An Input-Parsing Algorithm Supporting Integration of Deictic Gesture in Natural Language Interface

Yong Sun[1,2], Fang Chen[1,2], Yu Shi[1], and Vera Chung[2]

[1] National ICT Australia, Australian Technology Park
Eveleigh NSW 1430, Australia
[2] School of IT, The University of Sydney,
NSW 2006, Australia
yong.sun@nicta.com.au

Abstract. Natural language interface (NLI) enables an efficient and effective interaction by allowing a user to submit a single phrase in natural language to the system. Free hand gestures can be added to an NLI to specify the referents for deictic terms in speech. By combining NLI with other modalities to a multimodal user interface, speech utterance length can be reduced, and users need not clearly specify the referent verbally. Integrating deictic terms with deictic gestures is a critical function in multimodal user interface. This paper presents a novel approach to extend chart parsing used in natural language processing (NLP) to integrate multimodal input based on speech and manual deictic gesture. The effectiveness of the technique has been validated through experiments, using a traffic incident management scenario where an operator interacts with a map on large display at a distance and issues multimodal commands through speech and manual gestures. The preliminary experiment of the proposed algorithm shows encouraging results.

Keywords: Multimodal chart parsing, Multimodal Fusion, Deictic Gesture, Deictic Terms.

1 Introduction

NLI refers to the interface that allows a user to interact with a system using natural written or spoken language. It enables an efficient and effective interaction by allowing a user to submit a single phrase in natural language to the system. Although NLI is very attractive, it is often difficult to implement it due to the unpredictable nature and ambiguity of natural language. After applying some constrains on the language structures and lexicon, some restricted language interfaces still maintain most of the advantages of NLI; but a user will be required to adapt these restrictions in them. With these facts, an NLI can be redefined as an interface that allows a user to interact with a system using written or spoken language without explicitly learning the commands in it.

Speaking commands to a computer frees up a user's hands for other tasks; and an NLI will be more efficient when combined with other interaction modalities such as,

J. Jacko (Ed.): Human-Computer Interaction, Part III, HCII 2007, LNCS 4552, pp. 206–215, 2007.

hand gesture, body gesture, head gesture and eye gaze etc. By combining speech with other modalities, an NLI can capture the additional cues for disambiguation; and the bandwidth of the interaction between a user and the NLI is broadened.

Free hand gestures can be added to an NLI to specify the referents for deictic terms in speech. With such additional cues, speech utterance length can be reduced, and users need not explicitly specify the referent verbally e.g. “Watch the camera in the section of George street and Eddy avenue” can be reduced by “Watch this camera” <pointing>.

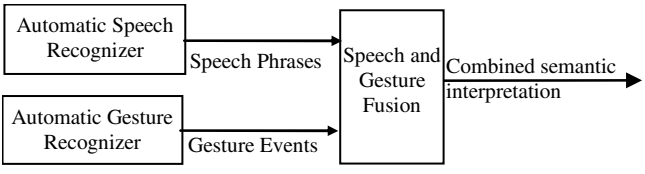

Fig. 1. Integrate Gestures with Speech in Parsing Process

Fig. 1 illustrates a structure to integrate speech with gesture in parsing process in an NLI. This fusion ability allows the system to utilize gesture cues in parsing deictic terms and disambiguation. With this capability, commands such as "Watch this <pointing>" and "Send this <pointing> there <pointing>", are feasible. However, by combining an NLI with other modalities to form a multimodal user interface (MMUI), more challenges emerge.

In MMUI, most multimodal inputs are not linearly ordered. For a multimodal command, multimodal input do not always follow the same order. Different input modalities such as speech and gestures can be used at any time in any order by the user to convey information. For example, in a traffic incident control room, when an operator points to a camera icon on a map and says “play this camera”, he/she wants to play a specific camera identified by the hand pointing. The order and timing of when “play this camera” and the pointing gesture occurred and was recognized by speech and gesture recognizers can be different from person to person. The objective of multimodal input parsing, a critical component in an MMUI, is to find the most consistent semantic interpretation when multiple inputs are temporally and/or semantically aligned. In the above example, multimodal parsing should provide as output the joint meaning of both playing the camera and pointing by the hand. The main challenge for multimodal input parsing lies in developing a logic-based or pattern-matching technique that can integrate semantic information derived from different input modalities into a representation with a common meaning.

The multimodal input in our application is discrete, which means it can be individually treated as token in time and modality. Both speech and gesture inputs are recognized as tokens. For example, the phrase ‘the end of the street’ represents 5 tokens. In one multimodal turn, all multimodal tokens belong to one multimodal utterance. The interpretation of multimodal input, semantic interpretation, is a coherent piece of information for the computer to act upon during human computer interaction.

We propose a new approach termed Mountable Unification-based Multimodal Input Fusion (MUMIF) to integrate gesture with deictic terms in speech. The architecture and implementation of the approach are introduced in [8]. This paper focuses on the parsing algorithm of MUMIF. It can seamlessly integrate individual interpretation provided by speech and gesture recognizers, and provide a joint or combined semantic interpretation of the user's intension. The proposed multimodal chart parsing algorithm is based on chart parsing used in NLP, with the novelties in parsing multimodal inputs. The algorithm provides an alternative method in unification-based multimodal parsing by introducing grammatical consecution conception. To test the effectiveness and performance of the algorithm, we used an MMUI research platform, called PEMMI, developed by National ICT Australia [2]. PEMMI was built for transport planning and traffic incident management applications. It is mostly implemented in Java and composed of a speech recognition module, a vision-based gesture recognition module, a simple state-machine based multimodal input parsing module, a dialog manager module, and output generation module. We removed the state-machine based parsing module in PEMMI and plugged in the integrating module equipped with the proposed algorithm. PEMMI served both as research platform to fine tune the performance of the algorithm, and as a test platform for multimodal parsing evaluation.

2 Related Work

There are two main approaches to integrate speech with gesture inputs in the literature; one is finite-state based and the other is unification-based.

The finite state based approach was adopted in [5]. It uses a finite state device to encode the multimodal integration pattern, the syntax of speech inputs, gesture inputs and the semantic interpretation of these multimodal inputs. Recently, in [7], a similar approach, which utilizes a modified temporal augmented transition network, is reported.

In the unification-based approach, the fusion module applies a unification operation on a speech and gesture input according to a multimodal grammar. It can be found in many documents, such as [4], [6] and [3].

This kind of approach can handle a versatile multimodal command style. However, it suffers from significant computational complexity [5]. Development of the grammar rules requires significant understanding of integration technique.

Unification-based parsing approaches use various algorithms to parsing multimodal input. [4] assumes multimodal input is not discrete and linearly ordered. A multidimensional parsing algorithm runs bottom-up from the input elements, building progressively larger constituents in accordance with the rule set. [3] also assumes multimodal input is not linearly ordered, Multimodal parsing is performed on a pool of elements, where new elements can be added and elements can be removed.

The MUMIF belongs to the unification-based approach. Both [4] and [6] agree the speech and gesture inputs are not linearly ordered. Further, we point out that inputs from one modality are linearly ordered. For example, in "Send this there <pointing1> <pointing2>", pointing1 always precedes pointing2.

With this observation, chart parsing in natural language processing is extended by parsing speech and gesture inputs separately at first, and then combining the parse edges from speech and gesture inputs according to speech-gesture combination rules in a multimodal grammar.

3 Chart Parser

The proposed multimodal parsing algorithm is based on chart parsing in NLP. In NLP, a grammar is a formal system that specifies which sequences of tokens are well formed in the language, and which provides one or more phrase structures for the sequence. For example, **S -> NP VP** says that a constituent of category **S** can consist of sub-constituents of categories **NP** and **VP** [1]. According to the productions of a grammar, a parser processes input tokens and builds one or more constituent structures which conform to the grammar.

A chart parser uses a structure called a *chart* to record the hypothesized constituents in a sentence. One way to envision this chart is as a graph whose nodes are the word boundaries in a sentence. Each hypothesized constituent can be drawn as an *edge*. For example, the chart in Fig. 2 hypothesizes that "hide" is a V (verb), "police" and "stations" are Ns (noun) and they comprise an NP (noun phrase).

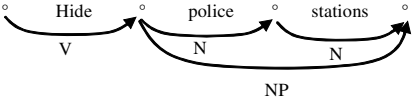

Fig. 2. A chart recording types of constituents in edges

To determine detailed information of a constituent, it is useful to record the types of its children. This is shown in Fig. 3.

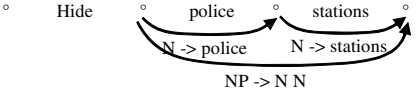

Fig. 3. A chart recording children types of constituents in an edge

If an edge spans the entire sentence, then the edge is called a *parse edge*, and it encodes one or more parse trees for the sentence. In Fig. 4, the verb phrase VP represented as [VP -> V NP] is a parse edge.

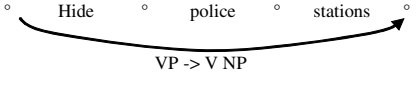

Fig. 4. A chart recording a parse edge

To parse a sentence, a chart parser uses different algorithms to find all *parse edges*.

4 Multimodal Chart Parser

To extend chart parser for multimodal input, the differences between unimodal and multimodal input need to be analyzed.

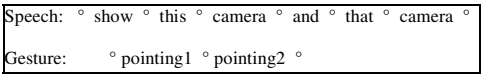

Fig. 5. Multimodal utterance: speech -- "show this camera and that camera" plus two pointing gestures. Pointing1: The pointing gesture pointing to the first camera. Pointing2: The pointing gesture pointing to the second camera.

The first difference is linear order. Tokens of a sentence always follow a same linear order. In a multimodal utterance, the linear order of tokens is variable, but the linear order of tokens from same modality is invariable. For example, as in Fig. 5, a traffic controller wants to monitor two cameras; he/she issues a multimodal command "show this camera and that camera" while pointing to two cameras with the cursor of his/her hand on screen. The gesture pointing1 and pointing2 may be issued before, in-between or after speech input, but pointing2 is always after pointing1.

The second difference is grammar consecution. Tokens of a sentence are consecutive in grammar; in other words, if any token of the sentence is missed the grammar structure of the sentence will not be preserved. In a multimodal utterance, tokens from one modality may not be consecutive in grammar. In Fig. 5, speech -- "show this camera and that camera" is consecutive in grammar. It can form a grammar structure though the structure is not complete. Gesture – "pointing1, pointing2" is not consecutive in grammar. Grammatically inconsecutive constituents are link with a list in the proposed algorithm. "pointing1, pointing2" is stored in a list.

Grammar structures of hypothesized constituents from each modality can be illustrated as in Fig. 6. Tokens from one modality can be parsed to a list of constituents [C1 … Cn] where n is the number of constituents. If the tokens are grammatically consecutive, then n=1, i.e., the Modality 1 parsing result in Fig. 6. If the tokens are not consecutive in grammar, then n>1. For example, in Fig. 6, there are 2 constituents for Modality 2 input.

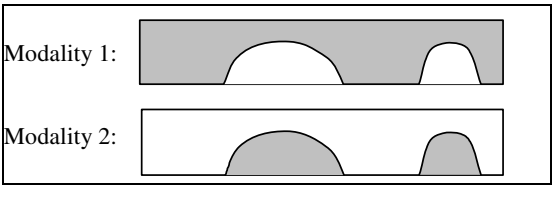

Fig. 6. Grammar structures formed by tokens from 2 modalities of a multimodal utterance. Shadow areas represent constituents which have been found. Blank areas are the expected constituents from another modality to complete a hypothesized category. The whole rectangle area represents a complete constituent for a multimodal utterance.

To record a hypothesized constituent that needs constituents from another modality to become complete, a vertical bar is added to the edge's right hand side. The constituents to the left of the vertical bar are the hypotheses in this modality. The constituents to the right of the vertical bar are the expected constituents from another modality; 'show this camera and that camera' can be expressed as **VP -> V NP | point, point**.

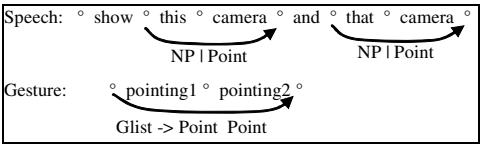

Fig. 7. Edges for "this camera", "that camera" and two pointing gestures

As in Fig. 7, edges for "this camera", "that camera" and "pointing1, pointing2" can be recorded as **NP | Point**, **NP | Point** and **Glist** respectively. Glist is a list of gesture events.

Then, from edges for "this camera" and "that camera", an **NP | Glist** can be derived in Fig. 8.

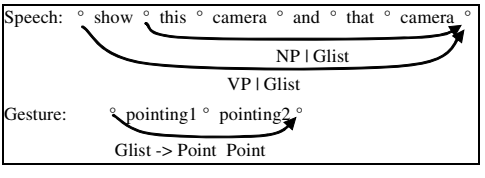

Fig. 8. Parse edges after hypothesizing "this camera" and "that camera" into an NP

Finally, parse edges that cover whole speech tokens and gesture tokens are generated as in Fig. 9. They are integrated to parse edge of the multimodal utterance.

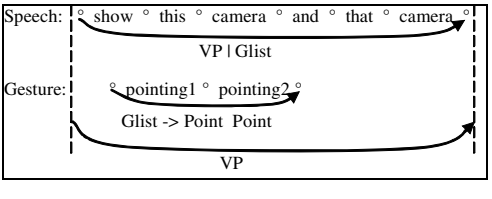

Fig. 9. Final multimodal parse edge and its children

So, a complete multimodal parse edge consists of constituents from different modalities. It has no more expected constituents.

As shown in Fig. 10, in the proposed multimodal chart parsing algorithm, to parse a multimodal utterance, speech and gesture tokens are parsed separately at first, and then the parse edges from speech and gesture tokens are parsed according to

speech-gesture combination rules in a multimodal grammar that provided lexical and rules for speech and gesture inputs, and speech-gesture combination rules.

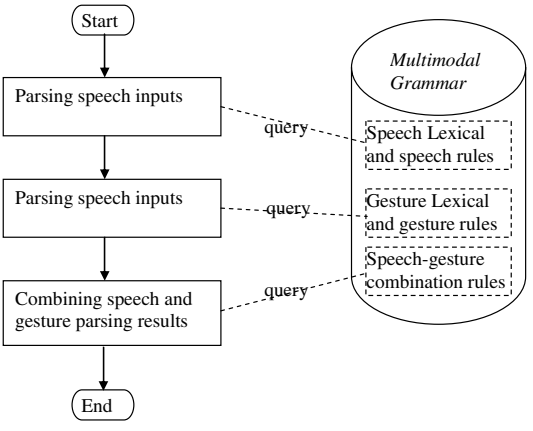

Fig. 10. Flow chart of proposed algorithm

5 Experiment and Analysis

To test the performance of the proposed multimodal parsing algorithm, an experiment has been designed and conducted to evaluate the applicability of the proposed multimodal chart parsing algorithm and the flexibility of multimodal chart parsing algorithm against different multimodal input orders.

5.1 Setup and Scenario

The evaluation experiment was conducted on a modified PEMMI platform. Fig. 11 shows the various system components involved in the experiment. ASR and AGR recognize signals captured by Microphone and Webcam, and provide parsing module with the recognized input. A dialog management module controls output generation according to a parsing result generated by parsing module.

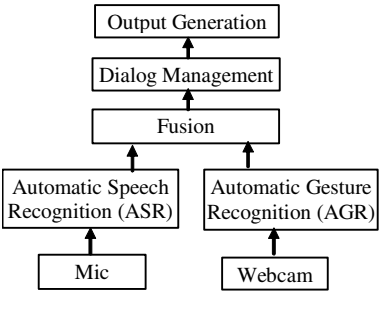

Fig. 11. Overview of testing experiment setup

Fig. 12 shows the user study setup of MUMIF algorithm, which is similar to the one, used in MUMIF experiment. A traffic control scenario was designed within an incident management task. In this scenario, a participant stands about 1.5 metres in front of a large rear-projection screen measuring 2x1.5 metres. A webcam mounted on a tripod, about 1 metre away from the participant, is used to capture manual gestures of the participant. A wireless microphone is worn by the participant.

Fig. 12. User study setup for evaluating MUMIF parsing algorithm

5.2 Preliminary Results and Analysis

During this experiment, we tested the proposed algorithm against a number of multimodal commands typical in map-based traffic incident management, such as

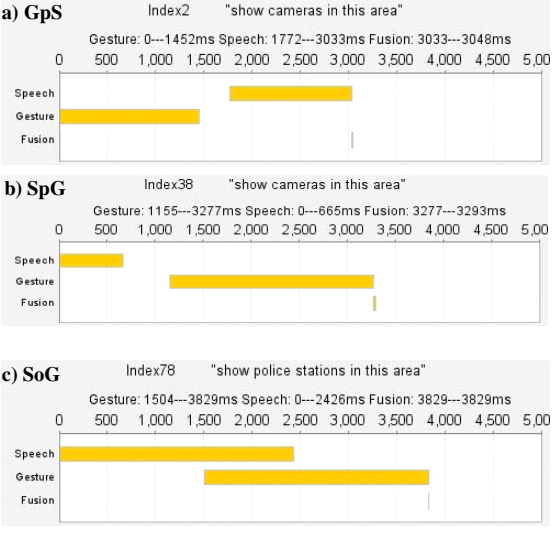

Fig. 13. Three multimodal input patterns

Table 1. Experiment results

Multimodal input pattern	Number of multimodal turns	Number of successful fusion
GpS	17	17
SpG	5	5
SoG	23	23

"show cameras in this area" with a circling/drawing gesture to indicate the area, "show police stations in this area" with a gesture drawing the area and "watch this" with a hand pause to specify the camera to play. One particular multimodal command, “show cameras in this area” with a gesture to draw the area, requires a test subject to issue the speech phrase and to draw an area using an on-screen cursor of his/her hand. The proposed parsing algorithm would generate a “show” action parameterized by the top-left and bottom-right coordinates of the area. In a multimodal command, multimodal tokens are not linearly ordered. Fig. 13 shows 3 of the possibilities of the temporal relationship between speech and gesture: GpS (Gesture precedes speech), SpG (Speech precedes gesture) and SoG (Speech overlaps gesture). The first bar shows the start and end time of speech input, the second for gesture input and the last (very short) for parsing process.

The proposed multimodal parsing algorithm worked in all these patterns (see Table 1).

6 Conclusion and Future Work

The proposed multimodal chart parsing is extended from chart parsing in NLP. By indicating expected constituents from another modality in hypothesized edges, the algorithm is able to handle multimodal tokens which are discrete but not linearly ordered. In a multimodal utterance, tokens from one modality may be consecutive in grammar. In this case, the hypothesised constituents are stored in a list to link them together.

By parsing unimodal input separately, the computation complexity of parsing is reduced. One parameter of computational complexity in chart parsing is the number of tokens. In a multimodal command, if there are m speech tokens and n gesture tokens, the parsing algorithm needs to search in $m+n$ tokens when the inputs are treated as a pool; when speech and gesture are treated separately, the parsing algorithm only needs to search in m speech tokens first and n gesture tokens second. The speech-gesture combination rules are more general than previous approaches. It does not care about the type of its speech daughter, only focus on the expected gestures.

Preliminary experiment result revealed that the proposed multimodal chart parsing algorithm can handle linearly unordered multimodal input and showed its promising applicability and flexibility in parsing multimodal input.

The proposed multimodal chart parsing algorithm is a work in progress. For the moment, it only processes the best interpretation from recognizers. In the future, to

develop a robust, flexible and portable multimodal input parsing technique, it will be extended to handle n-best list of inputs. The research of a semantic interpretation possibility can also be a pending topic.

References

1. Bird, S., Klein, E., Loper, E.: Parsing (2005) In http://nltk.sourceforge.net
2. Chen, F., Choi, E., Epps, J., Lichman, S., Ruiz, N., Shi, Y., Taib, R., Wu, M.A.: Study of Manual Gesture-Based Selection for the PEMMI Multimodal Transport Management Interface. In: Proceedings of ICMI'05, October 4–6, Trento, Italy, pp. 274–281 (2005)
3. Holzapfel, H., Nickel, K., Stiefelhagen, R.: Implementation and Evaluation of a Constraint-Based Multimodal Fusion System for Speech and 3D Pointing Gestures. In: Proceedings of ICMI'04, October 13-15, State College Pennsylvania, USA, pp. 175–182 (2004)
4. Johnston, M.: Unification-based Multimodal Parsing. In: Proceedings of ACL'1998, Montreal, Quebec, Canada, pp. 624–630. ACM, New York (1998)
5. Johnston, M., Bangalore, S.: Finite-state multimodal parsing and understanding. In: Proceedings of COLING 2000, Saarbrücken, Germany, pp. 369–375 (2000)
6. Kaiser, E., Demirdjian, D., Gruenstein, A., Li, X., Niekrasz, J., Wesson, M., Kumar, S., Demo.: A Multimodal Learning Interface for Sketch, Speak and Point Creation of a Schedule Chart. In: Proceedings of ICMI'04, October 13-15, State College Pennsylvania, USA, pp. 329-330 (2004)
7. Latoschik, M.E.: A User Interface Framework for Multimodal VR Interactions. In: Proc. ICMI 2005 (2005)
8. Sun, Y., Chen, F., Shi, Y., Chung, V.: A Novel Method for Multi-sensory Data Fusion in Multimodal Human Computer Interaction. In: Proc. OZCHI 2006 (2006)