syntactic information, three general solutions are proposed to determine the dependencies between two concepts:

The first is to look at the properties of the target concept, and match them with a set of premises stated by the source concept. If these premises hold, it is assumed that the source concept modifies the target concept, and additional constraints can be added. This method is best used when two concepts do not necessarily follow each other directly, and it is possible to have other 'non-related' concepts between them.

The second method is to have the grammar treat the concepts as a single concept. A filter later extracts the two concepts, and assigns the concept which modifies the other as a subconcept of the other one. This is especially useful when the two concepts always follow each other directly. It is easier to determine the relation this way, and misassignments are less likely to occur.

The third method is used when a compound reference occurs. A reference refers to a property of another concept, which is a reference itself. Usually these concepts occur directly after each other, so a similar approach as mentioned above can be used. The concepts are grouped together by the grammar and a filter extracts the different concepts, and assigns the concept which contains the property which the other concept refers to as the superconcept of the other. The superconcept is resolved first, and is used to resolve the other concept.

4.1 Test Results

Tests of the system proved that reference resolution is indeed hampered by recognition errors that may introduce non-existent concepts or delete relevant words. In particular, the speech recognizer has trouble recognizing certain words, which are important for reference resolution, e.g. *he*, and the definite article *the*. At other times references are wrongfully introduced. Typically, the pronouns *it* and *its* or the demonstratives *that*, *this* and *them*. It was found however that recognition errors do not really create strange shifts in the focus of attention of the system, which would cause correctly recognized references to be resolved wrongly. During the tests, some misrecognitions contained references that were resolved to the concept in focus, so no shifts in focus did occur. Also when the system would move away from the desired task, for example displaying a totally

different topic, the user would typically try to move back to the task at hand, rather than just relentlessly trying to have the system recognize the utterance.

5 Conclusions and Future Work

In this paper a generic framework for multimodal human-machine interaction was presented. The framework is language centric and uses agents to process multimodal input and resolve ambiguities. As an example of such an agent a reference resolution agent was discussed that does not rely on extensive syntactic and semantic knowledge to do its job. During evaluation of the system, no resolution errors could be traced to errors in determining the dependencies between the concepts. In the online test, many errors are generated by misrecognition of the user by the system. Future work will address the tuning of the entire system to be more robust for recognition errors. In particular, the contextual knowledge present in the context providers and resolving agents as well as in the dialog frames could be fed back to the speech recognizer to constrain its language model. Furthermore, the fusion module may be extended to better deal with uncertainty in the output provided by the parser and the agents for example using a Bayesian network approach.

References

1. S.E. Brennan et al., A centering approach to pronouns, Proc. of the Association for Computational Linguistics, pp. 155-162, July 1987.
2. Bolt, R., A., "Put-that-there": Voice and gesture at the graphics interface. Computer Graphics (SIGGRAPH '80 Proceedings), 14(3):262-270, July 1980.
3. Cohen, P., Johnston, M., NcGee, D., Oviatt, S., Pittman, J., Smith I., Chen L., Clow, J., Quickset: Multimodal interaction for distributed applications. ACM International Multimedia Conference, New York: ACM, pages, 31-40,1997.
4. Krahnstoever, N., Kettebekov, S., Yeasin, M., and Sharma, R., A real-time framework for natural multimodal interaction with large screen displays, in Proc. of Fourth Intl. Conference on Multimodal Interfaces (ICMI 2002), Pittsburgh, PA, USA, October 2002.
5. Larson, J.A., Raman, T.V., W3C multimodal interaction framework, http://www.w3.org/TR/mmi-framework, 2 December 2002, W3C Note.
6. McTear, M., F., Spoken dialog technology: enabling the conversational interface. ACM Computing Surveys, 34(1):90 - 169, March 2002.
7. Oviatt, S., Designing robust multimodal systems for diverse users and environments. In Workshop on universal accessibility of ubiquitous computing: providing for the elderly, 2001.
8. M. Strube. "Never Look Back: An Alternative to Centering", Proceedings of ACL-98, pages 1251-1257, 1998.
9. Flippo, F., A Natural Human-Computer Interface for Controlling Wheeled Robotic Vehicles, Technical Report DKS04-02, Delft University of Technology, 2004.
10. Woei-A-Jin, J.R.L.D., Reference Resolotion in a Speech Recognition Environment, Technical Report DKS04-01, Delft University of Technology, 2004.
11. Wahlster W., Reithinger N., Blocher A., SmartKom: Multimodal Communication with a Life-Like Charakter, in Proc. of Eurospeech'01, Aalborg, Denmark, 2001.