Parsing Spoken Phrases Despite Missing Words

Wayne H. Ward
Alexander G. Hauptmann
Richard M. Stern
Thomas Chanak

Department of Computer Science
Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Paper to be presented at the IEEE International Conference on
Acoustics, Speech, and Signal Processing, April, 1988

PARSING SPOKEN PHRASES DESPITE MISSING WORDS

ABSTRACT

This paper compares the recognition accuracy obtained in forming sentence hypotheses using island-driven sentence parsers with parsers that hypothesize sentences in left-to-right fashion. Island-driven parsing algorithms are especially valuable in speech recognition systems because they can function more gracefully when not all of the correct words of an utterance were produced by the word hypothesizer. The inputs to both types of parsers consist of a lattice of candidate words, which are identified by their begin and end times, and the quality of the acoustic-phonetic match. Grammatical constraints are expressed by trigram models of sequences of lexical and semantic labels. We found that the island-driven parser produces parses with a higher percentage of correct words than the left-to-right parser in all cases considered. When the quality of the input lattices is extremely high, differences in parsing accuracy can be directly attributed to the superior ability of the island-driven parser to handle lattices with missing words.

INTRODUCTION

In this paper we compare the recognition performance and computational burden of left-to-right and island-driven parsing algorithms for parsing phrases and sentences when the input to the parser is in the form of a lattice of unlinked word candidates. This type of parser input is frequently produced by speech recognition systems that consist of a combination of heterogeneous knowledge sources such as the HEARSAY-II [erman79] and HWIM [wolf80] systems. In all of these cases, the parser must be able to detect cases in which lower levels of the system fail to hypothesize one or more of the correct words in an utterance, and ensure that the remaining correctly hypothesized words are not discarded as well.

Our experiments in sentence parsing are performed in the context of the ANGEL system, a large-vocabulary speaker-independent speech recognition system which is presently under development at Carnegie Mellon University. The system includes a feature-based acoustic-phonetic hypothesizer [cole86], an island-driven word hypothesizer [rudnicky87], and several different implementations of a sentence parser that converts the outputs of the word hypothesizer into sentence candidates.

We have explored several schemes for representing syntactic and semantic knowledge in these parsers, including case frames [hayes86] and simple statistical models of sequences of syntactic and semantic categories of the word candidates. Most of the statistical grammars make use of a second-order Markov model to represent local syntactic and semantic phenomena. In order to reduce the storage and training required to effectively model word usage in tasks with very large vocabularies, we have made use of "trigram" representations of language that are expressed in terms of probabilities of sequences of lexical or semantic labels or "tags", rather than the individual words in the vocabulary themselves. For example, *DAY-OF-WEEK* is a category to which the names of days of the week belong. This approach has also been pursued by a small number of other research groups.

In initial experiments with the probabilistic grammars we compared the effect of the specificity of grammatical constraint on recognition accuracy by artificially imposing various different types of degradations on the output of the word hypothesizer using a left-to-right parsing strategy [stern87]. We

found that increasing the specificity of the trigram representation for a particular task domain tended to improve performance when the correct words are not among the very best word candidates, but that the most highly constrained grammars could result in very poor performance if some of the correct words are missing completely from the input word lattices. These problems were a result of the strict left-to-right nature of the parsing process used in the original study, as will be discussed below. Unfortunately, the large-vocabulary, speaker-independent, continuous-speech environment is the hardest of all tasks for a speech recognizer, and it is inevitable that some words will occasionally be missing from the word lattices that are hypothesized in bottom-up fashion. Because of this problem we have constructed a series of parsers using island-driven algorithms as an alternative to left-to-right parsing, even though performance of the acoustic-phonetic and word hypothesization modules continues to improve.

Left-to-right parsers are fairly well known and understood, and the approach has received extensive treatment in the literature. While left-to-right parsers can be made extremely efficient [Earley-parser], less is know about using this strategy in the face of errorful input. The Harpy system at CMU [Harpy] pioneered the beam search technique to parse left-to-right while tracing through a finite state net, and to date this is still the most viable approach to speech parsing. However, the real problem with lattice input (as in the ANGEL system) is dealing with missing words, which may throw the whole parse beam off the correct track.

Island parsing offers promise in this respect. The concept is familiar to many natural language researchers familiar with parsing text, e.g. [Multipar]. Island parsing has also been mentioned as a possible approach in some speech parsing studies [hayes86, Gatward86]. However, few researchers have described results using these parsers with anything except the smallest of grammars [islandparser]. While the BBN HWIM system [wolf80] included an island parser, it was not used in the official system evaluation because of speed considerations. This highlights the fundamental problem with island parsing: the number of islands and the number of connections between islands can become quite large in a typical semantic net. Every island depicts a different partial path through the network and must be represented separately. Several islands might have the same words, but represent different partial paths of the semantic network.

We believe that the use of trigram grammars rather than finite-state network grammars can significantly reduce the complexity of the island-parsing algorithms because only very limited (and local) context is taken into consideration. In network-based island driving, the context necessary to add a new word to a partial phrase island is as large as the complete phrase island. The context in trigram parsing is limited so that we must look at no more than two words to add a new word. This trigram extension procedure can be precomputed into a highly efficient table lookup. It is much more difficult to find the continuations of an arbitrarily long sequence of words which constitutes a phrase island in a network. In a typical parse we attempt several hundred thousand of these searches, so the savings obtained by resorting to the trigram representation are significant.

In the following paragraphs, we describe the overall speech system used in our experiments as well as the two parsing algorithms. Then we describe the data used for this comparison and how they were derived, followed by the results. A discussion of our findings concludes the report.

SENTENCE PARSING IN THE ANGEL SYSTEM

When a sentence is presented to the CMU recognition system, the lattice produced by the word hypothesizer contains a large number of candidate words that are each characterized by a begin time, an end time, and an acoustic-phonetic plausibility score.

New phrases are created by attempting to add new words to existing phrases. The following types of knowledge are considered when adding a new word to a candidate phrase:

- **Word score** - The word score represents the likelihood for the word based on acoustic-phonetic evidence, provided by the word hypothesizer.
- **Word-juncture quality** - The quality of the acoustic-phonetic juncture between two words is scored by the junction verifier in the word hypothesizer, based on tables of penalties for overlaps and gaps.
- **Syntactic and semantic information** - Syntactic and semantic plausibility is indicated by the trigram approximation to the probability of observing the hypothesized sequence of syntactic and semantic tags.

The score for a phrase is a linear combination of the scores provided by each of the above knowledge sources. The weights used to combine these scores were determined empirically from training data, and the recognition accuracy of the parsers is relatively insensitive to their exact value. While this procedure is somewhat *ad hoc* we found that it yields parsing accuracy that is comparable to that obtained with a consistently-normalized scoring strategy such as that used in the HWIM system [bartschat87].

We now consider the specifics of the left-to-right and island-driven implementations of the parsers in more detail.

Left-to-right parser. As the name implies, the left-to-right parser builds phrases by starting at the beginning of an utterance and adding words to the end of current phrase hypotheses until the end of the utterance is reached. Grammatical constraints are applied as each word is considered for the extension of current phrases. A beam search is used to limit the number of possibilities, so that only phrases within a predefined range of the best-scoring phrase are kept. Since this type of parser immediately applies the maximum amount of constraint as new phrases are being created, it can run into problems when some words are missing or have very bad scores. If a correct word is missing from the input lattice it is likely that the future portion of the otherwise correct partial parse will not be correctly extended. In order to continue, another word with the same class of tag must be (incorrectly) hypothesized with a begin and end time that are very similar to the missing word. If no such word candidate exists, only words with incorrect tags or incorrect word boundaries will be used to extend the phrases. If the correct word is present with a bad score, left-to-right beam searches are prone to garden-path problems. More favorable paths will be followed that later prove to be incorrect while the correct path is pruned. Wild-card extensions might allow any word to be inserted by the parser, but these will drastically reduce parser efficiency and still suffer from garden-path problems.

Island-driven parser. The island-driven parser forms phrase islands, which are sequences of words that do not span the complete utterance. An initial list of phrase candidates that are single-word islands is first created from words that meet extremely tight scoring criteria. Existing islands are then extended by concatenating any word or sequence of words that is grammatical and that can juncture to them. Words that were used to form the initial single-word islands may not also be used to extend other islands.

During the extension phase, any two islands that can juncture with each other and are grammatical can be concatenated to form a single larger island. Once the extension phase is completed, a queue of "complete phrases" is created from all of the candidate phrases that span the entire utterance.

The island-driven parser also has a partial capability of top-down insertion of words that were not hypothesized in bottom-up fashion by the word matcher. Specifically, each partial phrase can be joined to other partial phrases by assuming that only a single missing word exists between two adjacent phrase islands. In these cases, the most probable sequence of tags is assumed for the three-word sequence consisting of the last word from the first phrase island, an unknown missing word, and the first word from the second phrase island. If there is only one word in the lexicon that is characterized by the most likely (inserted) second tag in the three-tag sequence, that word is inserted directly. If the most probable second tag can refer to more than one word in the lexicon, a top-down word verifier determines which of the several word candidates represents the best phonetic match of the word model to the missing portion of the utterance. In either case the inserted second word is assigned an acoustic-phonetic score that is equal to the worst score allowed for words that are hypothesized in bottom-up fashion. A second penalty score is also imposed that is proportional to the duration of the putative missing word.

When all of the partial phrases have been extended as much as they can be, the best scoring phrase in the queue of complete phrases is presumed to be the best parse.

Grammar and perplexity calculations. The trigram grammar used in these evaluations was obtained by estimating statistics of sequences of approximately 450 tags of semantic categories in the 1029-word Resource Management task, which has become a common task domain for research in continuous-speech recognition within the DARPA community. The few words that were represented by more than one tag were labelled manually, with the 100 test sentences excluded from the training procedure. The perplexity of a grammar developed in this fashion is only about 17. Since we wished to evaluate the overall recognition system in the context of a less constraining grammar, we increased the perplexity of the grammar by computing a linear combination of the original tag-transition probabilities with the equal probabilities that would be observed in the absence of a grammar. The test-set perplexity of the 1029-word task that was used for these evaluations was found to be approximately 30.

WORD LATTICES USED IN EXPERIMENTS

In order to compare the effects of poor-quality word hypotheses and missing words at the parser input, we prepared four sets of word lattices lattices for a sample set of 100 sentences that are part of the Resource Management Task. The characteristics of these sets of lattices which were used in our performance calculations may be summarized as follows:

- **Original lattices** - The original data set used in these evaluations was a set of 100 sentences from the Resource Management task. 60 of these sentences were recorded at Texas Instruments and 40 more were recorded locally at CMU. The 100 sentences contained 362 different words (although the word hypothesizer could postulate any of the 1029 words in the complete task domain). These lattices were generated for experimental purposes using earlier versions of the acoustic-phonetic and word-hypothesization modules and they don't represent the current performance capabilities of the system as a whole.
- **Blind-labelled lattices:** In order to better understand how the parser performs with better quality input than what is presently provided by the acoustic-phonetic and word hypothesizers, an additional set of lattices was produced using acoustic-phonetic labels

created manually by expert spectrogram readers from spectrograms and other visual displays of the digitized waveforms. This labeling was "blind" in that the labelers did not know the identity of the correct utterance (although they were familiar with the syntax and semantics of the Resource Management task). Since these lattices nominally represent "ideal" output from the acoustic-phonetic module, they are useful for evaluating degradations in recognition performance introduced by the system's word and sentence hypothesizers. Word lattices were generated from these acoustic-phonetic labels in the fashion described in [rudnicky87].

- **Missing-word lattices** - "Missing-word lattices" were created by randomly deleting either 10 or 20 percent of the correct words from the blind-labelled lattices.

The overall quality of these lattice is summarized in Figure 1. Each curve of Figure 1 shows how many words in the lattice per correct word need be examined to ensure that a given percentage of correct words is included. For example, Figure 1 shows that the **blind-labelled lattices** contain approximately 92 percent of the correct words (if we are willing to consider a sufficiently large number of incorrect words as well), while the **missing-word lattices** with 10 and 20 percent of the correct words deleted contain only about 82 and 72 percent of the correct words, respectively. A similar curve representing the **original lattices** is shown for comparison, and its more gradual vertical rise indicates that a greater number of incorrect word candidates are hypothesized along with the correct words when actual output from the acoustic-phonetic module is used. We generally found that word recognition accuracy for these 100 sentences is somewhat worse than that obtained for the DARPA recorded database as a whole.

Figure 1: Comparison of quality of word lattices used in the sentence parsing experiments.

EXPERIMENTAL RESULTS AND DISCUSSION

Sentence parsing results were obtained by running the left-to-right and island-driven parsers over the data sets described above. To ensure comparability of the two parsers in terms of the computational resources available to them, the beam of the left-to-right parser was widened significantly. With the expanded beam width, the average run time per sentence was about 375 seconds for the left-to-right parser versus 330 seconds for the island-driven parser. The left-to-right parser parser examined about 610,000 word junctures per utterance while the island-driven parser examined about 700,000. At any given time, the left-to-right parser pursued an average of 1100 active parse candidates while the island-driven parser had about 490 parses in its queues. While it is impossible to perfectly match the computational burden incurred by the two types parsers (because of the differences in their architectures), we believe that they are reasonably comparable. Small differences may be more an artifact of the particular parser implementation than an indication of general properties of the algorithms. For these evaluations we also disabled the capability of the island-driven parser to insert words in top-down fashion since the left-to-right parser is not able to insert missing words.

Comparison of Parser Performance		
Lattice Type	Left-to-Right Parser	Island-Driven Parser
Blind-labelled	77.7, 1.9	91.4, 0.3
10-percent missing	51.4, 2.7	60.9, 1.9
20-percent missing	40.3, 2.6	46.9, 1.5

Table 1: Comparisons of the percentage of correct words detected, and incorrect words inserted, by the left-to-right and island-driven parsers for each of the four types of word lattices.

Overall Performance

Table 1 shows the recognition accuracies obtained with the two types of parsers for the four types of word lattices described in the previous section. The entries in the table consist of the ratios of the correct words detected, and the number of incorrect words inserted, divided by the number of words uttered for each sentence, expressed as a percentage. It is seen that the island-driven parser provides better detection performance than the left-to-right parser for every type of data set considered, and a better insertion percentage for all sets of data except for the original lattices.

Blind-labelled lattices		
Sentence Type	Left-to-Right Parser	Island-Driven Parser
"Good" sentences	89.0	91.0
"Bad" sentences	57.5	83.6

Table 2: Comparisons of word-detection percentages using the two parsers for "good" sentences (with no missing correct word hypotheses) and "bad' sentences (with at least one missing word candidate).

An experimental version of the left-to-right parser was constructed with a tighter and more efficient

search beam width, and it was able to recognize achieve a detection performance of 19 percent for the words in the real data set, and 76 percent using the blind-labelled lattices. The search time per utterance was reduced to 210 seconds per utterance average, 150,000 words pairs were examined for juncturing, and 425 phrase candidates were usually active in the queue. These results indicate that the left-to-right parser could have been implemented in a much more efficient fashion without sacrificing recognition accuracy.

Effects of Missing Words

In order to better understand the effects of missing words on parser performance we determined the percentage of correct words detected for two subsets of sentences from the **blind-labelled** word lattices. The first subset, which we will refer to as the "good" sentences, are those sentences for which the word hypothesizer produced all of the correct word candidates somewhere in the lattice. The word lattices of sentences in the second subset, which we call "bad" sentences, are all missing at least one of the words that was actually uttered by the speaker. The word-detection percentages obtained for these "good" and "bad" sentences from the two sets of lattices are summarized in Table 2.

It can be seen that with the high-quality blind-labelled input, the performance of the two parsers is almost equivalent when no words are missing. In fact, by comparing the parser performance for these sentences to the asymptote of the blind-labelled curve in Figure 1, we see that the first-choice sentence candidates include about all of the correct words that are in these lattices. Almost all of the difference between the detection percentages of the two parsers with blind-labelled input is attributable to the superior ability of the island-driven parser to handle input lattices with missing words.

In additional comparisons with the lower-quality word lattices we found that the overall performance is worse (as expected), and the island-driven parser obtains more correct words than the left-to-right parser, even when there are no missing words. We believe that these data reflect several factors. Even though all words are present somewhere in the lattices of the "good" sentences, the correct word candidates are more frequently hypothesized with poor scores, and they may be outscored by incorrect word candidates (with better scores). The relatively poor scores of correct word candidates can easily lead to missed words and garden-path sentences in both parsers. The island-driven parser is also handicapped by word candidates with poor scores because it requires that at least a few high-scoring correct words be in the lattices in order to form the seeds of the phrase islands.

SUMMARY

We have compared the recognition accuracy of sentence hypotheses obtained from island-driven and left-to-right sentence parsers that take a lattice of word candidates as input. We found that the island-driven parser produces better recognition accuracy in virtually every case considered, and that for high-quality input it achieves this level of accuracy by more gracefully processing input lattices when not all of the correct words of an utterance were produced by the word hypothesizer.

ACKNOWLEDGMENTS

This research was sponsored in part by the National Science Foundation, Grant IRI-85-12695, and in part by the Defense Advanced Research Projects Agency (DOD), ARPA Order No. 5167, monitored by the Air Force Avionics Laboratory under Contract N0039-85-C-0163. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the US Government.

REFERENCES

List of Figures

List of Tables