Parsing the Voyager Domain Using $\mathcal{P}$earl

David M. Magerman and *Mitchell P. Marcus*

CIS Department
University of Pennsylvania
Philadelphia, PA 19104

ABSTRACT

This paper[1] describes a natural language parsing algorithm for unrestricted text which uses a probability-based scoring function to select the "best" parse of a sentence according to a given grammar. The parser, $\mathcal{P}$earl, is a time-asynchronous bottom-up chart parser with Earley-type top-down prediction which pursues the highest-scoring theory in the chart, where the score of a theory represents the extent to which the context of the sentence predicts that interpretation. This parser differs from previous attempts at stochastic parsers in that it uses a richer form of conditional probabilities based on context to predict likelihood. $\mathcal{P}$earl also provides a framework for incorporating the results of previous work in part-of-speech assignment, unknown word models, and other probabilistic models of linguistic features into one parsing tool, interleaving these techniques instead of using the traditional pipeline architecture. In tests performed on the Voyager direction-finding domain, $\mathcal{P}$earl has been successful at resolving part-of-speech ambiguity, determining categories for unknown words, and selecting correct parses first using a very loosely fitting covering grammar.[2]

INTRODUCTION

All natural language grammars are ambiguous. Even tightly fitting natural language grammars are ambiguous in some ways. Loosely fitting grammars, which are necessary for handling the variability and complexity of unrestricted text and speech, are worse. The standard technique for dealing with this ambiguity, pruning grammars by hand, is painful, time-consuming, and usually arbitrary. The solution which many people have proposed is to use stochastic models to train statistical grammars automatically from a large corpus.

Attempts in applying statistical techniques to natural language parsing have exhibited varying degrees of success. These successful and unsuccessful attempts have suggested to us that:

- Stochastic techniques combined with traditional linguistic theories *can* (and indeed must) provide a solution to the natural language understanding problem.
- In order for stochastic techniques to be effective, they must be applied with restraint (poor estimates of context are worse than none[6]).
- Interactive, interleaved architectures are preferable to pipeline architectures in NLU systems, because they use more of the available information in the decision-making process.

We have constructed a stochastic parser, $\mathcal{P}$earl, which is based on these ideas.

The development of the $\mathcal{P}$earl parser is an effort to combine the statistical models developed recently into a single tool which incorporates all of these models into the decision-making component of a parser. While we have only attempted to incorporate a few simple statistical models into this parser, $\mathcal{P}$earl is structured in a way which allows any number of syntactic, semantic, and other knowledge sources to contribute to parsing decisions. The current implementation of $\mathcal{P}$earl uses Church's part-of-speech assignment trigram model, a simple probabilistic unknown word model, and a conditional probability model for grammar rules based on part-of-speech trigrams and parent rules.

By combining multiple knowledge sources and using a chart-parsing framework, $\mathcal{P}$earl attempts to handle a number of difficult problems. $\mathcal{P}$earl has the capability to parse word lattices, an ability which is useful in recognizing idioms in text processing, as well as in speech processing. The parser uses probabilistic training from a corpus to disambiguate between grammatically acceptable structures, such as determining prepositional phrase attachment and conjunction scope. Finally, $\mathcal{P}$earl maintains a well-formed substring table within its chart to allow for partial parse retrieval. Partial parses are useful both for error-message generation and for processing ungrammatical or incomplete sentences.

For preliminary tests of $\mathcal{P}$earl's capabilities, we are using the Voyager direction-finding domain, a spoken-language system developed at MIT.[3] We have selected this domain for a number of reasons. First, it exhibits the attachment regularities which we are trying to capture with the context-sensitive probability model. Also, since both MIT and Unisys have developed parsers and grammars for this domain, there are existing parsers with which we can compare $\mathcal{P}$earl. Finally, $\mathcal{P}$earl's dependence on a parsed corpus to train its models and to derive its grammar

[1]This work was partially supported by DARPA grant No. N0014-85-K0018, ONR contract No. N00014-89-C-0171 by DARPA and AFOSR jointly under grant No. AFOSR-90-0066, and by ARO grant No. DAAL 03-89-C0031 PRI. Special thanks to Carl Weir and Lynette Hirschman at Unisys for their valued input, guidance and support.

[2]The grammar used for our experiments is the string grammar used in Unisys' PUNDIT natural language understanding system.

[3]Special thanks to Victor Zue at MIT for the use of the speech data from MIT's Voyager system.

required that we use a domain for which a parsed corpus existed. A corpus of 1100 parsed sentences was generated by the Unisys' PUNDIT Language Understanding System. These parse trees were evaluated to be semantically correct by PUNDIT's semantics component, although no hand-verification of this corpus was performed. PUNDIT's parser uses a string grammar with many complicated, hand-generated restrictions. The goal of the experiments we performed was to reproduce (or improve upon) the parsing accuracy of PUNDIT using just the context-free backbone of the PUNDIT grammar, without the hand-generated restrictions and, equally important, without the benefit of semantic analysis.

In a test on 40 Voyager sentences excluded from the training material, $\mathcal{P}$earl has shown promising results in handling part-of-speech assignment, prepositional phrase attachment, and unknown word categorization. $\mathcal{P}$earl correctly parsed 35 out of 40 or 87.5% of these sentences, where a *correct* parse is defined to mean one which would produce a correct response from the Voyager system. We will describe the details of this experiment later.

In this paper, we will first explain our contribution to the stochastic models which are used in $\mathcal{P}$earl: a context-free grammar with context-sensitive conditional probabilities. Then, we will describe the parser's architecture and the parsing algorithm. Finally, we will give the results of experiments we performed using $\mathcal{P}$earl which explore its capabilities.

USING STATISTICS TO PARSE

Recent work involving context-free and context-sensitive probabilistic grammars provide little hope for the success of processing unrestricted text using probabilistic techniques. Works by Chitrao and Grishman[3] and by Sharman, Jelinek, and Mercer[11] exhibit accuracy rates lower than 50% using *supervised training*. Supervised training for probabilistic CFGs requires parsed corpora, which is very costly in time and man-power[2].

In our investigations, we have made two observations which attempt to explain the lack-luster performance of statistical parsing techniques:

- Simple probabilistic CFGs provide *general* information about how likely a construct is going to appear anywhere in a sample of a language. This average likelihood is often a poor estimate of probability.
- Parsing algorithms which accumulate probabilities of parse theories by simply multiplying them over-penalize infrequent constructs.

$\mathcal{P}$earl avoids the first pitfall by using a context-sensitive conditional probability CFG, where context of a theory is determined by the theories which predicted it and the part-of-speech sequences in the input sentence. To address the second issue, $\mathcal{P}$earl scores each theory by using the geometric mean of the contextual conditional probabilities of all of the theories which have contributed to that theory. This is equivalent to using the sum of the logs of these probabilities.

CFG with context-sensitive conditional probabilities

In a very large parsed corpus of English text, one finds that the most frequently occurring noun phrase structure in the text is a noun phrase containing a determiner followed by a noun. Simple probabilistic CFGs dictate that, given this information, "determiner noun" should be the most likely interpretation of a noun phrase.

Now, consider only those noun phrases which occur as subjects of a sentence. In a given corpus, you might find that pronouns occur just as frequently as "determiner noun"s in the subject position. This type of information can easily be captured by conditional probabilities.

Finally, assume that the sentence begins with a pronoun followed by a verb. In this case, it is quite clear that, while you can probably concoct a sentence which fits this description and does not have a pronoun for a subject, the first theory which you should pursue is one which makes this hypothesis.

The context-sensitive conditional probabilities which $\mathcal{P}$earl uses take into account the immediate parent of a theory[4] and the part-of-speech trigram centered at the beginning of the theory.

For example, consider the sentence:

My first love was named $\mathcal{P}$earl.
(no subliminal propaganda intended)

A theory which tries to interpret "love" as a verb will be scored based on the part-of-speech trigram "adjective verb verb" and the parent theory, probably "S $\rightarrow$ NP VP." A theory which interprets "love" as a noun will be scored based on the trigram "adjective noun verb." Although lexical probabilities favor "love" as a verb, the conditional probabilities will heavily favor "love" as a noun in this context.[5]

Using the Geometric Mean of Theory Scores

According to probability theory, the likelihood of two *independent* events occurring at the same time is the product of their individual probabilities. Previous statistical parsing techniques apply this definition to the cooccurrence of two theories in a parse, and claim that the likelihood of the two theories being correct is the product of the probabilities of the two theories.

This application of probability theory ignores two vital observations about the domain of statistical parsing:

- Two constructs occurring in the same sentence are not necessarily independent (and frequently are not). If the independence assumption is violated, then the product of individual probabilities has no meaning with respect to the joint probability of two events.
- Since statistical parsing suffers from sparse data, probability estimates of low frequency events will usually be inaccurate estimates. Extreme underestimates of the likelihood of low frequency events will produce misleading joint probability estimates.

[4] The parent of a theory is defined as a theory with a CF rule which contains the left-hand side of the theory. For instance, if "S $\rightarrow$ NP VP" and "NP $\rightarrow$ det n" are two grammar rules, the first rule can be a parent of the second, since the left-hand side of the second "NP" occurs in the right-hand side of the first rule.

[5] In fact, the part-of-speech tagging model which is also used in $\mathcal{P}$earl will heavily favor "love" as a noun. We ignore this behavior to demonstrate the benefits of the trigram conditioning.

From these observations, we have determined that estimating joint probabilities of theories using individual probabilities is too difficult with the available data. We have found that the geometric mean of these probability estimates provides an accurate assessment of a theory's viability.

The Actual Theory Scoring Function

In a departure from standard practice, and perhaps against better judgment, we will include a precise description of the theory scoring function used by $\mathcal{P}$earl. This scoring function tries to solve some of the problems noted in previous attempts at probabilistic parsing[3][11]:

- Theory scores should not depend on the length of the string which the theory spans.
- Sparse data (zero-frequency events) and even zero-probability events do occur, and should not result in zero scoring theories.
- Theory scores should not discriminate against unlikely constructs when the context predicts them.

In this discussion, a theory is defined to be a partial or complete syntactic interpretation of a word string, or, simply, a parse tree. The raw score of a theory, θ, is calculated by taking the product of the conditional probability of that theory's CFG rule given the context, where context is a part-of-speech trigram centered at the beginning of the theory and a parent theory's rule, and the score of the contextual trigram:

$$SC_{\text{raw}}(\theta) = \mathcal{P}(rule_\theta | (p_0 p_1 p_2), rule_{\text{parent}}) sc(p_0 p_1 p_2)$$

Here, the score of a trigram is the product of the mutual information of the part-of-speech trigram,[6] $p_0 p_1 p_2$, and the lexical probability of the word at the location of p_1 being assigned that part-of-speech p_1.[7] In the case of ambiguity (part-of-speech ambiguity or multiple parent theories), the maximum value of this product is used. The score of a partial theory or a complete theory is the geometric mean of the raw scores of all of the theories which are contained in that theory.

Theory Length Independence This scoring function, although heuristic in derivation, provides a method for evaluating the value of a theory, regardless of its length. When a rule is first predicted (Earley-style), its score is just its raw score, which represents how much the context predicts it. However, when the parse process hypothesizes interpretations of the sentence which reinforce this theory, the geometric mean of all of the raw scores of the rule's subtree is used, representing the overall likelihood of the theory given the context of the sentence.

Low-frequency Events Although some statistical natural language applications employ backing-off estimation techniques[10][5] to handle low-frequency events, $\mathcal{P}$earl uses a very simple estimation technique, reluctantly attributed to Church[6]. This technique estimates the probability of an event by adding 0.5 to every frequency count.[8] Low-scoring theories *will* be predicted by the Earley-style parser. And, if no other hypothesis is suggested, these theories will be pursued. If a high scoring theory advances a theory with a very low raw score, the resulting theory's score will be the geometric mean of all of the raw scores of theories contained in that theory, and thus will be much higher than the low-scoring theory's score.

Example of Scoring Function As an example of how the conditional-probability-based scoring function handles ambiguity, consider the sentence

Fruit flies like a banana.

in the domain of insect studies. Lexical probabilities should indicate that the word "flies" is more likely to be a plural noun than a tensed verb. This information is incorporated in the trigram scores. However, when the interpretation

$$S \rightarrow .\ NP\ VP$$

is proposed, two possible NPs will be parsed,

$$NP \rightarrow \text{noun (fruit)}$$

and

$$NP \rightarrow \text{noun noun (fruit flies)}.$$

Since this sentence is syntactically ambiguous, if the first hypothesis is tested first, the parser will interpret this sentence incorrectly.

However, this will not happen in this domain. Since "fruit flies" is a common idiom in insect studies, the score of its trigram, noun noun verb, will be much greater than the score of the trigram, noun verb verb. Thus, not only will the lexical probability of the word "flies/verb" be lower than that of "flies/noun," but also the raw score of "NP → noun (fruit)" will be lower than that of "NP → noun noun (fruit flies)," because of the differential between the trigram scores.

So, "NP → noun noun" will be used first to advance the "S → . NP VP" rule. Further, even if the parser advances both NP hypotheses, the "S → NP . VP" rule using "NP → noun noun" will have a higher score than the "S → NP . VP" rule using "NP → noun."

INTERLEAVED ARCHITECTURE IN $\mathcal{P}$EARL

The interleaved architecture implemented in $\mathcal{P}$earl provides many advantages over the traditional pipeline architecture, but it also introduces certain risks. Decisions about word and part-of-speech ambiguity can be delayed until syntactic processing can

[6] The mutual information of a part-of-speech trigram, $p_0p_1p_2$, is defined to be $\frac{\mathcal{P}(p_0p_1p_2)}{\mathcal{P}(p_0 x p_2)\mathcal{P}(p_1)}$, where x is any part-of-speech. See [4] for further explanation.

[7] The trigram scoring function actually used by the parser is somewhat more complicated than this.

[8] We are not deliberately avoiding using all probability estimation techniques, only those backing-off techniques which use independence assumptions that frequently provide misleading information when applied to natural language.

disambiguate them. And, using the appropriate score combination functions, the scoring of ambiguous choices can direct the parser towards the most likely interpretation efficiently.

However, with these delayed decisions comes a vastly enlarged search space. The effectiveness of the parser depends on a majority of the theories having very low scores based on either unlikely syntactic structures or low scoring input (such as low scores from a speech recognizer or low lexical probability). In experiments we have performed, this has been the case.

The Parsing Algorithm

$\mathcal{P}$earl is an agenda-based time-asynchronous bottom-up chart parser with Earley-type top-down prediction. The significant difference between $\mathcal{P}$earl and non-probabilistic bottom-up parsers is that instead of completely generating all grammatical interpretations of a word string, $\mathcal{P}$earl uses an agenda to order the incomplete theories in its chart to determine which theory to advance next. The agenda is sorted by the value of the theory scoring function described above. Instead of expanding all theories in the chart, $\mathcal{P}$earl pursues the highest-scoring incomplete theories in the chart, advancing up to N theories at each pass. However, $\mathcal{P}$earl parses *without pruning*. Although it is only advancing N incomplete theories at each pass, it retains the lower scoring theories in its agenda. If the higher scoring theories do not generate viable alternatives, the lower scoring theories may be used on subsequent passes.

The parsing algorithm begins with an input word lattice, which describes the input sentence and includes possible idiom hypothese and may include alternative word hypotheses.[9] Lexical rules for the input word lattice are inserted into the parser's chart. Using Earley-type prediction, a sentence (S) is predicted at the beginning of the input, and all of the theories which are predicted by that initial sentence are inserted into the chart. These incomplete theories are scored according to the context-sensitive conditional probabilities and the trigram part-of-speech model. The incomplete theories are tested in order by score, until N theories are advanced.[10] The resulting advanced theories are scored and predicted for, and the new incomplete predicted theories are scored and added to the chart. This process continues until an complete parse tree is determined, or until the parser decides, heuristically, that it should not continue. The heuristics we used for determining that no parse can be found for an input are based on the highest scoring incomplete theory in the chart, the number of passes the parser has made, and the size of the chart.

$\mathcal{P}$earl's Capabilities

Besides using statistical methods to guide the parser through the parsing search space, $\mathcal{P}$earl also performs other functions which are crucial to robustly processing unrestricted natural language text and speech.

Handling Unknown Words $\mathcal{P}$earl uses a very simple probabilistic unknown word model to hypothesize categories for unknown words. When a word is found which is unknown to the system's lexicon, the word is assumed to be any one of the open class categories. The lexical probability given a category is the probability of that category occurring in the training corpus.

Idiom Processing and Lattice Parsing Since the parsing search space can be simplified by recognizing idioms, $\mathcal{P}$earl allows the input string to include idioms that span more than one word in the sentence. This is accomplished by viewing the input sentence as a word lattice instead of a word string. Since idioms tend to be unambiguous with respect to part-of-speech, they are generally favored over processing the individual words that make up the idiom, since the scores of rules containing the words will tend to be less than 1, while a syntactically appropriate, unambiguous idiom will have a score of close to 1.

The ability to parse a sentence with multiple word hypotheses and word boundary hypotheses makes $\mathcal{P}$earl very useful in the domain of spoken language processing. By delaying decisions about word selection but maintaining scoring information from a speech recognizer, the parser can use grammatical information in word selection without slowing the speech recognition process. Because of $\mathcal{P}$earl's interleaved architecture, one could easily incorporate scoring information from a speech recognizer into the set of scoring functions used in the parser. $\mathcal{P}$earl could also provide feedback to the speech recognizer about the grammaticality of fragment hypotheses to guide the recognizer's search.

Partial Parses The main advantage of chart-based parsing over other parsing algorithms is that a chart-based parser can recognize well-formed substrings within the input string in the course of pursuing a complete parse. $\mathcal{P}$earl takes full advantage of this characteristic. Once $\mathcal{P}$earl is given the input sentence, it awaits instructions as to what type of parse should be attempted for this input. A standard parser automatically attempts to produce a sentence (S) spanning the entire input string. However, if this fails, the semantic interpreter might be able to derive some meaning from the sentence if given non-overlapping noun, verb, and prepositional phrases. If a sentence fails to parse, requests for partial parses of the input string can be made by specifying a range which the parse tree should cover and the category (NP, VP, etc.). These requests, however, must be initiated by an intelligent semantics processor which can manipulate these partial parses.

Trainability One of the major advantages of the probabilistic parsers is trainability. The conditional probabilities used by $\mathcal{P}$earl are estimated by using frequencies from a large corpus of parsed sentences. The parsed sentences must be parsed using the grammar formalism which the $\mathcal{P}$earl will use.

Assuming the grammar is not recursive in an unconstrained way, the parser can be trained in an unsupervised mode. This is accomplished by running the parser without the scoring functions, and generating many parse trees for each sentence. Previous work[11] has demonstrated that the correct information from

[9] Using alternative word hypotheses without incorporating a speech recognition model would not necessarily produce useful results. Given two unambiguous nouns at the same position in the sentence, $\mathcal{P}$earl has no information with which to disambiguate these words, and will invariably select the first one entered into the chart. The capability to process a alternate word hypotheses is included to suggest the future implementation of a speech recognition model in $\mathcal{P}$earl.

[10] We believe that N depends on the perplexity of the grammar used, but for the string grammar used for our experiments we used N=3. For the purposes of training, we suggest that a higher N should be used in order to generate more parses.

[11] This is an unpublished result, reportedly due to Fujisaki at IBM Japan.

these parse trees will be reinforced, while the incorrect substructure will not. Multiple passes of re-training using frequency data from the previous pass should cause the frequency tables to converge to a stable state. This hypothesis has not yet been tested.[12]

An alternative to completely unsupervised training is to take a parsed corpus for any domain of the same language using the same grammar, and use the frequency data from that corpus as the initial training material for the new corpus. This approach should serve only to minimize the number of unsupervised passes required for the frequency data to converge.

PARSING THE VOYAGER DOMAIN

In order to test $\mathcal{P}$earl's capabilities, we performed some simple tests to determine if its performance is at least consistent with the premises upon which it is based. The test sentences used for this evaluation are *not* from the training data on which the parser was trained. Using $\mathcal{P}$earl's context-free grammar, which is equivalent to the context-free backbone of PUNDIT's grammar, these test sentences produced an average of 64 parses per sentence, with some sentences producing over 100 parses.

Overall Parsing Accuracy

The 40 test sentences were parsed by $\mathcal{P}$earl and the highest scoring parse for each sentence was compared to the correct parse produced by PUNDIT. Of these 40 sentences, $\mathcal{P}$earl produced parse trees for 38 of them, and 35 of these parse trees were equivalent to the correct parse produced by PUNDIT, for an overall accuracy rate of 88%. Although precise accuracy statistics are not available for PUNDIT, this result is believed to be comparable to PUNDIT's performance. However, the result is achieved without the painfully hand-crafted restriction grammar associated with PUNDIT's parser.

Many of the test sentences were not difficult to parse for existing parsers, but most had *some* grammatical ambiguity which would produce multiple parses. In fact, on 2 of the 3 sentences which were incorrectly parsed, $\mathcal{P}$earl produced the correct parse as well, but the correct parse did not have the highest score. And both of these sentences would have been correctly processed if semantic filtering were used on the top three parses.

Of the two sentences which did not parse, one used passive voice, which only occurred in one sentence in the training corpus. While the other sentence,

```
How can I get from cafe sushi to Cambridge
City Hospital by walking
```

did not produce a parse for the entire word string, it could be processed using $\mathcal{P}$earl's partial parsing capability. By accessing the chart produced by the failed parse attempt, the parser can find a parsed sentence containing the first eleven words, and a prepositional phrase containing the final two words. This information could be used to interpret the sentence properly.

[12] In fact, for certain grammars, the frequency tables may not converge at all, or they may converge to zero, with the grammar generating no parses for the entire corpus. This is a worst-case scenario which we do not anticipate happening.

Unknown Word Part-of-speech Assignment

To determine how $\mathcal{P}$earl handles unknown words, we randomly selected five words from the test sentences, *I, know, tee, describe,* removed their entries from the lexicon, and *station*, and tried to parse the 40 sample sentences using the simple unknown word model previously described.[13]

In this test, the pronoun, *I*, was assigned the correct part-of-speech 9 of 10 times it occurred in the test sentences. The nouns, *tee* and *station*, were correctly tagged 4 of 5 times. And the verbs, *know* and *describe*, were correctly tagged 3 of 3 times. While this accuracy is expected for unknown words in isolation, based on the accuracy of the part-of-speech tagging model, the performance is expected to degrade for sequences of unknown words.

Category	Accuracy
pronoun	90%
noun	80%
verb	100%
overall	89%

Figure 1: Performance on Unknown Words in Test Sentences

Prepositional Phrase Attachment

Accurately determining prepositional phrase attachment in general is a difficult and well-documented problem. However, based on experience with several different domains, we have found prepositional phrase attachment to be a domain-specific phenomenon for which training can be very helpful. For instance, in the direction-finding domain, *from* and *to* prepositional phrases generally attach to the preceding verb and not to any noun phrase. This tendency is captured in the training process for $\mathcal{P}$earl and is used to guide the parser to the more likely attachment with respect to the domain. This does not mean that $\mathcal{P}$earl will get the correct parse when the less likely attachment is correct; in fact, $\mathcal{P}$earl will invariably get this case wrong. However, based on the premise that this is the less likely attachment, this will produce more correct analyses than incorrect. And, using a more sophisticated statistical model which uses more contextual information, this performance can likely be improved.

$\mathcal{P}$earl's performance on prepositional phrase attachment was very high (54/55 or 98.2% correct). The reason the accuracy rate is so high is that the direction-finding domain is very consistent in its use of individual prepositions. The accuracy rate is not expected to be as high in less consistent domains, although we expect it to be significantly higher than chance.

Search Space Reduction

One claim of $\mathcal{P}$earl, and of probabilistic parsers in general, is that probabilities can help guide a parser through the immense search space produced by ambiguous grammars. Since, without probabilisties, the test sentences produced an average of 64 parses per sentence, $\mathcal{P}$earl unquestionably has reduced the space of possibilities by only producing 3 parses per sentence while maintaining

[12] The unknown word model used in this test was augmented to include closed class categories as well as open class, since the words removed from the lexicon may have included (in fact did include) closed class words.

Prep.	Accuracy
from	92%
to	100%
on	100%
Overall	98.2%

Figure 2: Accuracy Rate for Prepositional Phrase Attachment, by Preposition

high accuracy. However, it is interesting to see how Pearl's scoring function performs against previously proposed scoring functions. The four scoring functions compared include a simple probabilistic CFG, where each context-free rule is assigned a fixed likelihood based on training, a CFG using probabilistic conditioning on the parent rule only, which is similar to the scoring function used by Chitrao and Grishman[3], and two versions of the CFG with CSP model, one using the geometric mean of raw theory scores and the other using the product of these raw scores. Using

Technique	Edges	Accuracy
P-CFG	929	35%
CFG with Parent Cond.	883	50%
CFG with CSP	210	88%
Prod. of Scores	657	60%

Figure 3: Search Space Reduction and Accuracy for Four Probabilistic Models

a simple probabilistic CFG model, the parser produced a much lower accuracy rate (35%). The parental conditioning brought this rate up to 50%, and the trigram conditioning brought this level up to 88%. The search space for CFG with CSP was 4 to 5 times lower than the simple probabilistic CFG.

FUTURE WORK

The Pearl parser takes advantage of domain-dependent information to select the most appropriate interpretation of an input. However, the statistical measure used to disambiguate these interpretations is sensitive to certain attributes of the grammatical formalism used, as well as to the part-of-speech categories used to label lexical entries. All of the experiments performed on Pearl thus far have been using one grammar, one part-of-speech tag set, and one domain (because of availability constraints). Future experiments are planned to evaluate Pearl's performance on different domains, as well as on a general corpus of English, and on different grammars, including a grammar derived from a manually parsed corpus.

Specifically, we plan to retrain Pearl on a corpus of terrorist-related messages from the Message Understanding Conference (MUC). Using this material, we will attempt two very different experiments. The first experiment will be similar to the one performed on the Voyager data. Using a corpus of correctly parsed MUC sentences from SRI's Tacitus system, we will derive a context-free grammar and extract training statistics for Pearl's models. Since the MUC sentences exhibit many more difficulties than Voyager, including 50 word sentences, punctuation, no sentence markers, and typographical errors, we expect Pearl to require significant re-engineering to handle this experiment.

The second experiment on the MUC corpus involves extracting a grammar and training statistics from a hand-parsed corpus. When the University of Pennsylvania's Treebank project[2] makes a hand-parsed version of the MUC training material available to the DARPA community, we will extract a context-free grammar from these parse trees, and retrain Pearl on this material. This experiment is even more interesting because, if successful, it will show that Pearl provides an alternative to the hand-pruning of grammars to cover specific domains. If a hand-parsed corpus can provide a covering grammar which can be used to accurately parse a particular domain, porting natural language applications to new domains will be greatly facilitated.

CONCLUSION

The probabilistic parser which we have described provides a platform for exploiting the useful information made available by statistical models in a manner which is consistent with existing grammar formalisms and parser designs. Pearl can be trained to use any context-free grammar, accompanied by the appropriate training material. And, the parsing algorithm is very similar to a standard bottom-up algorithm, with the exception of using theory scores to order the search.

In experiments on the Voyager direction-finding domain, Pearl, using only a context-free grammar and statistical models, performed at least as well as PUNDIT's parser, which includes hand-generated restrictions. In the future, we hope to demonstrate similar performance on more difficult domains and using manually parsed corpora.

REFERENCES

[1] Ayuso, D., Bobrow, R., et. al. 1990. Towards Understanding Text with a Very Large Vocabulary. In Proceedings of the June 1990 DARPA Speech and Natural Language Workshop. Hidden Valley, Pennsylvania.

[2] Brill, E., Magerman, D., Marcus, M., and Santorini, B. 1990. Deducing Linguistic Structure from the Statistics of Large Corpora. In Proceedings of the June 1990 DARPA Speech and Natural Language Workshop. Hidden Valley, Pennsylvania.

[3] Chitrao, M. and Grishman, R. 1990. Statistical Parsing of Messages. In Proceedings of the June 1990 DARPA Speech and Natural Language Workshop. Hidden Valley, Pennsylvania.

[4] Church, K. 1988. A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text. In Proceedings of the Second Conference on Applied Natural Language Processing. Austin, Texas.

[5] Church, K. and Gale, W. 1990. Enhanced Good-Turing and Cat-Cal: Two New Methods for Estimating Probabilities of English Bigrams. *Computers, Speech and Language.*

[6] Gale, W. A. and Church, K. 1990. Poor Estimates of Context are Worse than None. In Proceedings of the June 1990 DARPA Speech and Natural Language Workshop. Hidden Valley, Pennsylvania.

[7] Hindle, D. 1988. Acquiring a Noun Classification from Predicate-Argument Structures. Bell Laboratories.

[8] Hindle, D. and Rooth, M. 1990. Structural Ambiguity and Lexical Relations. In Proceedings of the June 1990 DARPA Speech and Natural Language Workshop. Hidden Valley, Pennsylvania.

[9] Jelinek, F. 1985. Self-organizing Language Modeling for Speech Recognition. IBM Report.

[10] Katz, S. M. 1987. Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer. *IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-35, No. 3.*

[11] Sharman, R. A., Jelinek, F., and Mercer, R. 1990. In Proceedings of the June 1990 DARPA Speech and Natural Language Workshop. Hidden Valley, Pennsylvania.