Software architectures for incremental understanding of human speech

Gregory Aist[*,1], *James Allen*[1,3], *Ellen Campana*[1,2], *Lucian Galescu*[3],
Carlos A. Gomez Gallo[1], *Scott Stoness*[1], *Mary Swift*[1], *Michael Tanenhaus*[2]

[1]Computer Science Department, University of Rochester, USA
[2]Brain and Cognitive Sciences Department, University of Rochester, USA
[3]Institute for Human and Machine Cognition, Pensacola, Florida, USA
*Contact address: gsa@gregoryaist.com

Abstract

The prevalent state of the art in spoken language understanding by spoken dialog systems is both modular and pipelined. It is modular in the sense that incoming utterances are processed by independent modules that handle different aspects of the signal, such as acoustics, syntax, semantics, and intention / goal recognition. It is pipelined in the sense that each module completes its work for an entire utterance prior to handing off the utterance to the next module. However, a growing body of evidence from the human language understanding literature suggests that humans do not process language in a modular, pipelined way. Rather, they process speech by rapidly integrating constraints from multiple sources of knowledge and multiple linguistic levels incrementally, as the utterance unfolds. In this paper we describe ongoing work aimed at developing an architecture that will allow machines to understand spoken language in a similar way. This revolutionary approach is promising for two reasons: 1) It more accurately reflects contemporary models of human language understanding, and 2) it results in technical improvements including increased parsing performance.

1. Introduction

Computational Natural Language Understanding is an interesting area to study because, despite decades of research, it is one of the many areas of Artificial Intelligence that remains difficult for computers, yet easy for people. The major reason that language understanding is so difficult for computers to understand is that ambiguity is rampant; each input is locally consistent with multiple interpretations, and each of those interpretations, in turn, is locally consistent with a number of potential inputs. This ambiguity occurs simultaneously at all levels of processing. For instance, the speech signal itself is locally consistent with multiple word sequences and each of these word sequences is locally consistent with multiple possible speech inputs. Likewise, each word sequence is locally consistent with multiple syntactic structures, each of which is locally consistent with other possible word sequences. In order to manage this tremendous complexity in real-time, it is necessary to make some simplifying assumptions about how the large problem of Natural Language Understanding can be broken down into smaller, more tractable, sub-problems, and about how each of those sub-problems might be solved. Spoken Dialogue System researchers, specifically, have tended to make the following two simplifying assumptions:

Standard Simplifying Assumption 1: Speech and linguistic information can be treated as independent of other inputs and knowledge sources.
Standard Simplifying Assumption 2: Speech / Language processing can be divided into a small number of levels, each of which depends only on the final utterance-level output of the previous level.

These simplifying assumptions were once consistent with linguistic and psycholinguistic models of how humans understand language. However, a growing body of evidence suggests that humans process spoken language incrementally. New models been developed to account for this data, and the common simplifying assumptions outlined above are not consistent with these models. In the next section we give a brief overview of the data and models of human language processing.

1.1. Incremental Human Language Understanding

In recent years, psycholinguists have begun to use more fine-grained tools and metrics to investigate language. This change has made it possible for researchers to investigate spoken language in more or less natural contexts (Tanenhaus et al., 1995). This body of research has demonstrated that as an utterance unfolds, listeners take advantage of both linguistic and extra-linguistic information to arrive at interpretations more quickly than they could with language alone. For instance, listeners have been shown to use visual information about the scene (Tanenhaus et al., 1995), the goals and perspectives of their partners (Hanna & Tanenhaus, 2003), and spatial / embodied constraints about how objects in the world can be manipulated (Chambers et al., 2004.) during language understanding to restrict the set of potential interpretations that are explored. Similarly, information from different levels of processing such as phonology, lexicon, syntax, semantics and discourse / reference can be combined by listeners to constrain the set of potential interpretations that are explored (Altmann & Kamide, 1999; Tanenhaus et al., 1995).

1.2. Incremental Computer Language Understanding

In the previous section we described the current understanding of how humans process spoken language – incrementally, rapidly integrating information from multiple sources and multiple levels to arrive at partial / local interpretations. Our goal is to develop an architecture that will allow machines to process spoken language in a similar way. However, to the

extent possible we would like to leverage existing technologies and modules. Thus, we propose replacing the common simplifying assumptions described previously with the following ones, which are more consistent with incremental models of human language understanding:

- **Proposed Simplifying Assumption 1:** Speech and linguistic information can be treated as independent of other inputs and knowledge sources with one exception -- dynamically updated non-linguistic knowledge and information can be used to improve search during speech / linguistic processing.
- **Proposed Simplifying Assumption 2:** Speech / Language processing can be divided into a small number of levels, that operate on partial information in parallel. The levels can be treated as independent of one another with one exception – dynamically updated outputs of other levels can be used to improve search within a given level.

We have implemented a system that is based on the TRIPS architecture (Allen et al. 2001), which has been modified to make use of the proposed simplifying assumptions. In the remainder of this paper we describe the architecture in detail, and demonstrate that the new incremental architecture provides technical advantages in addition to the theoretical advances discussed above.

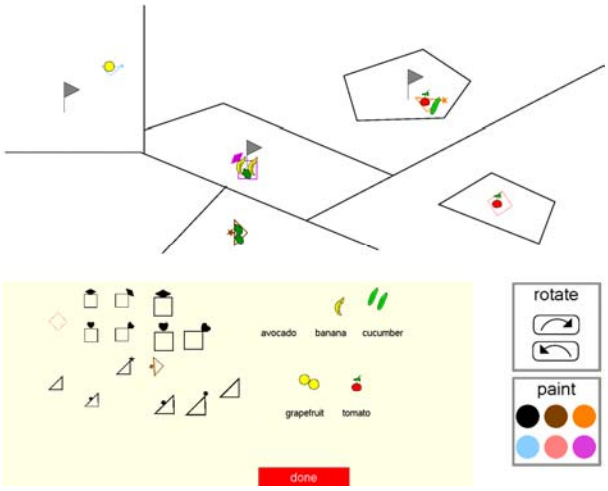

Figure 1 Fruit carts domain – example screen.

2. Human-Human Conversation in a Testbed for Incremental Understanding

In this section we describe the human-human conversational data that we collected which led us to our redesign for incremental understanding.

For the work described in this paper, we utilized the Fruit Carts domain, a testbed we've developed in order to explore issues of incremental understanding. The domain itself is described in detail elsewhere; here we summarize key aspects. Subjects are given a map showing a number of shapes placed on the map, with varying colors, locations, angles, and contents. Their task is to describe how to replicate this map, giving instructions either to another person (for human-human dialog) or to a computer (for human-computer dialog). The main screen is shown in Figure 1. (The subjects have access to a "key" which has names for the regions.) Possible actions include selecting a shape, moving it to a region, painting it, turning it, and filling it with other objects (the fruit).

We used human-human conversations collected in this domain to form the basis for formalizing various aspects of incremental understanding, and for gauging the behavior of the spoken dialog system that we built to operate in this domain.

3. Incremental understanding system, with interleaved results

We now describe how the TRIPS architecture, which is a state-of-the-art spoken dialog system that has been used in a variety of experimental domains including emergency management, equipment purchasing, and learning-from-instruction, has been redesigned to accommodate incremental understanding. In this section, we discuss each major component of the system in turn: speech recognition, segmenter, parser, VP advisor, input manager (IM), behavioral agent (BA), simulator, sequencer, GUI, and eyetracker. Along the way we describe technical improvements where relevant, throughout.

3.1. Speech recognition and segmentation

We used the human-human conversations described above to construct a specialized statistical language model for this domain using the techniques described by Galescu, Ringger, and Allen (1998). This language model was used as one of the inputs to an automatic speech recognizer from the Sphinx family: Sphinx 2, Sphinx 3, or Sphinx 4 depending on the system configuration (Lamere et al. 2003, Mosur et al. n.d., Huang et al. 1993, Lee 1989). The results from the speech recognizer are fed to the parser and to a separate segmentation module (the Segmenter) which uses a small top-down fragment grammar to incrementally make predictions about the presence of interaction-relevant fragments such as verb phrase prefixes ("we need to move") and referring expressions ("a large triangle"). The Segmenter passes its advice on to the Parser and also (for referring expressions) to the GUI, in order to allow the highlighting of possible referents on the screen.

3.2. Parser

From the output of the speech recognizer, the parser produces semantic representations with enough detail for analysis by the system reasoners. The grammar and lexicon used in the system are part of the TRIPS generic language processing front-end, which is used across multiple domains. Porting these components to a new domain requires adding lexical items and increasing grammatical coverage as needed. Lexicon and grammar development for the incremental understanding domain is driven by transcripts from recorded experimental sessions involving people interacting with the system. Domain specific interpretations for the parser output are obtained via a set of transformation rules (Dzikovska, Allen & Swift, 2003) which we constructed for this domain. These transformation rules are also used to boost in-domain word senses so that they will be tried first during parsing.

To evaluate parser performance in incremental understanding mode compared to standard utterance by utterance interpretation we developed a gold standard corpus of

parsed output for a sample dialogue. For each utterance the gold standard includes complete and correct syntactic analysis, word sense disambiguation, semantic role assignment and surface speech act analysis, as well as timing results and number of constituents produced during the parse. The high level of ambiguity in this domain often presents the parser with multiple possible interpretations, and the correct one is not always the first choice of the parser in standard mode.

We have also developed a parsed corpus based on transcripts from experimental sessions to use as training data for new system components such as the VP advisor.

3.3. Interpretation Manager

The Interpretation Manager (IM) takes syntactic analysis from the parser and constructs semantic interpretations. The IM also mediates between the Parser and advice agents such as the VP Advisor and the Simulator/KB, as described below.

3.4. VP Advisor

Even though the fruitcarts experiment allows users to use free style language, the set of actions that can be performed on objects provide us with a well defined constructions we can exploit. In examining the data collected we can summarize the following library of actions with all of their possible thematic roles expressed at one time or another.

Table 1. *Actions and their prototypical arguments*.

Action	arguments
Move	Verb-object-distance-heading-location
Rotate	Verb-object-angle-heading
Select	Verb-object
Paint	Verb-object-color

The action library lists all verb arguments that were seen on corpus. However, due to common elliptical constructions in speech dialogue (Fernandez, Ginzburg, and Lappin 2004), examples of all cases where there was a missing verb or any verb argument were seen. Nevertheless, certain constructions were more likely than others, knowledge, which might help the parser arrive at a more accurate analysis with less effort.

To this end an initial set of six dialogues were manually annotated with verb and verb argument type labels. Then statistics that measured how often a verb argument appears given the verb were collected. Table 2 is an example for the statistics found for the action MOVE.

Table 2. *Statistics for MOVE action*.

args	Probs
-ver-obj-loc	0.658
-ver-obj-hea	0.109
-ver-obj	0.061
-ver-obj-dis	0.049
-ver-loc	0.037
-ver	0.037
-ver-obj-dis-hea	0.036

For example the most likely MOVE action is performed by giving the verb, object and location which is intuitively correct. However this only occurs 66% of the time; MOVE actions are also done by stating a location only. The object is presumably already in the context by a previous SELECT action. This is the case of object elision.

The mechanism works as follows: when the parser is constructing a VP, it asks the VP advisor how likely the construction under consideration is in this domain. This advice is taking place after the logical form of the utterance has been translated into our domain specific semantics. Therefore we can think of the advice as a way to encode semantic restrictions for each verb. The parser then modifies the probability of the constituent in the chart and puts it back into the agenda.

Experimental results show us that on average the number of constituents built by the parser decreases with the VP advice. The best result can be seen on sentences as complicated as the following: “take the box in morningside and put it into pine tree mountain on the bottom of the flag”; here, the number of constituents were decreased by as much as 19%. On less complex sentences such as "and then change it to brown" there is no difference in number of constituents since the parser already finds a spanning parse efficiently.

3.5. Simulator / Real-World Knowledge Base

One of the additional sources of knowledge that can be brought to bear on the process of incremental understanding is knowledge about what is present in the visual world. To explore the potential of this type of information, we began by parsing the sentence “put the square near the flag in the park” with the standard version of the parser, operating without incremental understanding. Now, for a non-incremental parser this sentence is inherently ambiguous, so the choice of a most likely parse is somewhat arbitrary; in the event, the parser selected “the square” as the direct object of the verb, and during the course of the parse built 197 total constituents. (Measurements of parsing efficiency are always tricky, but since both versions of the parser use identical grammars, the number of constituents built should serve as a reasonable measure for our purposes.) Then we created a simple knowledge base, *KB-selected*, which features a selected square, and a flag in a park, but no square near a flag. This set of knowledge clearly favors the interpretation selected by the non-incremental parser above. The incremental parser output the desired interpretation as its most likely parse, but only built 121 constituents; an efficiency improvement of almost 40%.

Operating in incremental mode doesn’t just improve the efficiency of the parser, but its accuracy as well. A different initial knowledge base, *KB-near*, features a square near a flag, but no flag in a park, and has no square selected. This KB favors an interpretation in which “the square near the flag” is the direct object. The non-incremental parser cannot make this distinction, even in principle, and so to capture the multiple possible interpretations, each preferable in a different context, it is necessary for the parser to feed forward a number of complete parses at the completion of its processing.

A incremental understanding parser, however, has at its disposal, incrementally and immediately, the same knowledge that would be used to disambiguate the complete parses in a non-incremental system. Purely by changing the knowledge base to *KB-near* and allowing the reference feedback to be incorporated into the parse, the incremental system finds the

correct parse as its most likely candidate, while building only 131 constituents. *KB-park* is a third knowledge base which has neither a selected square nor a square near a flag, but does feature a square that is near a flag which is in a park. With this KB, the favored NP is "the square near the flag in the park". However, restrictions on the verb "put" require the parse to have both a direct and indirect object, and the parser thus returns to the same interpretation it favored in the absence of any information from the KB. Interestingly, this entire process requires the construction of only 165 constituents; that is, even when the KB leads the parse somewhat astray, the incorporation of the domain knowledge still improves on the base parser's efficiency of 197 constituents. Finally, we tested the sentence "put the square near the flag in the park in the forest". The non-incremental parser found "in the forest" as the indirect object, building 396 constituents in the process. Using *KB-park*, however, the incremental parser arrived at the same interpretation in only 196 constituents.

As well as the proof-in-principle sentences interpreted in context, we have run the system on the transcript of a complete dialogue from the corpus that we collected for this domain:

1 okay so
2 we're going to put a large triangle with nothing into morningside
3 we're going to make it blue
4 and rotate it to the left forty five degrees
5 take one tomato and put it in the center of that triangle
6 take two avocados and put it in the bottom of that triangle
7 and move that entire set a little bit to the left and down
8 mmkay
9 now take a small square with a heart on the corner
10 put it onto the flag area in central park
11 rotate it a little more than forty five degrees to the left
12 now make it brown
13 and put a tomato in the center of it
14 yeah that's good
15 and we'll take a square with a diamond on the corner
16 small
17 put it in oceanview terrace
18 rotate it to the right forty five degrees
19 make it orange
20 take two grapefruit and put them inside that square
21 now take a triangle with the star in the center
22 small
23 put it in oceanview just to the left of oceanview terrace
24 and rotate it left ninety degrees
25 okay
26 and put two cucumbers in that triangle
27 and make the color of the triangle purple

The experiment proceeded in much the same manner as the proof-in-principle, with candidate NPs being sent forward through the Interpretation Manager to the Knowledge Base, which provided feedback on whether the NP was a reasonable candidate, taking into account both domain-specific knowledge and the current state of the world. Because the user's utterances had to be interpreted relative to the state of the world that the user had been aware of during dialogue collection, a series of knowledge base updates were performed between sentences to ensure that the KB was an accurate reflection of what the user had seen.

The results of the experiment are as follows: Overall, the incremental understanding parser only had to build 75% as many constituents as the standard parser in order to find its first complete parse of each utterance. Stoness et al. (2005) provides further detail for the interested reader.

3.6. Behavioral Agent; Output Components

The Behavioral Agent produces a decision about what to do, based on input from the Interpretation Manager. These decisions are passed on to a string of components (Simulator, Sequencer, and GUI) which in the end results in an action, such as highlighting an object or moving it to a new location.

4. Related Work and Conclusion

Higashinaka et al. (2002) describe work on a process they term Incremental Sentence Sequence Search (ISSS), where both sentences and sentence fragments are used to update the dialog state. ISSS constructs multiple dialog states which can be decided upon as needed after any desired interval of speech. In a sense this can be viewed as a late binding process, whereas our work generally takes an earlier binding approach where information is brought to bear on the search as soon as possible. (In principle either system could no doubt be configured to perform late binding or early binding as desired, depending on configuration desired.)

Rose et al. (2002) describe briefly a reworking of a chart parser to handle incremental typed input, where "as the text is progressively revised, only minimal changes are made to the chart" – their primary finding was that incrementally parsing incoming text allows for the parsing time to be folded into the time it takes to type, which can be substantial especially for longer user responses. Our current work operates on spoken input as well as typed input and makes extensive use of the visual context and of pragmatic constraints in order to help with the parsing process.

The most closely related work to this paper is probably that of DeVault and Stone (2003), where they describe techniques for incremental interpretation that involve annotating edges in a parser's chart with the constraints of various types that must be fulfilled in order to the edge to be valid. This architecture has a clean and appealing simplicity to it, but seems to impose a degree of uniformity on the sort of information and reasoning processes that can be brought to bear on the parsing process. Our approach is more agnostic: advice to the parser is represented as modifications to the chart, and can thus be in any framework appropriate to the source.

In conclusion, we have presented a system architecture for incremental understanding of human speech, during human-computer spoken dialog. In addition, we have demonstrated a number of technical improvements that arise from the incremental understanding process. Incremental understanding is proving to be an exciting and productive area for spoken language research.

5. Acknowledgements

This paper is based on work supported by the National Science Foundation (NSF), the National Institutes of Health (NIH), and the National Aeronautics and Space Administration (NASA). (Grant numbers omitted for review).

6. References

[1] Allen, J., Byron, D., Dzikovska, M., Ferguson, G., Galescu, L. and Stent, A. 2001. "Towards Conversational Human-Computer Interaction," *AI Magazine* **22**(4), pages 27-38.

[2] Altmann, G.T.M., and Kamide, Y. 1999. "Incremental interpretation at verbs: resticting the domain of of subsequent reference." Cognition, 73, pp. 247-264.

[3] Chambers, C.G., Tanenhaus, M.K., & Magnuson, J.S., 2004. "Actions and affordances in syntactic ambiguity resolution." Journal of Experimental Psychology: Learning, Memory & Cognition, 30, pages 687-696.

[4] Clark, H., and Wilkes-Gibbs, D. 1990. "Referring as a collaborative process." In Cohen, P., Morgan, J. and Pollack, M. E., eds. Intentions in Communication, MIT Press. Pages 463-493.

[5] DeVault, D., and Stone, M. Domain inference in incremental interpretation. ICOS 2003.

[6] Dzikovska, M.O., Allen, J.F., Swift, M.D. Integrating linguistic and domain knowledge for spoken dialogue systems in multiple domains. In Proceedings of Workshop on Knowledge and Reasoning in Practical Dialogue Systems at The Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-2003), pp. 25-35. Acapulco, Mexico, August 2003.

[7] Fernandez, R., Ginzburg, J., and Lappin, S. Classifying Ellipsis in Dialogue: A Machine Learning Approach. Proceedings of the 20th International Conference on Computational Linguistics. 2004.

[8] Galescu, L., Ringger, E., and Allen, J. "Rapid Language Model Development for New Task Domains," in Proceedings of the First International Conference on Language Resources and Evaluation (LREC), Granada, Spain. May 1998.

[9] Hanna, J. E., and Tanenhaus, M.K. "The effects of common ground and perspective on domains of referential interpretation." Journal of Memory and Language, 49, pages 43-61. 2003.

[10] Higashinaka, R., Miyazaki, N., Nakano, M., and Aikawa, K. A method for evaluating incremental utterance understanding in spoken dialogue systems. ICSLP 2002. www.kecl.ntt.co.jp/icl/kpro/rh/pdf/ICSLP2002Poster.pdf

[11] Huang, X.D., Alleva, F.; Hon, H.-W.; Hwang, M.-Y.; Lee, K.-F., and Rosenfeld, R. SPHINX-II speech recognition system: An overview. Computer Speech & Language. Vol. 7, no. 2, pp. 137-148. 1993.

[12] Lamere, P., Kwok, P., Gouvêa, E., Bhiksha Raj, Singh, R., Walker, W., Warmuth, M., and Wolf, P. The CMU Sphinx-4 Speech Recognition System. ICASSP 2003.

[13] Lee, K.-F., Automatic Speech Recognition: The Development of the SPHINX SYSTEM, Kluwer Academic Publishers, Boston, 1989.

[14] Mosur, R., and the Sphinx Speech Group. n.d. Sphinx-3 s3.X Decoder (X=5). Online documentation at SourceForge: http://cmusphinx.sourceforge.net/sphinx3/

[15] Rose, C.P., Roque, A., Bhembe, D., and Van Lehn, K. An efficient incremental architecture for robust interpretation. Human Language Technology Conference, 2002.

[16] Stoness, S.C., Allen, J., Aist, G., and Swift, M. Using real-world reference to improve spoken language understanding. AAAI Workshop on Spoken Language Understanding. 2005.

[17] Tanenhaus, M.K., Spivey-Knowlton, M.J., Eberhard, K.M., and Sedivy, J.C. 1995. Integration of visual and linguistic information in spoken language comprehension. Science, Vol. 268 (5217), 1632-1634.

[18] Tanenhaus, M.K., Magnuson, J.S., Dahan, D., and Chambers, C. 2000. Eye movements and lexical access in spoken-language comprehension: Evaluating a linking hypothesis between fixations and linguistic processing. Journal of Psycholinguistic Research, Vol. 29 (6), 557-580.