PORTING PUNDIT TO THE RESOURCE MANAGEMENT DOMAIN

Lynette Hirschman, François-Michel Lang,
John Dowding, Carl Weir
Paoli Research Center
Unisys Defense Systems
P.O. Box 517
Paoli, PA 19301
Arpanet: hirsch@prc.unisys.com

INTRODUCTION

This paper describes our experiences porting the PUNDIT natural language processing system to the Resource Management domain. PUNDIT has previously been applied to a range of messages (see the paper *Analyzing Explicitly Structured Discourse in a Limited Domain: Trouble and Failure Reports* by C. Ball (appearing in this volume), and also [Hirschman1989]. However, it had not not been tested on any significant corpus of queries, such as that represented by the Resource Management corpus. Our goal was to assess PUNDIT's portability, and to determine its coverage of syntax over this domain. Time constraints precluded testing of the semantic component, but we plan to report on this at subsequent meetings. We performed this port with the intention of coupling PUNDIT to the MIT SUMMIT speech recognition system. This work is described in another paper in this volume, *Reducing Search by Partitioning the Word Network*, by J. Dowding.

Our philosophy in porting has been to *tune* the system to a new domain, rather than rewriting the grammar or building the grammar from scratch. The rationale for this approach is to continue to develop the coverage of PUNDIT's grammar; each new application should motivate principled extensions to the system that can also apply to other domains. Thus, over time, the coverage of PUNDIT has grown to cover a very large portion of English, and each succeeding port requires less effort. The disadvantage of this approach is that as the coverage grows, the grammar becomes "looser" -- the number of parses for any given word sequence tends to increase and also the grammar tends to overgenerate, letting through constructions that are not grammatical.

This philosophy is quite different from the "language modeling" approach taken by some groups working in speech recognition. The language modeling approach has as its goal the development of a *minimal* covering grammar needed to describe the phenomena observed in the particular corpus. The benefit of the language modeling approach is that it produces a very tight, highly constrained grammar. The disadvantage is the porting cost, and a very fragile system, whose syntactic boundaries are very easy to exceed.

Our approach to lexicon development has the same focus as our approach to syntactic coverage: to try to capture the general English definitions, rather than to limit ourselves to the particular domain-specific usages encountered in the training data. The rationale is also similar to that used in the syntactic component: generation of lexical entries is a time-consuming process; our goal is to develop a broad coverage system, so when entering a word in the lexicon, we enter the general English categories for the word. In many cases, this provides a much more general definition than

* This work has been supported in part by DARPA under contract N00014-85-C-0012, administered by the Office of Naval Research; and in part by internal Unisys R&D funding.

what is specifically required by an application. For example, the word *alert* occurs exclusively as a noun in the Resource Management domain. However, it must be classified as an adjective and a verb if the entry is made general to English.

The challenge for the broad-coverage grammar/lexicon approach is to develop methods of tuning the grammar and the lexicon to the particular corpus. It is clear that integration of PUNDIT with a speech recognition system will require that we bring to bear as many constraints as possible, in an attempt to prune the explosive search space that results from indeterminacy in analyzing the acoustic signal. We discuss several possible approaches to tuning both the grammar and the lexicon in the final section of the paper. What these results provide is a solid indication that our porting strategy is successful: only a very modest effort was required to obtain reasonable results in the Resource Management domain (85% of the training sentences and 76% of the test sentences received a correct parse, given a porting effort of 10 person-weeks). The next steps will be to add semantics and pragmatics, and to develop techniques for (semi-) automatically tuning the grammar to a new domain.

THE PORT

As mentioned above, in this initial experiment, we undertook only the syntactic processing of the Resource Management training and test corpus. In the PUNDIT system, the syntactic stage consists of the generation of a detailed surface parse tree and the construction of a regularized *Intermediate Syntactic Representation* or ISR. The ISR uses an operator/argument notation to represent the regularized syntax. The regularization includes insertion of omitted constituents in relative clause constructions or as a result of various raising and equi operations. In addition, we performed some limited experiments running with selection, with provides a shallow (selection-based) semantic filtering during parsing [Lang1988].

The tasks associated with the port are summarized below, with estimates of the time in person-weeks (PW). The total elapsed time was 1.5 months; the total port time was 10 person-weeks.

Steps in Porting PUNDIT
(4 PW) 1. Build lexicon
(3 PW) 2. Run training sentences
(1 PW) 3. Build Knowledge Base
(2 PW) 4. Collect selection patterns

THE LEXICON

The final lexicon consisted of approximately 1100 words; this number is greater than the usually quoted vocabulary size for the resource management corpus, due to the inclusion of a number of multi-word expressions in our lexicon, particularly for handling geographic names (*Bering Straights*, *Gulf of Tonkin*). Of these, approximately 450 words were already in our general lexicon (which is still quite small, some 5000 words). We entered the remaining 650 words. This total number represents a mix of general English entries (some 150 words), ship names (200), numbers (about 50, handled by the *shapes* component for productive expressions), place names (150), and some domain-specific entries (approximately 100), which were kept separate from the general English lexicon (e.g., *hfdf*).

SYNTAX

Changes to the syntax focused on adding coverage, but not removing any definitions. It even turned out that our treatment of fragmentary or incomplete sentences [Linebarger1988] was needed to run the resource management corpus, for sentences such as *The Kirk's distance from Horne?*. A few months prior to the beginning of the Resource Management port, we had added a comprehensive treatment of wh-expressions [Hirschman1988], which includes both relative clauses and question forms; at the same time, we had also added a treatment of imperatives. The fact that the grammar already contained these constructions made the port possible.

There were only some ten constructions that were missing from the grammar. Of these, the most significant was a detailed treatment of the comparative. Fortunately, most of these could be handled (syntactically) by treating the comparative *than* operator as a right adjunct to the word being modified, e.g., *than 12 knots* is a right-modifier of *greater* in *speed greater than 12 knots*. This required only that *than* be treated in the lexicon as a preposition. This certainly does not represent an adequate treatment of the comparative, and indeed, certain complex comparative constructions were not covered by this minimal treatment, for example *Is Puffer's position nearer to OSGP than Queenfish's location is?*.

Other additions to the grammar included:

1. A treatment for *what if* questions, based on the existing treatment of wh-expressions.
2. A treatment for prepositionless time expressions, e.g., *Monday* or *September 4*, etc.
3. A change to allow determiners to have left modifiers, as in *half the fuel* or *only these*.
4. A change to allow adjectives to have a certain class of left modifiers, as in *last three minutes*.
5. A change to allow multiple right noun adjuncts, as in *problems for Fanning that affect mission areas*.
6. A change to allow a preposed nominal argument to an adjective, as in *harpoon capable*.
7. A change to allow fraction expressions (e.g., *two thirds*).
8. Domain specific changes to handle degree expressions and the particular forms of dates encountered in the corpus.

These changes, coupled with a few changes to the restrictions, were sufficient to cover a very substantial portion of the corpus. Constructions that we did not cover (but which would require only modest grammar extensions to cover) include:

1. *or + comparative* as a right-modifier of comparative adjectives, e.g., *m5 or lower*.
2. Certain combinations of right noun adjuncts, e.g., *cruisers that are in the Indian Ocean that went to c2 August twenty*.
3. Questions containing the form *how + adjective* (*how bad*) and *how + adverb* (*how soon*). This hole accounted for a substantial portion of of the incorrectly parsed sentences.

SELECTION

One way to constrain the search space that results from a broad-coverage grammar and lexicon is to apply semantic constraints. Although we did not perform a deep semantic analysis, we did apply shallow semantic (selectional) constraints, to filter out semantically anomalous parses, in a second experiment. This procedure used PUNDIT's Selection Pattern Query and Response (SPQR) component [Lang1988]. We first used SPQR in acquisition mode, to collect semantic patterns, These patterns were then used to constrain search in parsing the test sentences.

The acquisition procedure queries the "domain expert" during parsing, whenever it finds a new pattern, such as a new subject-verb-object pattern, or a new adjective-noun pattern. The expert declares that the pattern is valid, allowing parsing to continue, or that the pattern is invalid, which causes backtracking to find a different analysis (and associated pattern). Information about valid and invalid patterns is stored in a pattern database; as the parser generates each phrase, it checks

the pattern database to see whether the expert has ruled on this pattern; if the user has already classified the pattern, then the user need not be queried again. Thus the system "learns" as it parses more sentences. Following the acquisition (or training) phase, the system can be run in one of two modes: allowing any unknown pattern to succeed (which will overgenerate, assuming that the set of patterns is incomplete), or forcing unknown patterns to fail, which will undergenerate.

To try to obtain maximum coverage of patterns, we generalized the patterns to *semantic class* patterns, rather than patterns of actual words. For example, the subject-verb-object word pattern

[Yorktown,decrease,speed],

can be generalized (using the taxonomy provided by the knowledge base) to the semantic class pattern (the suffix *_C* stands for concept):

[platform_C,change_C,transient_ship_attribute_C].

Previous experience had shown that use of word-level selectional patterns reduced the search by 20%, and the number of parses by a factor of three. We had hoped to achieve greater generality by use of the generalized semantic class patterns. However, due to time constraints, we were only able to process the first 100 training sentences, from which we collected some 450 patterns. This turned out (not surprisingly) to be far too small a set to generate any useful constraints in parsing. We therefore plan to complete our pattern collection on the full training set and rerun our experiment. This should provide us with a good measure of two things: the amount of pruning provided by application of shallow semantic constraints; and the amount of data that is required to obtain a complete set of patterns.

THE KNOWLEDGE BASE

Our experiment with generalization of semantic patterns required the use of a class hierarchy residing in a knowledge base. To support selection, we constructed a first pass at a knowledge base for the resource management domain. The KB contained some 750 concepts. One interesting observation that resulted from this exercise was that the semantic classes required for selection are not necessarily those classes that a knowledge engineer would develop as part of a domain model. In particular, certain words may exhibit similar distribution linguistically (e.g., *average* and *maximum*) but may not necessarily be collected under a single concept to permit easy generalization. For this reason, we may move to a more data-driven paradigm for building the knowledge base in our subsequent experiments.

THE METHODOLOGY

As previously stated, we added domain-independent rules to the grammar, and domain-independent entries to the lexicon, to cover the major constructions observed in the resource management corpus. We then trained on a (subset of) this corpus. The training involved parsing the first 200 sentences and examining and fixing parsing problems in these 200 sentences. We were able to collect semantic patterns only for the first 100 sentences. In both cases, this represents only a small fraction of the available training data (791 sentences). The sentences (training and test) were run on PUNDIT, under Quintus Prolog 2.2 on a Sun 3/60 with 8 MB of memory.

Because PUNDIT normally produces many parses, especially when run without selectional constraints, we allowed the system to run to a maximum of 15 parses per sentence. We report several results below, for purposes of comparison with other groups presenting parsing results. The first result is the number of sentences *obtaining a parse.* We believe that this is not a meaningful figure, however, since it is possible for a sentence to obtain a parse, but never to obtain a *correct* parse. For

this reason, we report a second result: the number of sentences obtaining a *correct parse* within the first 15 parses. In some cases, the system obtained a parse, but did NOT obtain the correct parse within the first 15 parses. In this case, we report it a NOT GETTING A CORRECT PARSE.

Our criteria for counting a parse correct were generally very stringent, and also required obtaining the correct regularized syntactic expression (or ISR). Our criteria included, for example: correct scoping of modifiers under conjunction; correct attachment of prepositional phrase and relative clause modifiers; and correct analysis of complex verb objects.

RESULTS

The table below shows the results obtained with parsing alone (no selectional constraints). We did not report the results obtained from selection, because it turned out that, given our very limited collection of patterns, selection failed to change the test results significantly. However, we plan to collect patterns for the entire training set and rerun this portion of the experiment.

There are several things worth noting in these results. First, the system is quite fast, even running to 15 parses: the average parse time to the correct parse is under 10 seconds for sentences averaging about 10 words/sentence. Second, although the correct parse appears on the average in the third parse, the first parse is correct more than 40% of that time. By adding semantic constraints, we expect to improve that figure substantially, thus driving down further the time to obtain the correct parse.

FUTURE DIRECTIONS

There are several directions that we plan to pursue. The first is to complete our experiments using selectional constraints to prune parses. A second general area that we will focus on over the next few months is the notion of how to *train* the system, that is, using the training set to customize the system to the given domain automatically. In particular, we plan to experiment with a "minimal" lexicon, to determine if we can improve our results by pruning out unneeded syntactic class information (e.g., just having *alert* entered as a noun for this domain). If pruning the lexicon improves our performance significantly, then we will experiment with various ways to use the

	Training (200 sentences)	Test (200 sentences)
Get A PARSE	94%	92%
Get A CORRECT PARSE using SYNTAX only	85%	76%
avg. # of correct parse	2.9	2.6
avg. # of parses/sentence	7.1	6.2
avg. secs. to correct parse	7.5	4.9
avg. secs. total	25.5	17.8

Parsing Results for the Resource Management Domain

training data to tune the system (in some automatic way) to "specialize" the lexicon to the particular application. Similarly, we plan to investigate techniques for using the training corpus to tune the parser to a new domain.

Our ultimate objective is to couple PUNDIT to a speech recognition system. To achieve this, we must focus not only on obtaining the correct parse, but on ruling out incorrect parses. So far, most development work has focused on extending coverage, and not on tightening the grammar to prevent overgeneration. Clearly, it is critical to address this problem if we plan to use a broad-coverage natural language system for spoken language understanding. This will also include developing metrics to measure overgeneration.

Finally, we expect to add the rules to support the in-depth semantic coverage that we have produced for our message domains. Overall, we are optimistic that by adding semantic constraints, plus extending the syntactic coverage in some quite limited ways, we will be able to exceed a 90% correct analysis rate on the test data, which brings the system within the bounds of a realistically useful system.

REFERENCES

[Hirschman1989]
Lynette Hirschman, Martha Palmer, John Dowding, Deborah Dahl, Marcia Linebarger, Rebecca Passonneau, Francois Lang, Catherine Ball, and Carl Weir, The PUNDIT Natural Language Processing System. In *Proc. of the Conference on Artificial Intelligence Systems in Government*, Washington, D.C., March 1989.

[Hirschman1988]
L. Hirschman, A Meta-Treatment of Wh-Constructions in English. In *Proc. of META88, Meta-Programming in Logic Programming*, Bristol, UK, June 1988.

[Lang1988]
F.-M. Lang and L. Hirschman, Improved Parsing Through Interactive Acquisition of Selectional Patterns. In *Proc. of the Second Conference on Applied Computational Linguistics*, Austin,TX, February, 1988.

[Linebarger1988]
M. Linebarger, D. Dahl, L. Hirschman, and R. Passonneau, Sentence Fragments Regular Structures. In *Proc. of the 1988 Annual Conference on Computational Linguistics*, Buffalo, NY, June 1988, pp. 7-16.