Title	Speech Recognition for Japanese Spoken Language
Author	SADAOKI FURUI
Journal/Book name	ISSIPNN '94, Vol. 1, No. , pp. 122-126
発行日 / Issue date	1994, 4
権利情報 / Copyright	(c)1994 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

1994 International Symposium on Speech, Image Processing and Neural Networks, 13-16 April 1994, Hong Kong

SPEECH RECOGNITION FOR JAPANESE SPOKEN LANGUAGE

Sadaoki Furui

NTT Human Interface Laboratories
3-9-11 Midori-cho, Musashino-shi, Tokyo, 180 Japan

ABSTRACT

This paper overviews the speech recognition issues for the Japanese language, and introduces three recent research projects conducted at NTT (Nippon Telegraph and Telephone) Human Interface Laboratories and ATR Interpreting Telephony Research Laboratories. The first topic is stochastic language models for sequences of Japanese characters to be used in a Japanese dictation system with unlimited vocabulary. The second topic is an accurate and efficient algorithm for very-large-vocabulary continuous speech recognition based on an HMM-LR algorithm. This algorithm was applied to a telephone directory assistance system that recognizes spontaneous speech having the vocabulary size of roughly 80,000. The third topic is a continuous speech recognition system based on strategies of the phoneme-context-dependent (allophonic) modeling and parsing.

1. OVERVIEW OF SPEECH RECOGNITION ISSUES FOR JAPANESE

One of the special features of the Japanese language is that written sentences usually include both Kana (Japanese alphabet) and Kanji (Chinese characters). Kana, consisting of Hira-gana and Kata-kana, are the minimal linguistic units in the written form and correspond to Japanese syllables. These syllables are made up of a consonant-vowel pair or a single vowel. Kanji are linguistic units having one or more meanings and readings; the readings, which depend on contexts, can be written as Kana sequences. Japanese words are made up of sequences of Kana and Kanji.

In English, word sequence probability is usually used to make a language model. However in Japanese, since words are not clearly delimited, Kana sequence probability has usually been effectively used for speech recognition. Recently, a Japanese dictation system using a character (Kana and Kanji) source model instead of a Kana source model has been built for the following reasons [1][2]. (i) For a given length of character source, a character source model can effectively deal with a longer phoneme context. (ii) A character source model can directly convert speech into Kana and Kanji sequences, without post-processing of Kana-to-Kanji conversion. The technical details will be described in Section 2.

The Japanese language also has following charactersitics.

(1) The number of phonemes, especially the number of vowels, is relatively small; therefore, the number of homonyms is large. For this reason, it is convenient to use character displays as output devices and choose the right words or sentences from displayed candidates.
(2) Verbs and adjectives have inflections.
(3) Loan words can easily be created as sequences of Katakana.
(4) Meanings of homonyms are distinguished by tones.
(5) Order of phrase sequences has high freedom.
(6) Affirmative or negative sentences are distinguished at the end of sentence.
(7) Difference between spoken and written languages is relatively large.
(8) Linguistic structure is largely different from most western languages.
(9) Difference between dialects is large.
(10) Word pronunciation and accent are changed by word concatenation.
(11) The Mora, rather than the syllable, is the timing unit.

These charactersitics are tightly related to the difficulties in recognizing spoken Japanese. Three recent topics in Japanese speech recognition research coping with some of these problems will be introduced in the following sections.

2. CHARACTER SOURCE MODELING FOR A JAPANESE DICTATION SYSTEM

2.1 Character Source Modeling

This section introduces a Japanese dictation system using a character source model developed at NTT. A character trigram probability was calculated using a text database to construct a character source model. Since ordinary Japanese texts may use any of several thousand different characters, the trigrams obtained using practical databases are very sparse. This problem was alleviated by applying the deleted interpolation algorithm. That is, an improved trigram was estimated by linear combination of a zerogram, unigram, bigram, and trigram.

Test-set perplexities and the number of different characters for three different tasks are listed in Table 1. The task of

Table 1 - Test-set Kana-based perplexity and number of different characters for text database

Text database for training	Kana-based perplexity		Number of different characters	
	Kana models	Character models	Kana models	Character models
Conference registration	10.5	9.7	117	1362
Travel arrangement	18.6	31.3	114	1480
Both	9.6	10.1	120	1696

0-7803-1865-X/94/$3.00 © 1994 IEEE

the recognition test data was conference registration. When the tasks of training and test data are the same, the Kana-based perplexities of character source models are smaller than those of Kana source models. The results shown in the table indicate that a character source model is efficient for the Japanese dictation system, and that the source model is highly dependent on the task.

2.2 Japanese Dictation System

Figure 1 is a schematic diagram of the dictation system. This system processes phrase-by-phrase input speech using the HMM-LR algorithm [3][4]. The HMM-LR algorithm uses a generalized predictive LR parser [5] as a language model and hidden Markov models (HMMs) as phoneme models. The LR parser predicts phonemes of the input speech successively from left to right (from the beginning to the end) according to context-free rewriting rules, and sends them to the HMM phoneme verifier. The phoneme verifier calculates the likelihood of each predicted phoneme for the input speech, and returns the score to the LR parser. In the reduce action of the LR parser, a phoneme sequence is converted into a character, based on the weighted sum of the HMM likelihood and the trigram likelihood.

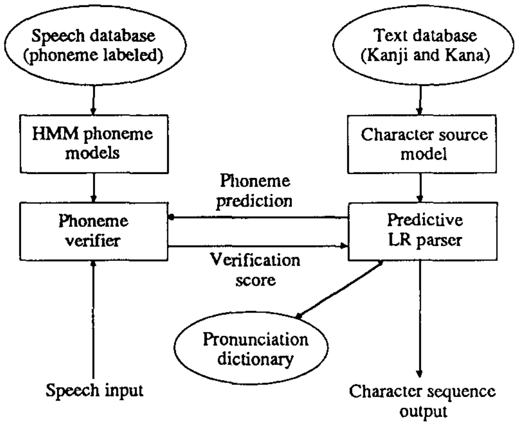

Fig. 1 - Schematic diagram of Japanese dictation system.

Each Kanji character may have several readings, depending on the context. The character trigram, however, is calculated from only the character sequences in the training text database, neglecting the reading of the Kanji, and context-independent rewriting rules for a Kanji-to-phoneme sequence are given to make an LR table. Therefore, the parser produces many contextually wrong candidates. To solve this problem, the step of consulting a dictionary to check the phoneme sequence of the candidate was added, and the candidates whose phoneme sequences were inappropriate to the character sequence were eliminated. This reduced the test-set Kana-based perplexities for the character source models by roughly 20%.

2.3 Experimental Results

Speaker-dependent transcription experiments were performed. HMM phoneme models were made from 5,240 Japanese words and 216 phonetically balanced words spoken by a male speaker. The character source model was obtained from the text database of the conference registration task. Test data consisted of 274 phrases uttered by the same speaker.

The transcription rate, that is, the rate that the output phrase candidate whose character sequence and pronunciation were both correct were included in the top four choices, was increased from 70.8% to 74.5% by the pronunciation check. These results indicate that the proposed method of pruning based on the character sequence pronunciation is effective in eliminating candidates whose readings do not fit the context.

2.4 Discussion

Another method using a pronunciation-tagged character source model has also been tried to further reduce erroneous outputs that have inappropriate readings of Kanji [2]. In this method, the source model was made from a text database that consisted of characters with their pronunciation tags, and the inappropriate candidates were statistically eliminated without using the dictionary. Since each Kanji was divided into several characters according to the pronunciation tags, the number of different characters increased to 2,500. However, the test-set perplexity of the tagged character source model was 7.6, which is almost the same as that of the non-tagged character source model with dictionary.

Other topics being studied but not mentioned here include task adaptation in source models [6]. Since the source model is highly dependent on the task, and it is not always possible to obtain a very large text database for each new task, it is very important to establish a method of adapting the model to the new task using a small amount of text.

3. LARGE-VOCABULARY CONTINUOUS SPEECH RECOGNITION ALGORITHM AND ITS APPLICATION TO TELEPHONE DIRECTORY ASSISTANCE

3.1 Speech Recognition Algorithm

This section introduces an NTT's accurate and efficient algorithm for very-large-vocabulary continuous speech recognition based on the HMM-LR algorithm [7][8]. Algorithms for recognizing large-vocabulary continuous speech require (i) accurate scoring for phoneme sequences, (ii) reduction of trellis calculation, and (iii) efficient pruning of phoneme sequence candidates. For these requirements, we proposed the following four methods.

(1) Two-Stage LR Parser

This two-stage LR parser uses two classes of LR tables: a main grammar table and sub-grammar tables (Fig. 2). These grammar tables are separately compiled from a context-free grammar. The sub-grammar tables deal with semantically classified items, such as city names, area names, block numbers, and subscriber names. The main grammar table controls the relationships between these semantic items. Dividing the grammar into two classes has two advantages; since each grammar can be compiled separately, the time needed for compiling the LR table is reduced, and the system can easily be adapted to many types of utterances by changing the main grammar rules.

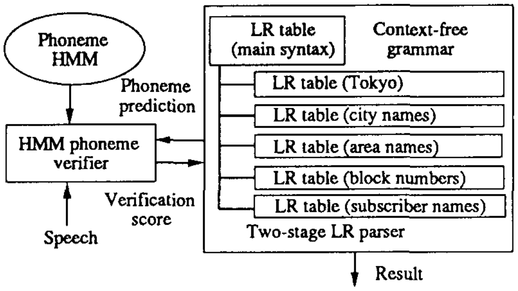

Fig. 2 - Structure of the continuous speech recognition system.

(2) Accurate Scoring

The algorithm uses a backward trellis as well as a forward trellis so as to accurately calculate the score of a phoneme sequence candidate. The backward trellis likelihood is calculated without any grammatical constraints on the phoneme sequences; it is used as a likelihood estimate for potential succeeding phoneme sequences.

(3) Adjusting Window

An algorithm for determining an adjusting window that restricts calculation to a probable part of the trellis for each predicted phoneme has been proposed (Fig. 3). The adjusting window (shaded rectangle in Fig. 3) has a length of 50 frames (400 ms). The score within the adjusting window is calculated by taking the convolution of the forward and backward trellises. In this procedure, the likelihood in the backward trellis is multiplied by (1-ε), where ε is a small positive value.

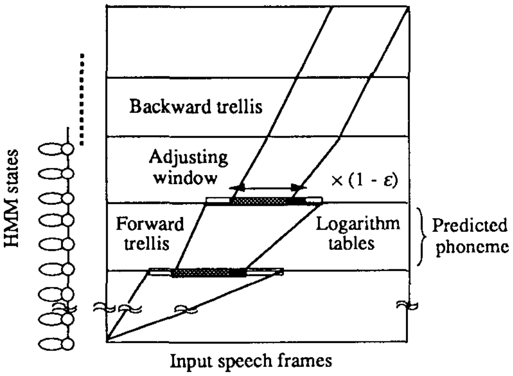

Fig. 3 - Search algorithm for the continuous speech recognition system.

(4) Merging Candidates

The LR tables need multiple pronunciation rules to cover allophonic phonemes, such as devoicing and long vowels in Japanese pronunciation. These multiple rules cause an explosion of the search space. To reduce the search space, phoneme sequence candidates as well as grammatical states are merged when they are phonetically and semantically the same. Candidate word sequences having the same meaning are further merged, ignoring the differences in non-keywords.

3.2 Recognition Experiments

This algorithm is applied to a telephone directory assistance system that recognizes spontaneous speech that contains addresses and names of more than 70,000 subscribers (about 80,000 words). The grammar used in this system has various rules for interjections, verb phrases, post-positional particles, etc. It was made by analyzing 300 sentences in simulated telephone directory assistance dialogs. The word perplexity was about 70,000. In this task, no constraints by the directory database were placed on the combination of addresses and subscriber names.

Two speaker-independent HMM types were prepared to evaluate the algorithm: 56 context-independent phoneme HMMs, and 358 context-dependent phoneme HMMs. The proposed algorithm was evaluated on the basis of 51 sentences that included 184 keywords. These utterances were prepared as text with various interjections and verb phrases. They were "spontaneously" uttered by eight different speakers. Experimental results confirmed the effectiveness of merging at the meaning level and context-dependent HMMs. These techniques achieved an average sentence understanding rate of 65% and an average keyword recognition rate of 89%. The results show that the system performs well in spite of the large perplexity.

3.3 Discussion

Based on this algorithm, a multi-modal speech dialog system for telephone directory assistance with three input devices (microphone, keyboard, and mouse) and two output devices (speaker and display) has been constructed. This system incorporates an HMM composition technique [9] to cope with the problems of noisy speech. The system is now being evaluated from the human-machine-interface point of view.

Other related activities not mentioned in this section include a proposal of phoneme HMMs constrained by frame correlations [10], which effectively utilize statistical transition information of spectra.

4. SSS-LR CONTINUOUS SPEECH RECOGNITION SYSTEM

4.1 System Structure

This section introduces ATR's continuous speech recognition system called "SSS-LR", which is based on strategies of the phoneme-context-dependent (allophonic) modeling and parsing [11][12]. In allophonic modeling, it is very important to attain a precise and robust model-set from limited training samples. To solve this problem, ATR proposed the Successive State Splitting (SSS) algorithm that automatically generates an efficient representation of allophonic continuous density HMMs, which is called a Hidden Markov Network (HMnet). The SSS principle has also been applied to duration clustering: optimal clusters of phoneme-context-dependent durations are automatically generated independently of the HMnet-based allophonic classes.

In order to handle the allophonic HMMs, phoneme-context-dependent LR parsing algorithms [13] based on a generalized LR parser has also been proposed. The phoneme-context-dependent LR parser dynamically predicts the current phoneme context using a phoneme-context-independent LR table. It drives precise allophonic HMMs and duration models by exploiting phoneme context dependency both inside words and at the word boundaries.

The LR parser predicts a phoneme-triplet hypothesis referring to the LR table. In the allophone verifying module, the sequence of HMM states in the HMnet that corresponds to the phoneme-triplet is selected, and likelihood is calculated. The parser then checks phoneme durations using phoneme-context-dependent duration models.

4.2 Phoneme-Context-Dependent HMMs and Successive State Splitting Algorithm

(1) The Successive State Splitting Algorithm (SSS)

By using the maximum likelihood criterion, the SSS can simultaneously and automatically optimize the following three items that are important for constructing phoneme-context-dependent HMMs: (i) the model unit (i.e., the set of phoneme context classes); (ii) the model architecture (i.e., the number of states per model and the architecture of state sharing); and (iii) the model parameters (i.e., output probability density distributions and state transition probabilities).

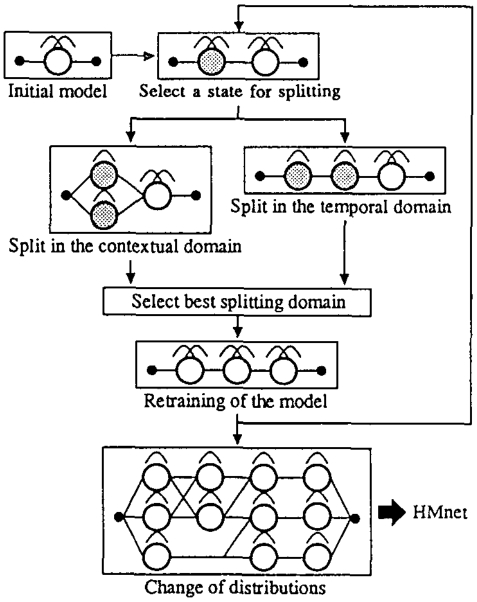
Initial model
Select a state for splitting
Split in the contextual domain
Split in the temporal domain
Select best splitting domain
Retraining of the model
HMnet
Change of distributions

Fig. 4 - The Successive State Splitting (SSS) algorithm.

Figure 4 illustrates the SSS algorithm. The concept of the SSS is to successively create each model by iterating the split of a probabilistic statistical signal source (i.e., a hidden Markov state) into either a phoneme contextual domain or a temporal domain. However, to achieve this concept directly, it is necessary to evaluate all possible combinations and determine in which state and in which domain a split can produce the maximum likelihood.

Since an exhaustive search would require a huge amount of computation, the following two approximations were introduced in the actual algorithm: (i) at each iteration, the state having the largest output probability density distribution is selected to be split; (ii) the output probability density distribution of each state is formed as two-mixture Gaussian density distribution, and when a state is split, each of the two Gaussian density distributions of the original state is assigned to one of the two new states. These approximations greatly reduced the amount of computation.

(2) The Hidden Markov Network (HMnet)

The HMnet is a network of multiple hidden Markov states. Each state has the following information: (a) state index; (b) acceptable contextual class; (c) lists of preceding states and succeeding states; (d) parameters of the output probability density distribution; and (e) state transition probabilities. In the HMnet, the model corresponding to each phoneme context is made by concatenating several states, each of which is chosen by applying the restrictions of the preceding state list and the succeeding state list. Since this model is equivalent to a common HMM, the forward-pass algorithm can be used to calculate the likelihoods for input samples, similarly as for common HMMs.

(3) Phoneme Duration Modeling

The SSS algorithm is also applied to a set of scalar values, phoneme duration training data that indicates the duration of phonemes. Phoneme duration models, each of which has one state and a single Gaussian density distribution, are constructed by this algorithm.

4.3 Phoneme-Context-Dependent Parsing Algorithm

A modified LR parser predicts the phoneme context dynamically, using a phoneme-context-independent LR table. This is done by searching succeeding phonemes at the next state and repeating this process until parsing is complete. After the LR parser hypothesizes a phoneme triplet, it evaluates the likelihood for the current phoneme with an allophonic model that corresponds to the phoneme context.

4.4 Experiments

Continuous speech recognition experiments were carried out using ATR's conference registration task with a 1,000-word vocabulary. In these experiments, a diagonal-covariance single Gaussian distribution was used as the output probability density distribution of each state. The grammar included 1,407 rules, the task entropy was 17.0, and the phoneme perplexity was 5.9 [14]. About 1,700 phoneme-context-depenent models were represented by a 600-state HMnet. The beam-search technique was used. The speaker was a professional announcer.

Experimental results show that fast parsing with high accuracy can be realized by the SSS-LR. Because of the high accuracy, the beam width can be smaller than with other conventional methods.

4.5 Discussion

The SSS-LR achieved good performance in the 1,000-word recognition experiments, probably due to the high accuracy of the HMnet-based representation. Speaker-independent experiments based on the speaker-mixture HMnet [15] have also been tried. Each mixture component is derived from a particular speaker, and training speech data are used to determine the speaker-mixing weights.

The SSS-LR played an important role in "ATREUS", the final speech recognition system of ATR's seven year interpreting telephone project [16]. ATREUS was implemented on the speech translation system "ASURA", which translates Japanese into both English and German [17].

5. SUMMARY

This paper has overviewed the speech recognition issues for the Japanese language, and introduced three recent topics researched in Japan. Some of the algorithms described in this paper are Japanese-language specific, whereas some of them can be applied to other Asian languages or even almost universally. The development of multi-language speech recognition algorithms and systems is expected to become more important in the near future. This can be accomplished by separating speech recognition algorithms into two parts; language independent and language specific parts. To achieve quick technological progress, it is important to increase the proportion of the language-independent part and to have international collaboration in the development of these algorithms.

REFERENCES

[1] T. Yamada, et al: "Phonetic typewriter based on phoneme source modeling", Proc. ICASSP 91, Toronto, pp.169-172 (1991)

[2] T. Yamada, et al.: "Japanese dictation system using character source modeling", Proc. ICASSP 92, San Francisco, pp.I-37-40 (1992)

[3] K. Kita, et al.: "HMM continuous speech recognition using predictive LR parsing, " ICASSP 89, Glasgow, pp. 703-706 (1989).

[4] T. Hanazawa, et al.: "ATR HMM-LR continuous speech recognition system", Proc. ICASSP 90, Albuquerque, pp.53-56 (1990)

[5] M. Tomita: "Efficient parsing for natural language: A fast algorithm for practical systems, " Kluwer Academic Publishers (1986).

[6] S. Matsunaga, et al.: "Language model adaptation for continuous speech recognition", 1991 IEEE-SPS Arden House Workshop on Speech Recognition, 8.2 (1991)

[7] Y. Minami, et al.: "Very-large-vocabulary continuous speech recognition algorithm for telephone directory assistance", Proc. Eurospeech '93, Berlin, pp.2129-2132 (1993)

[8] Y. Minami, et al.: "Large-vocabulary continuous speech recognition system for telephone directory assistance", Proc. Int. Symposium on Spoken Dialogue, Tokyo, pp.169-172 (1993)

[9] F. Martin, et al.: "Recognition of noisy speech by composition of hidden Markov models", Proc. Eurospeech '93, Berlin, pp.1031-1034 (1993)

[10] S. Takahashi, et al.: "Phoneme HMMs constrained by frame correlations", Proc. ICASSP 93, Minneapolis, pp.II-219-222 (1993)

[11] J. Takami and S. Sagayama: "A successive state splitting algorithm for efficient allophone modeling", Proc. ICASSP 92, San Francisco, pp.I-573-576 (1992)

[12] A. Nagai, et al.: "The SSS-LR continuous speech recognition system: Integrating SSS-derived allophone models and a phoneme-context dependent LR parser", Proc. ICSLP 92, Banff, pp.1511-1514 (1992)

[13] A. Nagai, et al.: "Phoneme-context-dependent LR parsing algorithms for HMM-based continuous speech recognition, " Eurospeech '91, Genova, pp. 1397-1400 (1991).

[14] T. Kawabata, et al.: "Task entropy and phone perplexity", Proc. Acoust. Soc. Japan Spring Meeting, 3-6-12 (1989) (In Japanese).

[15] T. Kosaka, et al.: "Rapid speaker adaptation using speaker-mixture allophone models applied to speaker-independent speech recognition, " Proc. ICASSP 93, Minneapolis, pp.II-570-573 (1993).

[16] S. Sagayama, et al.: "ATREUS: a speech recognition front-end for a speech translation system", Proc. Eurospeech '93, Berlin, pp.1287-1290 (1993)

[17] T. Morimoto, et al.: "ATR's speech translation system: ASURA", Proc. Eurospeech '93, Berlin, pp.1291-1294 (1993)