# Incremental Parsing with Reference Interaction

**Scott C. Stoness, Joel Tetreault, James Allen**
Department of Computer Science
University of Rochester
Rochester, NY, USA
stoness@cs.rochester.edu
tetreaul@cs.rochester.edu, james@cs.rochester.edu

## Abstract

We present a general architecture for incremental interaction between modules in a speech-to-intention continuous understanding dialogue system. This architecture is then instantiated in the form of an incremental parser which receives suitability feedback on NP constituents from a reference resolution module. Oracle results indicate that perfect NP suitability judgments can provide a labelled-bracket error reduction of as much as 42% and an efficiency improvement of 30%. Preliminary experiments in which the parser incorporates feedback judgments based on the set of referents found in the discourse context achieve a maximum error reduction of 9.3% and efficiency gain of 4.6%. The parser is also able to incrementally instantiate the semantics of underspecified pronouns based on matches from the discourse context. These results suggest that the architecture holds promise as a platform for incremental parsing supporting continuous understanding.

## 1 Introduction

Humans process language incrementally, as has been shown by classic psycholinguistic discussions surrounding the garden-path phenomenon and parsing preferences (Altmann and Steedman, 1988; Konieczny, 1996; Phillips, 1996). Moreover, a variety of eye-tracking experiments (Cooper, 1974; Tanenhaus and Spivey, 1996; Allopenna et al., 1998; Sedivy et al., 1999) suggest that complex semantic and referential constraints are incorporated on an incremental basis in human parsing decisions.

Computational parsers, however, still tend to operate an entire sentence at a time, despite the advent of speech-to-intention dialogue systems such as Verbmobil (Kasper et al., 1996; Noth et al., 2000; Pinkal et al., 2000), Gemini (Dowding et al., 1993; Dowding et al., 1994; Moore et al., 1995) and TRIPS (Allen et al., 1996; Ferguson et al., 1996; Ferguson and Allen, 1998). Naturalness, robustness, and interactivity are goals of such systems, but control flow is typically the sequential execution of modules, each operating on the output of its predecessor; only after the entire sentence has been parsed do higher-level modules such as intention recognition and reference resolution get involved.

In contrast to this sequential model is the *continuous understanding* approach, in which all levels of language analysis occur simultaneously, from speech recognition to intention recognition. As well as being psycholinguistically motivated, continuous understanding models offer potential computational advantages, including accuracy and efficiency improvements for real-time spoken language understanding and better support for the spontaneities of natural human speech. Continuous understanding is necessary if the system is to respond before the entire utterance is analyzed, a prerequisite for incremental confirmation and clarification. The major computational advantage of continuous understanding models is that high-level expectations and feedback should be able to influence the search of lower-level processes, thus leading to a focused search through hypotheses that are plausible at all levels of processing.

One of the major current applications of parsers that operate incrementally is for language modelling in speech recognition (Brill et al., 1998; Jelinek and Chelba, 1999). This work is important not only for its ability to improve performance on the speech recognition task; it also models the interactions between speech recognition and parsing in a continuous understanding system. Our research attempts to further the quest for continuous understanding by moving one step up the hierarchy, building an incremental parser which is the advisee rather than the advisor.

We begin by presenting a general architecture for incremental interaction between the parser and higher-level modules, and then discuss a specific instantiation of this general architecture in which a reference resolution module provides feedback to the parser on the suitability of noun phrases. Experiments with incremental feedback from a refer-

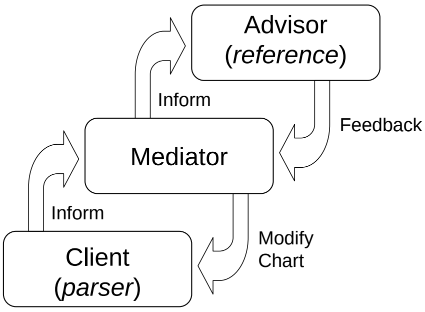


Figure 1: A General Architecture for Incremental Parsing

ence resolution module and an NP suitability oracle are reported, and the ability of the implementation to incrementally instantiate semantically underspecified pronouns is outlined. We believe this research provides an important start towards developing end-to-end continuous understanding models.

## 2 An Incremental Parsing Architecture

Many current parsers fall into the class of history-based grammars (Black et al., 1992). The independence assumptions of these models make the parsing problem both stochastically and computationally tractable, but represent a simplification and may therefore be a source of error. In a continuous understanding framework, higher-level modules may have additional information that suggests loci for improvement, recognizing either invalid independence assumptions or errors in the underlying probability model.

We have designed a general incremental parsing architecture (Figure 1) in which the Client, a dynamic programming parser, performs its calculations, the results of which are incrementally passed on via a Mediator to an Advisor with access to higher-level information. This higher-level Advisor sends feedback to the Mediator which has access to the Client's chart, and which then surreptitiously changes and/or adds to the chart in order to make the judgments conform more closely to those of the Advisor. The parser, whose chart has (unbeknownst to it) been changed, then simply calculates chart expansions for the next word, naïvely expanding the currently available (and possibly modified) hypotheses.

This architecture is general in that neither the Mediator nor the Advisor have been specified; either of these modules can be instantiated in any number of ways within the general framework. The typical dynamic programming component will function in very much the same way that it does in the vanilla algorithm, except that the chart in which partial results are recorded may be modified between time steps. The Client can be any system which uses dynamic programming to efficiently encode independence assumptions, so long as it provides the Mediator with the ability to modify chart probabilities and add chart entries; otherwise the original parser can remain untouched. By having the Mediator perform these modifications rather than the Advisor, we preserve modularity: in this architecture the Advisor need not be aware of the specific implementation of the Client, although depending on the type of advice provided, it may need access to the underlying grammar. The Mediator isolates the Advisor and Client from each other as well as determining how the feedback will be introduced into into the Client's chart.

Stoness (2004) identifies two broad categories of *subversion* - our term for the Mediator's surreptitious modification of the Client's chart - as outlined below:

- **Heuristic Subversion:** the Mediator uses the Advisor's feedback as heuristic information, affecting the search sequence but not the probabilities calculated for a given hypothesis; and
- **Chart Subversion:** the Mediator is free to modify the Client's chart as necessary, but does not directly affect the search sequence of the Client (except insofar as this is accomplished by the modifications to the chart).

The two types of subversion have very different properties. Heuristic subversion will affect the set of analyses which is output by the parser, but each of those analyses will have exactly the same probability score as under the original parser; the effects of the Advisor are essentially limited to determining which hypotheses remain within the beam, or the order in which hypotheses are expanded, depending on whether the underlying parser uses a beam search or an agenda. Chart subversion, on the other hand, will actually change the scores assigned analyses, resulting in a new probability distribution. Heuristic subversion is considerably less powerful, but more stable; the effects of chart subversion can be fairly chaotic, especially if care is not taken to avoid feedback loops. Stoness (2004) outlines conditions under which the effects of chart subversion are predictable, becoming broadly equivalent to an incremental version of a post-hoc re-ranking of the Client's output hypotheses.

Further details on the general architecture, including properties of various modes of feedback integration, a discussion of the relationship between incremental parsing and parse re-ranking, the possibilities of multiple Advisors working in combination, and provisions in the model for asynchronous feedback are available in a University of Rochester Technical Report (Stoness, 2004).

## 3 Instantiating the Architecture

Working in the context of TRIPS, an existing task-oriented dialogue system, we have modified the existing parser and reference resolution modules so that they communicate incrementally with each other. This models the early incorporation of reference resolution information seen in humans (Chambers et al., 1999; Allopenna et al., 1998), and allows reference resolution information to affect parsing decisions.

For example, in "Put the apple in the box in the corner" there is an attachment ambiguity. Reference resolution can determine the number of matches for the noun phrase "the apple" incrementally; if there is a single match, the parser would expect this to be a complete NP, and prefer the reading where the box is in the corner. If reference returns multiple matches for "the apple", the parser would expect disambiguating information, and prefer a reading where additional information about the apple is provided: in this case, an the NP "the apple in the box".

With solid feedback from reference, it should be possible to remove some of the ambiguity inherent in the search process within the parser. This will simultaneously guide the search to the most likely region of the search space, improving accuracy, and delay the search of unlikely regions, improving efficiency. Of course, this comes at the cost of some communication overhead and additional reference resolution. Ideally, the overall improvement in the parser's search space would be enough to cover the additional incremental operation costs of other modules.

### 3.1 An Incremental Parser

The pre-existing parser in the dialogue system was a pure bottom-up chart parser with a hand-built grammar suited for parsing task-oriented dialogue. The grammar consisted of a context-free backbone with a set of associated features and semantic restrictions, including agreement, hard subcategorization constraints, and soft selectional restriction preferences. The parser has been modified so that whenever a constituent is built, it can be sent forward to the Mediator, allowing for the possibility of feedback. The architecture and experiments described in this paper were performed in a synchronous mode, but the parser can also operate in an incrementally asynchronous mode, where it continues to build the chart in parallel with other modules' operations; probability adjustments to the chart then cascade to dependent constituents.

### 3.2 Interaction with Reference

When the parser builds a potential referring expression (e.g. any NP), it is immediately passed on to the Advisor, the reference resolution module described in Tetreault et. al. (2004) modified for incremental interaction. This module then determines all possible discourse referents, providing the parser with a ranked classification based on the salience of the referents and the (incremental) syntactic environment.

The reference module keeps a dynamically updated list of currently salient discourse entities against which incoming incrementally constructed NP constituents are matched. Before any utterances are processed, the module loads a static database of *relevant* place names in the domain; all other possible referents are discourse entities which have been spoken of during the course of the dialogue. For efficiency, the dynamic portion of the context list is limited to the ten most recent contentful utterances; human-annotated antecedent data for this corpus shows that 99% of all pronoun antecendents fall within this threshold. After each sentence is fully parsed the context list is updated with new discourse entities introduced in the utterance; ideally, these context updates would also be incremental, but this feature was omitted in the current version for simplicity.

The matching process is based on that described by Byron (2000), and differs from that of many other reference modules in that every entity and NP-constituent has a (possibly underspecified) semantic feature vector, and it is both the logical and semantic forms which determine successful matchings. Adding semantic information increases the accuracy of the reference resolution from 44% to 58% (Tetreault and Allen, 2004), and consequently improves the feedback provided to the parser.

The Mediator receives the set of all possible referents, including the semantic content of the referent and a classification of whether the referent is the single salient entity in *focus*, has previously been *mentioned*, or is a *relevant* place name.

### 3.3 Mediator

The Mediator interprets the information received from reference and determines how the parser's chart should be modified. If the NP matches nothing in the discourse context, *no match* is returned; otherwise each referent is annotated with its type and discourse distance, and this set is run through a classifier to reduce it to a single tag. The resulting tag is the reference resolution tag, or $R$. The NP constituents are also classified by definiteness and number, giving an NP tag $N$.

For each classifier, we trained a probability model which calculated $P_r$, the probability that a noun phrase constituent $c$ would be in the final parse, conditioned on $R$ and $N$, or

$$P_r = p(c \text{ in final parse}|R, N).$$

This probability was then linearly combined with the parser's constituent probability,

$$P_p = p(c \to w_m^n),$$

according to the equation

$$P(c) = (1 - \lambda) \cdot P_p + \lambda \cdot P_r$$

for various values of $\lambda$. Evaluation using held-out data suggested that a value of $\lambda = 0.2$ would be optimal. This style of feedback is an example of chart subversion, as it is a direct modification of constituent probabilities by the Mediator, defining a new probability distribution.

## 4 Experiments

The Monroe domain (Tetreault et al., 2004; Stent, 2001) is a series of task-oriented dialogues between human participants set in a simulated rescue operation domain, where participants collaboratively plan responses to emergency calls. Dialogues were recorded, broken up into utterances, and then transcribed by hand, removing speech repairs from the parser input. These transcriptions served as input for all experiments reported below.

A probabilistic grammar was trained from supervised data, assigning PCFG probabilities for the rule expansions in the CFG backbone of the hand-crafted, semantically constrained grammar. The parser was run using this grammar, but without any incremental interaction whatsoever, in order to establish baseline accuracy and efficiency numbers. The corpus consists of six task-oriented dialogues; four were used for the PCFG training, one was held out to establish appropriate parameter values, and one was selected for testing. The held-out and test dialogues contain hand-checked gold standard parses.

Under normal operation of the sequential dialogue system, the parser is run in best-first mode, providing only a single analysis to higher-level modules, and has a constituent construction limit in an attempt to simulate the demands of a real-time system. When the parser reaches the constituent limit, appropriate partial analyses are collected and forwarded to higher-level modules. These constraints were kept in place during our experiments, because they would be necessary under normal operation of the system. Thus, the inability to parse a sentence does not necessarily indicate a lack of coverage of the grammar, but rather a lack of efficiency in the parsing process.

|  | Base | All NPs | Def-Sing |
|---|---|---|---|
| Precision | 94.6 | 97.2 | 96.3 |
| Recall | 71.1 | 83.1 | 78.8 |
| F-statistic | 82.9 | 90.2 | 87.6 |
| Improvement | N/A | 7.3 | 4.7 |
| Error Red. | N/A | 42.4 | 27.2 |
| Work Red. | N/A | 30.3 | 18.7 |
| Perfect S | 224 | 241 | 236 |
| Parsed S | 270 | 282 | 279 |

Table 1: Results for (a) The baseline parser without reference feedback, (b) An Oracle Advisor correctly determining status of all NPs, (c) An Oracle Advisor correctly determining status of definite singular NPs.

As can be seen in Table 1, the parser achieves a 94.6% labelled bracket precision, and a 71.1% labelled bracket recall. Note that only constituents of complete parses were checked against the gold standard, to avoid any bias introduced by the partial parse evaluation metric. Of the 290 gold standard utterances in the test data, 270 could be parsed, and 224 were parsed perfectly.

### 4.1 Oracle Evaluation

We began with a feasibility study to determine how significant the effects of incremental advice on noun phrases could be in principle. The feedback from the reference module is designed to determine whether particular NPs are good or bad from a reference standpoint. We constructed a simple feedback oracle from supervised data which determined, for each NP, whether or not the final parse of the sentence contained an NP constituent which spanned the same input. Those NPs marked "good", which did appear in the parse, were added to the chart as new constituents. NPs marked "bad" were added to the chart with a probability of zero[1]. A second or-

[1] In some sense, this style of feedback is an example of heuristic subversion, as it has the effect of keeping "good" analyses around while removing "bad" analyses from the search space. Technically, this is also chart subversion, as each hypothesis has its score multiplied by 1 or 0, depending on

acle evaluation performed this same task, but only providing feedback on definite singular NPs.

The results of both oracles are shown in Table 1. The first five rows give the precision, recall, f-statistic, the raw f-statistic improvement, and the f-statistic error reduction percentage, all determined in terms of labelled bracket accuracy. There is a marked increase in both precision and recall, with an overall error reduction of 42.4% with the full oracle and 27.2% with the definite singular oracle. Thus, in this domain over a quarter of all incorrectly labelled constituents are attributable to syntactically incorrect definite singular NPs. The number of constituents built during the parse is used as a measure of efficiency, and the work reduction is reported in the sixth row of the table, showing an efficiency improvement of 30.3% or 18.7%, depending on the oracle. The final two lines of the table show that both the number of sentences which can be parsed and the number of sentences which are perfectly parsed increase under both models.

The nature of the oracle experiment ensures some reduction in error and complexity, but the magnitude of the improvement is surprising, and certainly encouraging for the prospects of incremental reference. Definite singular NPs typically have a unique referent, providing a locus for effective feedback, and we believe that incremental interaction with an accurate reference module might approach the oracle performance.

### 4.2 Dialogue Experiments

For these experiments the parser interacted with the actual reference module, incorporating feedback according to the model discussed in Section 3.3. The first data column of Table 2 repeats the baseline results of the parser without reference feedback. The next two columns show statistics for a run of the parser with incremental feedback from reference, using a probability model based on a classification scheme which distinguished only whether or not the set of referent matches was empty. The second data column shows the results for the estimated interpolation parameter value of $\lambda = 0.2$, while the third data column shows results for the empirically determined optimal $\lambda$ value of 0.1.

The results are encouraging, with an error reduction of 8.2% or 9.3% on the test dialogue, although the amount of work the parser performed was reduced by only 4.0% and 3.6%. A further encouraging sign is that for every exploratory $\lambda$ value we tried in either the held-out or the test data, both the accuracy and efficiency improved. Reference information also helped increase both the number of sentences that could be parsed and the number of sentences that were parsed perfectly, although the improvements were small.

| | Base | SC | SC | CC |
|---|---|---|---|---|
| $\lambda =$ | N/A | 0.2 | 0.1 | 0.2 |
| Precision | 94.6 | 94.5 | 94.8 | 93.9 |
| Recall | 71.1 | 74.1 | 74.2 | 73.9 |
| F-statistic | 82.9 | 84.3 | 84.5 | 83.9 |
| F-stat Imp. | N/A | 1.4 | 1.6 | 1.0 |
| Error Red. | N/A | 8.2 | 9.3 | 5.8 |
| Work Red. | N/A | 3.6 | 4.0 | 4.6 |
| Perfect S | 224 | 225 | 228 | 223 |
| Parsed S | 270 | 273 | 273 | 273 |

Table 2: Results for Discourse Experiment with Simple (SC) and Complex (CC) Classifiers

The estimated value of $\lambda = 0.2$ produced an error reduction that was approximately 20% of the oracular, which is a very good start, especially considering that this experiment used only the information of whether there was a referent match or not. The efficiency gains were more modest at just above 10% of the oracular results, although one would expect less radical efficiency improvements from this experiment, since under the linear interpolation of the experiment, even extremely dispreferred analyses may be expanded, whereas the oracle simply drops all dispreferred NPs off the beam immediately.

We performed a second experiment that made more complete use of the reference data, breaking down referent sets according to when and how often they were mentioned, whether they matched the focus, and whether they were in the set of relevant place names. We expected that this information would provide considerably better results than the simple match/no-match classification above. For example, consider a definite singular NP: if it matches a single referent, one would expect it to be in the parse with high probability, but multiple matches would indicate that the referent was not unique, and that the base noun probably requires additional discriminating information (e.g. a prepositional phrase or restrictive relative clause).

Unfortunately, as the final column of Table 2 shows, the additional information did not provide much of an advantage. The amount of work done was reduced by 4.6%, the largest of any efficiency improvement, but error reduction was only 5.8%, and the number of sentences parsed perfectly actu-

whether it is “good” or “bad”. In this degenerate case of all-or-nothing feedback, chart subversion and heuristic subversion are equivalent.

ally decreased by one.

We conjecture that co-reference chains may be a significant source of confusion in the reference data. Ideally, if several entities in the discourse context all refer to the same real-world entity, they should be counted as a single match. The current reference module does construct co-referential chains, but a single error in co-reference identification will cause all future NPs to match both the chain and the misidentified item, instead of producing the single match desired.

The reference module has to rely on the parser to provide the correct context, so there is something of a bootstrapping problem at work, which indicates both a drawback and a potential of this type of incremental interaction. The positive feedback loop bodes well for the potential benefits of the incremental system, because as the incremental reference information begins to improve the performance of the parser, the context provided to the reference resolution module improves, which provides even more accurate reference information. Of course, in the early stages of such a system, this works against us; many of the reference resolution errors could be a result of the poor quality of the discourse context.

Our current efforts aim to identify and correct these and other reference resolution issues. Not only will this improve the performance of the Reference Advisor from an incremental parsing standpoint, but it should also further our understanding of reference resolution itself.

We have shown efficiency improvements in terms of the overall number of constituents constructed by the parser; however, one might ask whether this improvement in parsing speed comes at a large cost to the overall efficiency of the system. We suggest that this is in some sense the wrong question to ask, because for a real-time interactive system the primary concern is to keep up with the human interlocutor, and the incremental approach offers a far greater opportunity for parallelism between modules. In terms of time elapsed from speech to analysis, the system as a whole should benefit from the incremental architecture.

## 5 Semantic Replacement

When the word "it" is parsed as a referential NP, it is given highly underspecified semantics. We have implemented a Mediator which, for each possible referent for "it", adds a new item to the parser's chart with the underspecified semantics of "it" instantiated to the semantics of the referent.

Consider the sentence sequence "Send the bus to the hospital", "Send it to the mall". At the point that the NP "it" is encountered in the second sentence, it has not yet been connected to the verb, so the incremental reference resolution determines that "the bus" and "the hospital" are both possible referents. We add two new constituents to the chart: "it"[the hospital] and "it"[the bus]. They are given probabilities infinitesimally higher than the "it"[underspecified] which already exists on the chart. Thus, if either of the new versions of "it" match the semantic restrictions inherent in the rest of the parse, they will be featured in parses with a higher probability than the underspecified version. "It"[the bus] matches the mobility required of the object of "send", while "it"[the hospital] does not. This results in a parse where the semantics of "it" are instantiated early and incrementally.

This sort of capability is key for an end-to-end incremental system, because neither the reference module nor the parser is capable, by itself, of determining incrementally that the reference in question must be "the bus". If we want an end-to-end system which can interact incrementally with the user, this type of decision-making must be made in an incremental fashion.

This ability is also key in the presence of soft constraints or other Advisors which prefer one possible moveable referent to another; under incremental parsing, these constraints would have the chance to be applied during the parsing process, whereas a sequential system has no alternatives to the default, underspecified pronoun, and so cannot apply these restrictions to discriminate between referents.

Our implementation performs the semantic vetting discussed above, but we have done no large-scale experiments in this area.

## 6 Related Work

There are instances in the literature of incremental parsers that pass forward information to higher-level modules, but none, to our knowledge, are designed as continuous understanding systems, where all levels of language analysis occur (virtually) simultaneously.

For example, there are a number of robust semantic processing systems (Pinkal et al., 2000; Rose, 2000; Worm, 1998; Zechner, 1998) which contain incremental parsers that pass on partial results immediately to the robust semantic analysis component, which begins to work on combining these sentence fragments. If the parser cannot find a parse, then the semantic analysis program has already done at least part of its work. However, none of the above systems have a feedback loop between the semantic analysis component and the incremental parser. So, while all of these are in some sense examples of incremental parsing, they are not continuous understanding models.

Schuler (2002) describes a parser which builds both a syntactic tree and a denotation-based semantic analysis as it parses. The denotations of constituents in the environment are used to inform parsing decisions, much as we use the static database of place names. However, the feedback in our system is richer, based on the context provided by the preceding discourse. Furthermore, as an instantiation of the general architecture presented in Section 2, our system is more easily extensible to other forms of feedback.

## 7 Future Work

There is a catch-22 in that the accurate reference information necessary to improve parsing accuracy is dependent on an accurate discourse context which is reliant on accurate parsing. One way to cut this Gordian Knot is to use supervised data to ensure that the discourse context in the reference module is updated with the gold standard parse of the sentence rather than the parse chosen by the parser; a context oracle, if you will.

A major undertaking necessary to advance this work is an error analysis of the reference module and of the parser's response to feedback; when does feedback lead to additional work or decreased accuracy on the part of the incremental parser, and is the feedback that leads to these errors correct from a reference standpoint?

Currently, the accuracy of the parser is couched in syntactic terms. The precision of the baseline PCFG is fairly high at 94.6%, but that could conceal semantic errors, which could be corrected with reference information. Assessing semantic accuracy is one of a number of alternative evaluation metrics that we are exploring.

We intend to gather timing data and investigate other efficiency metrics to determine to what extent the efficiency gains in the parser offset the communication overhead and the extra work performed by the reference module.

We also plan to do experiments with different feedback regimes, experimenting both with the actual reference results and with the oracle data. Further experiments with this oracle data should enable us to appropriately parameterize the linear interpolation, and indeed, to investigate whether linear interpolation itself is a productive feedback scheme, or whether an integrated probability distribution over parser and reference judgments is more effective. The latter scheme is not only more elegant, but can also be shown to produce probabilities equivalent to those assigned parses in the parse re-ranking task (Stoness, 2004).

We've shown (Stoness, 2004) that feedback which punishes constituents that are not in the final parse cannot result in reduced accuracy or efficiency; under certain restrictions, the same holds of rewarding constituents that will be in the final parse. However, it is not clear how quickly the efficiency and accuracy gains drop off as errors mount. By introducing random mistakes into the Oracle Advisor, we can artificially achieve any desired level of accuracy, which will enable us to explore the characteristics of this curve. The accuracy and efficiency response under error has drastic consequences on the types of Advisors that will be suitable under this architecture.

Finally, it is clear that finding only the discourse context referents of a noun phrase is not sufficient; intuitively, and as shown by Schuler (2002), real-world referents can also aid in the parsing task. We intend to enhance the reference resolution component of the system to identify both discourse and real-world referents.

## 8 Conclusion

These preliminary experiments, using the coarsest grain of reference information possible, achieve a significant fraction of the oracular accuracy improvements, highlighting the potential benefits of incremental interaction between the parser and reference in a continuous understanding system.

The Oracle feedback for NPs shows that it is possible to simultaneously improve both the accuracy and efficiency of an incremental parser, providing a proof-in-principle for the general incremental processing architecture we introduced. This architecture holds great promise as a platform for instantiating the wide range of interactions necessary for true continuous understanding.

## 9 Acknowledgements

Partial support for this project was provided by ONR grant no. N00014-01-1-1015, "Portable Dialog Interfaces" and NSF grant 0328810 "Continuous Understanding".

## References

J. Allen, B. Miller, E. Ringger, and T. Sikorski. 1996. Robust understanding in a dialogue system. In *Proc. of ACL-96*, pages 62–70.

P. D. Allopenna, J. S. Magnuson, and M. K. Tanenhaus. 1998. Tracking the time course of spoken word recognition using eye movements: evidence for continuous mapping models. *Journal of Memory and Language*, 38:419–439.

G. Altmann and M. Steedman. 1988. Interaction with context during human sentence processing. *Cognition*, 30:191–238.

E. Black, F. Jelinkek, J. Lafferty, D. Magerman, R. Mercer, and S. Roukos. 1992. Towards history-based grammars: using richer models for probabilistic parsing. In *Proc. of the Fifth DARPA Speech and Natural Language Workshop*.

E. Brill, R. Florian, J. C. Henderson, and L. Mangu. 1998. Beyond n-grams: Can linguistic sophistication improve language modeling? In *Proc. of COLING-ACL-98*, pages 186–190.

D. K. Byron. 2000. Semantically enhanced pronouns. In *Proc. of DAARC2000: 3rd International Conference on Discourse Anaphora and Anaphor Resolution*.

C. G. Chambers, M. K. Tanenhaus, and J. S. Magnuson. 1999. Real world knowledge modulates referential effects on pp-attachment: Evidence from eye movements in spoken language comprehension. Conference Abstract. Architechtures and Mechanisms for Language Processing.

R. M. Cooper. 1974. The control of eye fixation by the meaning of spoken language. *Cognitive Psychology*, 6:84–107.

J. Dowding, J. M. Gawron, D. Appelt, J. Bear, L. Cherny, R. Moore, and D. Moran. 1993. Gemini: A natural language system for spoken-language understanding. In *Proc. of ACL-93*, pages 54–61.

J. Dowding, R. Moore, F. Andry, and D. Moran. 1994. Interleaving syntax and semantics in an efficient bottom-up parser. In *Proc. of ACL-94*, pages 110–116.

G. Ferguson and J. Allen. 1998. Trips: An integrated intelligent problem-solving assistant. In *Proc. of AAAI-98*, pages 567–572.

G. Ferguson, J. Allen, and B. Miller. 1996. Trains-95: Towards a mixed-initiative planning assistant. In *Proc. of the 3rd International Conference on Artificial Intelligence Planning Systems*, pages 70–77.

Frederick Jelinek and Ciprian Chelba. 1999. Putting language into language modeling. In *Proc. of Eurospeech-99*.

W. Kasper, H.-U. Krieger, J. Spilker, and H. Weber. 1996. From word hypotheses to logical form: An efficient interleaved approach. In *Natural Language Processing and Speech Technology: Results of the 3rd Konvens Conference*, pages 77–88.

Lars Konieczny. 1996. *Human Sentence Processing: A Semantics-Oriented Parsing Approach*. Ph.D. thesis, Universitat Freiburg.

R. Moore, D. Appelt, J. Dowding, J. M. Gawron, and D. Moran. 1995. Combining linguistic and statistical knowledge sources in natural-language processing for atis. In *Proc. ARPA Spoken Language Systems Technology Workshop*.

E. Noth, A. Batliner, A. Kiessling, R. Kompe, and H. Niemann. 2000. Verbmobil: The use of prosody in the linguistic components of a speech understanding system. *IEEE Transactions on Speech and Audio Processing*, 8(5):519–531.

Colin Phillips. 1996. *Order and Structure*. Ph.D. thesis, MIT.

M. Pinkal, C.J. Rupp, and K. Worm. 2000. Robust semantic processing of spoken language. In *Verbmobil: Foundations of Speech-to-Speech Translation*, pages 321–335.

C. P. Rose. 2000. A framework for robust semantic interpretation. In *Proc. of the Sixth Conference on Applied Natural Language Processing*.

W. Schuler. 2002. Interleaved semantic interpretation in environment-based parsing. In *Proc. of COLING-02*.

J. C. Sedivy, M. K. Tanenhaus, C. G. Chambers, and G. N. Carlson. 1999. Achieving incremental semantic interpretation through contextual representation. *Cognition*, 71:109–147.

A. Stent. 2001. *Dialogue Systems as Conversational Partners*. Ph.D. thesis, University of Rochester.

S. C. Stoness. 2004. A general architecture for incremental parsing. Technical report, TR 838, University of Rochester.

M. K. Tanenhaus and M. Spivey. 1996. Eye-tracking. *Language and Cognition Processes*, 11(6):583–588.

J. Tetreault and J. Allen. 2004. Semantics, dialogue, and reference resolution. In *Catalog-04: 8th Workshop on the Semantics and Pragmatics of Dialogue*.

J. Tetreault, M. Swift, P. Prithviraj, M. Dzikovska, and J. Allen. 2004. Discourse annotation in the monroe corpus. In *ACL-04 Discourse Annotation Workshop*.

K. Worm. 1998. A model for robust processing of spontaneous speech by integrating viable fragments. In *Proc. of COLING-ACL-98*, pages 1403–1407.

K. Zechner. 1998. Automatic construction of frame representations for spontaneous speech in unrestricted domains. In *Proc. of COLING-ACL-98*, pages 1448–1452.