The Syntax-Oriented Speech Understanding System — SPOJUS-SYNO —

Selichi Nakagawa, Yoshihisa Ohguro and Yasuhide Hashimoto
Toyohashi University of Technology
Toyohashi, Tenpakuchou, 440, Japan

ABSTRACT
This paper describes a syntax oriented spoken Japanese understanding system named "SPOJUS-SYNO". At first, this system makes word HMMs automatically by concatenating syllable-based (trained) HMMs. Then a word lattice is hypothesized by using a word spotting algorithm and word-based HMMs. Finally, the time-synchronous left-to-right parsing algorithm is executed to find the best word sequence from the word lattice according to syntactic knowledge represented by CFG. This system was implemented in the "UNIX-QA" task with the vocabulary size of 621 words. Experimental result shows that the sentence understanding rate was about 80% for six male speakers.

1. INTRODUCTION
In speech understanding systems, there are two basic control strategies for the syntactic analyses. One is a left-to-right parsing control strategy. The other control strategy is an island-driven strategy. There are also the other basic choices for the parsing strategy. They are a backtracking search versus a parallel search. In speech understanding systems, the most optimal word sequence should be found in a word lattice because the detected words were not perfect and had scores of reliability. In such a case, the parallel search is suitable and is usually implemented in a beam search[2,3]. The search is more efficient than a best-first(A*) search[4].

Before constructing a speech understanding system, we compared the left-to-right & top-down parsing strategy with the island-driven & bottom-up strategy by using a simulated phoneme recognizer[4]. The both strategies adopted the beam search. The syntactic constraint was represented by a context-free grammar. The word lattice for an utterance was generated by a word spotting algorithm from an ambiguous phoneme sequence. The input of parsers consists of a word lattice of candidate or spotted words, which are identified by their begin and end times, and the score of the acoustic-phonetic match. Recently, Ward et al. have also studied on a similar comparison[5]. They 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. However they did not use the grammatical constraints expressed in a context-free grammar, but trigram models of sentence labels. The evaluation criterion was the rate of correct word. Our criterion was the rate of correct sentence. Therefore, our conclusion was not comparable with their results.

From simulation experiments, we found that (1)the left-to-right & top-down parsing strategy was superior to the island-driven & bottom-up strategy in terms of the processing time, (2)the recognition accuracy was almost the same for both strategies and (3)when the initial part of an utterance was noisy, the island-driven strategy became superior to the left-to-right strategy. According to those comparison results, we developed a left to right parsing oriented spoken Japanese understanding system named SPOJUS-SYNO. Many successful continuous speech recognition systems such as BYBLOS[10], SPHINX[11] and SPICOS[12] have adopted phoneme models based on HMM. The SPOJUS-SYNO also used syllable HMMs as the basic unit of speech recognition.

Fig.1 illustrates the system organization. At first, this system makes word HMMs automatically by concatenating syllable-based (trained) HMMs. Then a word lattice is hypothesized by using a word spotting algorithm and word-based HMMs. Finally, the time-synchronous left-to-right parsing algorithm is executed to find the best word sequence from the word lattice according to syntactic knowledge represented by CFG. This system was implemented in the "UNIX-QA" task with the vocabulary size of 621 words. Experimental result shows that sentence understanding rate was 80% for six male speakers in a speaker-adaptation mode.

\[\text{utterance} \rightarrow \text{Acoustic Processor} \rightarrow \text{time sequence of feature parameter} \rightarrow \text{LPC Mel Cepstrum} \rightarrow \text{automatic construction of word HMM} \rightarrow \text{Word Spotter} \rightarrow \text{Word Lattice} \rightarrow \text{Syntactic Analysis} \rightarrow \text{Syntactic Knowledge (Context Free Grammar)} \rightarrow \text{Recognized Sentence}\]

Fig.1 System Organization of SPOJUS-SYNO

2. SYLLABLE HIDDEN MARKOV MODEL
In the Japanese syllables consist of about 110 syllables, each of which is composed of a consonant and a vowel (CV), a syllabic nasal(N), a vowel (V), or a consonant, a semivowel and a vowel (CYV). We adopted a continuous output probability HMM with a discrete duration probability as shown in Fig.2. In this model, the output between state q1 and state q3 corresponds to a part of consonant and the output between state q3 and state q5 corresponds to a part of vowel. To estimate the initial parameters, the training data were divided into four parts at the ratio of 30-20, each of which corresponds to outputs of a state transition between q1 and q2, q2 and q3, q3 and q4, or q4 and q5 respectively. The four parameter sets of the mean vector and covariance matrix of feature vectors were calculated from these divided parts for each syllable. The initial distribution of duration was set to a uniform distribution.

The Baum-Welch reestimation algorithm was iterated.
by ten times for each HMM. We assumed that the output was observed from a multi-variate normal distribution.

After each iteration, the estimated distribution of duration is smoothed. In speaker-adaptation mode, only the mean vectors of multivariate distributions were adapted by using training samples.

was observed from a multi-variate normal distribution. This grammar is represented by the context-free grammar as shown in Figure 3. The variable with the suffix "O" shows that it is a nonterminal symbol, and the variable with "s" a word class (a kind of nonterminal symbols). The word class means the set of words with the same syntactical category. The numbers of row and column define the position of a production rule. They will be used for the parsing algorithm described in the next section.

4. SENTENCE RECOGNITION ALGORITHM from WORD LATTICE [7,8]
4.1 Representation of Grammar

The syntactic knowledge in terms of the "task" is given by the grammar. This grammar is represented by the context-free grammar as shown in Figure 3. The variable with the suffix "O" shows that it is a nonterminal symbol, and the variable with "s" a word class (a kind of nonterminal symbols). The word class means the set of words with the same syntactical category. The numbers of row and column define the position of a production rule. They will be used for the parsing algorithm described in the next section.

4.2 Time-Synchronous Context-Free Parsing Algorithm
4.2.1 Parser

In the LITHAN speech understanding system[2], we prepared the different

\[ P_n(t, N) = \text{Pre-set score depends on the length of word.} \]

We prepared the different scores for every different number of syllables included in words.

The spotting score is transformed into the lattice score in the following:

Thus the maximum score of lattice score becomes 1000. By using this lattice score, the system prunes candidate words if the lattice score is less than the pre-set score, the word hypothesis is rejected. The pre-set score depends on the length of word. We prepared the different score for every different number of syllables included in words.

4. SENTENCE RECOGNITION ALGORITHM from WORD LATTICE [7,8]
4.1 Representation of Grammar

The syntactic knowledge in terms of the "task" is given by the grammar. This grammar is represented by the context-free grammar as shown in Figure 3. The variable with the suffix "O" shows that it is a nonterminal symbol, and the variable with "s" a word class (a kind of nonterminal symbols). The word class means the set of words with the same syntactical category. The numbers of row and column define the position of a production rule. They will be used for the parsing algorithm described in the next section.

4.2 Time-Synchronous Context-Free Parsing Algorithm
4.2.1 Parser

In the LITHAN speech understanding system[2], we prepared the different scores for every different number of syllables included in words.
proposed an efficient context-free parsing algorithm which is similar to Earley algorithm[5]. We also use this algorithm in this paper.

Let the position of a production rule represent by the number. For example, the number "8" denotes "S", and "9" "NP" and "19" "PP" in the above grammar. Fig 3. The basic problem is formalized as follows: Which words are predicted as the succeeding words when a partial sentence is given? For example, when the partial sentence "MARY WILL PLAY" is given, which words could be added in the right hand side? In this case, the partial sentence is derived by the following production rules: "S" → "NP" + "VP" → "NP" + "VP" + "NP" + "PP" or "S" → "NP" "VP" + "NP" + ... . Therefore the succeeding words could be predicted from "NP" and "VP". Of course, "MARY WILL PLAY" may be regarded as a complete sentence in the first alternative derivation. By", "OF", "WITH", "THE", "A", "BIG", "YOUNG", "JOHN", "MARY", "MAIN", "I", "TENNIS" and "GAME" are predicted.

We can memorize the application order of the production rules by the sequence of positions in the grammar. For the above example, "MARY WILL PLAY" is derived by the sequence "8" + "17" + "41" + "17" → "17 41 56" → "17 41 57 104" → "17 41 57 105" → prediction of "NOUN" → "18" → prediction of "AUX" → "19" → "19 64", "19 72" or "19 80" → "19 65", "19 73" or "19 81" → prediction of "VERB". For convenience sake, we call this sequence "grammar path". The recursive algorithm for parsing or prediction is given below:

<PARSER>
1. Enter the given grammar path into "path list".
2. If the path list is empty, stop. Otherwise, select a grammar path from the path list. Add the number of the most right hand side in the grammar path by 1. This number indicates the next processed position in the grammar.
3. If the variable of the position is a terminal symbol, predict the word (terminal symbol) and generate the grammar path. Then go to step 2.
4. If the variable of the position is a word class with the affix "", predict the set of words for the word class and generate the grammar path. Then go to step 2.
5. If the variable of the position is a nonterminal with the affix "", the production rules with the same nonterminal at the left hand side are predicted. That is, the head positions of these rules are concatenated at the most right hand side of the grammar path. Enter these paths into the path list. Then go to step 2.
6. If the variable of the position is empty, eliminate the number of the most right hand side in the grammar path and enter this path into the path list. Then go to step 2.

In this procedure, we should pay attention to the representation of left recursion in a production rule. For example, the production rule "S → NP + VP + NP + PP" is not used at all because the recursion of "NP" will make the algorithm time-synchronous in terms of the ending frame of generated partial sentences. This algorithm executes the prediction of words at the right hand side of a partial sentence and the concatenation of a spotted word (candidate word) at the same time. If all possible partial sentences are taken into consideration, the computation time or memory spaces will become large. Therefore we select a few best partial sentences and the other abandon. We proposed this pruning technique in the LHAR understanding system[2]. In general, this search technique is well known as "beam search"[3].

The number of partial sentences which covers the same range of the utterance is restricted less than a pre-set value. That is, the width or radius of syllable search. The sentence with the highest score which covers the whole of the utterance is decided as the recognition result.

This forward algorithm S is described in brief below. [1] At the initial step, a partial sentence is set to "empty", and i=1.

[2] The algorithm predicts words for a partial sentence, which covers from the head to the i-th position of input sentence.

[3] The algorithm expands the partial sentence (concatenation of partial sentence and predicted word which is found in the word lattice) and sorts the expanded partial sentences in terms of corresponding scores.

[4] If "i" is the last position of the input sentence, the best word sentence is regarded as the recognition result. Otherwise, i → i+1 and go to the step [2].

5. EXPERIMENTAL RESULTS

5.1 Speech Materials and Feature Parameters

Six male speakers uttered 216 words, 80 foreign words and 50 sentences in a soundproof room, respectively. These words were segmented into syllable units by the inspection and used for training syllable-based HMMs. Fifty sentences for test data were related to the content of "Question or Demand for Electric Mail", which was a part of the task of UNIX-QA. The speed of utterances ranged from 8 to 9 moras per second (about 16 to 18 phonemes per second). It was moderately fast. These utterances were sampled/digitized with the accuracy of 12 bits / sampling by 12 KHz and analyzed by the 14 order LPC. We obtained 30 LPC cepstrum coefficients and signal power for every 5ms. These coefficients were transformed to 10 LPC mel-cepstrum coefficients.

The vocabulary size of a part of the task is 521 words. The related sentences are generated by the context free grammar which is represented by rewriting rules of number of 354, nonterminal symbols of 259, word classes (a kind of nonterminal symbols) of 268, and 600 direct rewriting rules from word classes to terminal symbols. The average branching factor is about 26 (static) or 14 (dynamic). The perplexity is about 10^16. The number of plausible sentences in this subtask is about 10^16[16].

5.2 Word Spotting Results

Table 1 shows the evaluation results of the word spotting performance. The rate in the column of the n-th rank shows the percent accuracy by which an input word is correctly identified as one among the best n words in the neighborhood. The number of missing words denotes the number of unspotted input words in the total of 48for speaker SN, 47 (for speaker TL), 49for speaker HU), 46(for speaker KO), 46for speaker MA) and 46for speaker SE) sentences. In comparison with the pure speech recognizer (see Table 2), our syllable HMM-based word spotting in a multi-speaker mode is slightly worse than the word spotting of simulator in the case 80% of phoneme recognition rate and it is comparable with one in a speaker-adaptation mode.
Independent Sentence

postposition" shown in Table 4. In deed, we obtained the sentence results in the case of postpositions in the recognition unit of such as are comparable with or better than the simulation semantically similar postpositions such as and misrecognition was understandable. These accuracies in a speaker-adaptation mode, respectively.

Table 4 Sentence recognition results by simulation (phoneme recognition rate = 80%)

<table>
<thead>
<tr>
<th>rank</th>
<th>sentence recognition rate</th>
<th>predicted word</th>
<th>duration</th>
<th>parsing time</th>
</tr>
</thead>
<tbody>
<tr>
<td>SN</td>
<td>60.0 %</td>
<td>86625</td>
<td>320 sec</td>
<td></td>
</tr>
<tr>
<td>TI</td>
<td>74.5 %</td>
<td>53711</td>
<td>2197 sec</td>
<td></td>
</tr>
<tr>
<td>HU</td>
<td>60.0 %</td>
<td>85985</td>
<td>2197 sec</td>
<td></td>
</tr>
<tr>
<td>KO</td>
<td>60.0 %</td>
<td>54966</td>
<td>1954 sec</td>
<td></td>
</tr>
<tr>
<td>WA</td>
<td>65.7 %</td>
<td>56522</td>
<td>2056 sec</td>
<td></td>
</tr>
<tr>
<td>SE</td>
<td>58.5 %</td>
<td>62121</td>
<td>2312 sec</td>
<td></td>
</tr>
</tbody>
</table>

6. CONCLUSION

In this paper, we described the syntax-oriented spoken Japanese understanding system (SPOJUS-SYNO), which was based on syllable HMMs and the time synchronous parsing algorithm of a context-free grammar. We obtained the accuracy about 80% for sentence recognition of six male speakers. Some mis-recognition was caused from the confusion between postpositions or acoustically similar words. We are now improving the system in the following points: syllable duration, estimation of output probability distribution for every frame (similar to a stochastic dynamic time warping[14,15]), mixture distributions, dynamic feature & multi-distribution.

REFERENCES