A New Approach to Automatic Speech Summarization

Chiori Hori, *Member, IEEE*, and Sadaoki Furui, *Fellow, IEEE*

***Abstract*—This paper proposes a new automatic speech summarization method. In this method, a set of words maximizing a summarization score is extracted from automatically transcribed speech. This extraction is performed according to a target compression ratio using a dynamic programming (DP) technique. The extracted set of words is then connected to build a summarization sentence. The summarization score consists of a word significance measure, a confidence measure, linguistic likelihood, and a word concatenation probability. The word concatenation score is determined by a dependency structure in the original speech given by stochastic dependency context free grammar (SDCFG). Japanese broadcast news speech transcribed using a large-vocabulary continuous-speech recognition (LVCSR) system is summarized using our proposed method and compared with manual summarization by human subjects. The manual summarization results are combined to build a word network. This word network is used to calculate the word accuracy of each automatic summarization result using the most similar word string in the network. Experimental results show that the proposed method effectively extracts relatively important information by removing redundant and irrelevant information.**

***Index Terms*—Dynamic programming, objective evaluation, speech summarization, summarization scores.**

I. Introduction

RECENTLY, large-vocabulary continuous-speech recognition (LVCSR) technology has made significant advancement. Real time systems can now achieve word accuracy of 90% and above for speech dictated from newspapers. Currently various applications of LVCSR systems, such as automatic closed captioning [1], meeting/conference summarization [2] and indexing for information retrieval [3], are actively investigated. Transcribed speech usually includes not only redundant information such as disfluencies, filled pauses, repetitions, repairs, and word fragments, but also irrelevant information caused by recognition errors. Therefore, practical applications using LVCSR systems require a process of speech summarization which removes redundant and irrelevant information and extracts relatively important information depending on users' requirements, especially for spontaneous speech.

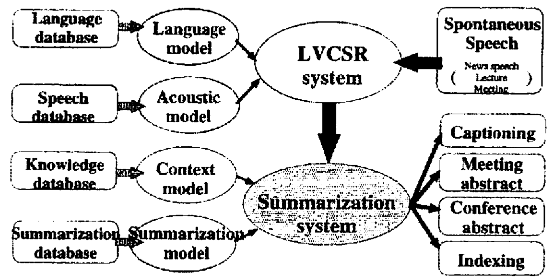

Fig. 1. Automatic speech summarization system.

Techniques of automatically summarizing written text have been actively investigated in the field of natural language processing [4]. One of the major techniques for summarizing written text is the process of extracting important sentences. A major difference between text summarization and speech summarization exists in the fact that transcribed speech is sometimes linguistically incorrect due to the spontaneity of speech and recognition errors. A new approach is needed to automatically summarizing speech to cope with such problems.

Our goal is to build a system that extracts and presents information from spoken utterances according to users' desired amount of information. Fig. 1 shows our proposed system. The output of the system can be either a simple set of keywords, a summarized sentence for each utterance, or summarization of an article consisting of multiple utterances. These outputs can be used for indexing, making closed captions and abstracts, etc. In the closed captioning of broadcast news, the number of words spoken by professional announcers sometimes exceeds the number of words that people can read and understand if all of them are presented on the TV screen. Therefore, reduction of the number of words in speech is indispensable. Meeting/conference summarization should be useful if it can extract relatively important information scattering about in the original speech.

In this paper, we first propose a new method of automatically summarizing each utterance. In this method, relatively important words are extracted removing redundant and irrelevant words according to a target compression ratio. The summarization method focuses on topic word extraction, weighting linguistically and semantically correct word concatenation [5], [6], and acoustically as well as linguistically reliable parts of speech recognition results [7]. All of these features are represented as probabilistic scores. Summarization results obtained by this method simultaneously maintain topic words and keep a syntactic structure by properly weighting the scores.

We then extend this method to summarization of a set of multiple utterances (sentences) having consistent meanings. This is

Manuscript received August 31, 2001; revised June 17, 2002. The associate editor coordinating the review of this paper was Dr. Sankar Basu.

C. Hori is with the Department of Computer Science, Graduate School of Information Science and Engineering, Tokyo Institute of Technology, Tokyo 152-8552, Japan and also with the NTT Communication Science Laboratories, Kyoto 619-0237, Japan (e-mail: chiori@furui.cs.titech.ac.jp; chiori@cslab.kecl.ntt.co.jp).

S. Furui is with the Department of Computer Science, Graduate School of Information Science and Engineering, Tokyo Institute of Technology, Tokyo 152-8552, Japan (e-mail: furui@furui.cs.titech.ac.jp).

Digital Object Identifier 10.1109/TMM.2003.813274

done by adding a rule which restricts application of the score beyond the sentence boundaries. As a result, original sentences including many informative words are preserved and those including less informative words are deleted or shortened. This summarization technique can be considered as a combination of the summarization method extracting important sentences investigated in the field of natural language processing and the sentence-by-sentence summarization method. The multiple utterance summarization method should be especially useful for making lecture abstracts, meeting minutes, etc.

II. Summarization of Each Sentence Utterance

Our proposed method to summarize speech, sentence by sentence, extracts a set of words maximizing a summarization score from an automatically transcribed sentence according to a target compression ratio. The summarization score indicates goodness of a summarized sentence, and it consists of a word significance score I as well as a confidence score C of each word in the original sentence, a linguistic score L of the word string in the summarized sentence [5], [7], and a word concatenation score T. The word concatenation score indicates a word concatenation probability determined by a dependency structure in the original sentence given by SDCFG [6]. The total score is maximized using a dynamic programming (DP) technique [5], [7]. This method is effective in reducing the number of words by removing redundant and irrelevant information without losing relatively important information.

Given a transcription result consisting of N words, $W = w_1, w_2, \ldots, w_N$, the summarization is performed by extracting a set of M $(M < N)$ words, $V = v_1, v_2, \ldots, v_M$, which maximizes the summarization score given by

$$S(V) = \sum_{m=1}^{M} \{L(v_m \mid \cdots v_{m-1}) + \lambda_I I(v_m) + \lambda_C C(v_m) + \lambda_T T(v_{m-1}, v_m)\} \quad (1)$$

where λ_I, λ_C, and λ_T are weighting factors for balancing among I, L, C, and T.

A. Word Significance Score

The word significance score I indicates relative significance of each word in a original sentence [5]. The amount of information based on the frequency of each word given by (2) is used as the word significance score for each noun:

$$I(w_i) = f_i \log \frac{F_A}{F_i} \quad (2)$$

where

- w_i a noun in the transcribed speech;
- f_i number of occurrences of w_i in the transcribed article;
- F_i number of occurrences of w_i in all the training news articles;
- F_A summation of all F_i in all the training news articles $(=\sum_i F_i)$

w_i which occurs homogeneously among documents in the collection data is deweighted by the tf-idf. On the other hand, w_i which occurs frequently over all documents is deweighted by our measure given by (2).

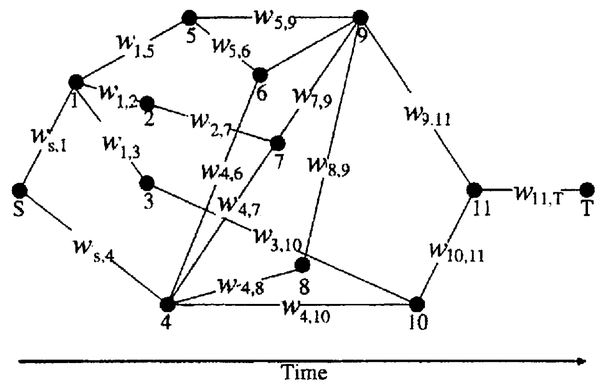

Fig. 2. Example of word graph.

A flat score is given to words other than nouns. To reduce the repetition of words in the summarized sentence, a flat score is also given to each reappearing noun.

B. Linguistic Score

The linguistic score $L(v_m \mid \cdots v_{m-1})$ indicates goodness of word strings in a summarized sentence, and is measured by a trigram probability $P(v_m \mid v_{m-2} v_{m-1})$ [5].

C. Confidence Score

The confidence score $C(v_m)$ is incorporated to weight acoustically as well as linguistically reliable hypotheses [7]. Specifically, a posterior probability of each transcribed word, that is the ratio of a word hypothesis probability to that of all other hypotheses, is calculated using a word graph obtained by a decoder and used as a confidence measure [8], [9]. A word graph consisting of nodes and links from a beginning node S to an end node T in time course is shown in Fig. 2.

Nodes represent time boundaries between possible word hypotheses and links connecting these nodes represent word hypotheses. Each link is given acoustic log likelihood and linguistic log likelihood of a word hypothesis.

The posterior probability of a word hypothesis $w_{k,l}$ is given by

$$C(w_{k,l}) = \log \frac{\alpha_k P_{ac}(w_{k,l}) P_{lg}(w_{k,l}) \beta_l}{\mathcal{G}} \quad (3)$$

where

- k, l node number in a word graph $(k < l)$;
- $w_{k,l}$ word hypothesis occurred between node k and node l;
- $C(w_{k,l})$ log of the posterior probability of $w_{k,l}$;
- α_k forward probability from the beginning node S to node k;
- β_l backward probability from node l to the end node T;
- $P_{ac}(w_{k,l})$ acoustic likelihood of $w_{k,l}$;
- $P_{lg}(w_{k,l})$ linguistic likelihood of $w_{k,l}$;
- $\mathcal{G}$ forward probability from the beginning node S to the end node T $(=\alpha_T)$.

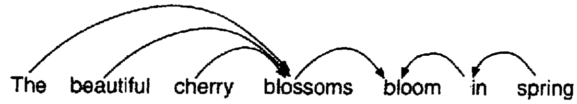

Fig. 3. Example of dependency structure.

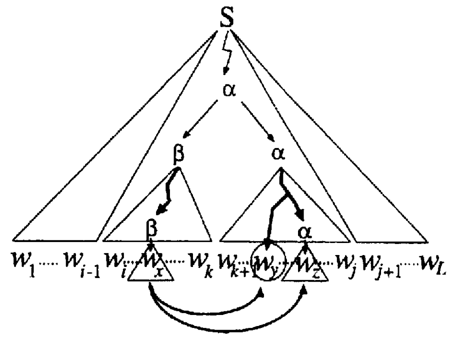

Fig. 4. Phrase structure tree based on a word-based dependency structure.

D. Word Concatenation Score

Suppose "the beautiful cherry blossoms in Japan" is summarized as "the beautiful Japan." The latter phrase is grammatically correct but a semantically incorrect summarization. Since the above linguistic score is not powerful enough to alleviate such a problem, a word concatenation score $T(v_{m-1}, v_m)$ is incorporated to give a penalty for a concatenation between words with no dependency in the original sentence. Every language has its own dependency structure, and in Section II-D1, a basic computation of the word concatenation score independent of the type of language is described. In the following section, this computation is adjusted to process the dependency structure specific to the Japanese language.

1) Word Concatenation Score: Dependency structure—An example of the dependency structure represented by a dependency grammar is shown as the curved arrows in Fig. 3. In a dependency grammar, one word is designated as the head of a sentence, and all other words are either a dependent of that word, or dependent on some other word which connects to the head word through a sequence of dependencies [10]. The word at the beginning of an arrow is named the "modifier" and the word at the end of the arrow is named the "head" respectively. For instance, the dependency grammar of English consists of both "right-headed" dependency indicated by right arrows and "left-headed" dependency indicated by left arrows as shown in Fig. 3. These dependencies can be represented by a phrase structure grammar, dependency context free grammar (DCFG), using the following rewrite rules based on Chomsky normal form:

$$\begin{aligned}&\alpha \rightarrow \beta\alpha \quad \texttt{(right-headed)}\\ &\alpha \rightarrow \alpha\beta \quad \texttt{(left-headed)}\\ &\alpha \rightarrow w\end{aligned}$$

where α and β are nonterminal symbols and w is a terminal symbol (word). Fig. 4 illustrates an example of a phrase structure tree based on a word-based dependency structure for a sentence which consists of L words, $w_1, \ldots, w_L$. The w_x modifies w_z, when a sentence is derived from the initial symbol S and the following requirements are fulfilled: 1) the rule of $\alpha \rightarrow \beta\alpha$ is applied; 2) $w_i \ldots w_k$ is derived from β; 3) w_x is derived from β; 4) $w_{k+1} \ldots w_j$ is derived from α; and 5) w_z is derived from α.

Dependency probability: Since dependencies between words are usually ambiguous, whether dependencies exist or not between words must be estimated by a dependency probability that one word is modified by others. In this study, the dependency probability is calculated as a posterior probability estimated by the inside–outside probabilities [11] based on SDCFG. The probability that the w_x and w_z relationship has a "right-headed" dependency structure is calculated as a product of the probabilities of the above-mentioned steps from 1) to 5). On the other hand, the "left-headed" dependency probability is calculated as the product of the probabilities when the rule of $\alpha \rightarrow \alpha\beta$ is applied. Since English has both right and left dependencies, the dependency probability is defined as the sum of the "right-headed" and "left-headed" dependency probabilities. If a language has only "right-headed" dependency, the "right-headed" dependency probability is used for the dependency probability. For simplicity, the dependency probabilities between w_x and w_z is denoted by $d(w_x, w_z, i, k, j)$, where i, k are the indices of the initial and final words derived from β, and j is the index of the final word derived from α.

Word concatenation probability: In a summarized sentence generated from the example in Fig. 3, "beautiful" can be directly connected with "blossoms" and also with "cherry" which modifies "blossoms." In general, as shown in Fig. 4, a modifier derived from β can be directly connected with a head derived from α in a summarized sentence. In addition, the modifier can be also connected with each word which modifies the head. The word concatenation probability between w_x and w_y is defined as a sum of the dependency probabilities between w_x and w_y, and between w_x and each of the $w_{y+1} \ldots w_z$. Using the dependency probabilities $d(w_x, w_y, i, k, j)$, the word concatenation score is calculated as a logarithmic value of the word concatenation probability given by

$$T(w_x, w_y) = \log \sum_{i=1}^{x} \sum_{k=x}^{y-1} \sum_{j=y}^{L} \sum_{z=y}^{j} d(w_x, w_z, i, k, j). \quad (4)$$

2) Word Concatenation Score for Japanese: Japanese has a different dependency structure from English. In order to efficiently summarize Japanese speech, the word concatenation score must be converted for the dependency structure of Japanese. Japanese sentences are divided into phrase-like units (*bunsetsu*), as exemplified in Fig. 5. We denote the phrase-like unit *bunsetsu* by "phrase." Since each content word always starts a new phrase, it is easy to convert a sentence into a phrase sequence. According to the modification rules for Japanese, a content word modifies function words following it, and forms one phrase. Each phrase is made up of a content word followed by zero or more function words, and each word modifies succeeding words within the phrase.

Japanese sentences have only "right-headed" dependency indicated by right arrows in Fig. 5. In addition, word dependency structures in each phrase are deterministic and can be represented by the regular grammar. The dependency structures of Japanese sentences can be represented by *interphrase* and *intraphrase* dependencies. The dependency structures

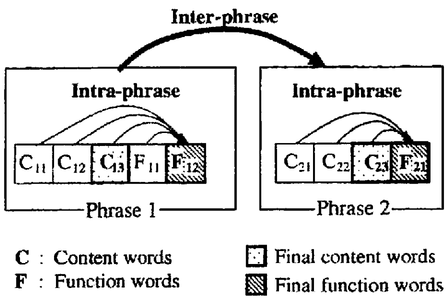

Fig. 5. Japanese phrase-based dependency structure.

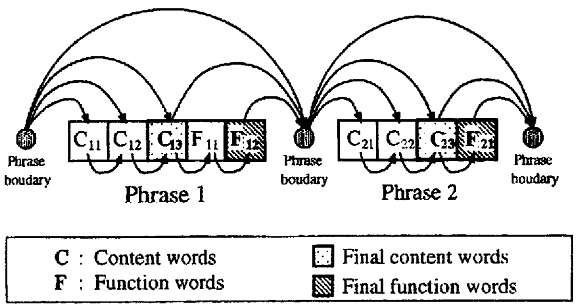

Fig. 6. Intraphrase rule.

between phrases (*interphrase* dependency) can be represented as follows:

$$\alpha \rightarrow \beta\alpha$$
$$\alpha \rightarrow P$$

where P is a phrase. On the other hand, the dependency structures between words in each phrase (*intraphrase* dependency) can be represented as follows:

$$\alpha \rightarrow \beta w$$
$$\alpha \rightarrow w$$

where w is a word. A word concatenation probability between words within a phrase of the original sentence is calculated using *intraphrase word concatenation probability* based on a rule described below. Word concatenation probability between words in different phrases is calculated using *interphrase word concatenation probability* based on a phrase-based SDCFG.

Intraphrase word concatenation probability: Since a dependency structure between words within a phrase is deterministic in Japanese, *intraphrase word concatenation probability* is set to 0 or 1 by the *intraphrase word concatenation rule* consisting of the following four rules.

1) A phrase boundary can be connected to any content words in the succeeding phrase.
2) The final content word or the final function word in a phrase can be connected to the succeeding phrase boundary.
3) Each word in a phrase can be connected to the next word in the same phrase.
4) A phrase boundary can be connected to any following phrase boundaries.

Fig. 6 illustrates word concatenations allowed in a summarized sentence based on the *intraphrase word concatenation rule* for

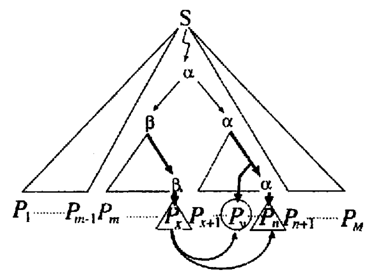

Fig. 7. Phrase structure tree based on a phrased-based dependency structure.

a sentence consisting of two phrases in Fig. 5. The arrows toward the right direction indicate possible concatenations between words within a phrase in a summarized sentence. Word concatenation probabilities between words within a phrase in the original sentence satisfying the *intraphrase word concatenation rule* in Fig. 6 are set to 1, and probabilities between words without satisfying the rule are set to 0. Summarizing a sentence based on the *intraphrase word concatenation rule* is exemplified using "phrase 1" in Fig. 6. The summarization process is one of the following types of word extractions.

1) No word is extracted from a phrase.
2) Only the final content word is extracted.
   $\{C_{13}\}$
3) Content word sequences including the final content words are extracted.
   $\{C_{12}C_{13}\}, \{C_{11}C_{12}C_{13}\}$
4) The final content word or content word sequence are attached to all function words.
   $\{C_{13}F_{11}F_{12}\}, \{C_{12}C_{13}F_{11}F_{12}\}, \{C_{11}C_{12}C_{13}F_{11}F_{12}\}$

Interphrase word concatenation probability: A word concatenation probability between words in different phrases is determined by a dependency structure between phrases. Since dependency between phrases is ambiguous, an *interphrase word concatenation probability* is calculated as a probability (phrase dependency probability) that one phrase is modified by others based on a phrase-based SDCFG [6].

The dependency probability between phrases is represented using the dependency probability between words described in Section II-D1. Suppose a sentence consists of M phrases, $P_1, \ldots, P_M$, the phrase dependency probabilities between P_x and $P_z (1 \le x \le z \le M)$ is defined as $d_p(P_x, P_z, m, l, n)$ by converting a word dependency probability as shown in Fig. 4 in Sectioin II-D1, where M, m, l, and n in $d_p(P_x, P_z, m, l, n)$ correspond to L, i, k, and j in $d(w_x, w_z, i, k, j)$ respectively.

Using the phrase dependency probabilities $d_p(P_x, P_z, m, l, n)$, the word concatenation score $T_p(P_x, P_y)$ between words in different phrases is calculated by

$$T_p(P_x, P_y) = \log \sum_{m=1}^{x} \sum_{l=x}^{y-1} \sum_{n=y}^{M} \sum_{z=y}^{n} d_p(P_x, P_z, m, l, n). \quad (5)$$

Since Japanese sentences can be represented only by the rule of $\alpha \rightarrow \beta\alpha$, the final phrase P_l, in a phrase string $P_m, \ldots, P_l$ derived from β, is always derived from the same nonterminal symbol β. The final phrase P_n, in a phrase strings $P_{l+1}, \ldots, P_n$ derived from α, is also derived from the same nonterminal symbol α. As shown in Fig. 7, the phrase dependency structure is simpler than the general word dependency structure illustrated in Fig. 4. Therefore, applying only $\alpha \rightarrow \beta\alpha$ results in $l = x$ and $z = n$. The word concatenation score $T_p(P_x, P_y)$ given by (5) is simplified as follows:

$$T_p(P_x, P_y) = \log \sum_{m=1}^{x} \sum_{n=y}^{M} d_p(P_x, P_y, m, x, n). \quad (6)$$

Here, $d_p(P_x, P_y, m, x, n)$ is calculated as a posterior probability estimated using the Inside-Outside probability [11] based on a phrase-based SDCFG described in the Appendix:

$$d(P_x, P_y, m, x, n) = \sum_{\alpha,\beta} g(\alpha \rightarrow \beta\alpha; m, x, n). \quad (7)$$

SDCFG is constructed using a manually parsed corpus. Parameters of SDCFG are estimated using the Inside-Outside algorithm as described in the Appendix. In our SDCFG [6], only the number of nonterminal symbols is determined and all possible phrase trees are considered. The rules consisting of all combinations of nonterminal symbols are applied to each rewriting symbol in a phrase tree. In this method, the nonterminal symbol is not given a specific function such as a noun phrase function, and the function of nonterminal symbols are automatically learned from data. Probabilities for frequently used rules become bigger, and those for rarely used rules become smaller. Since words in the learning data for SDCFG are tagged with POS (part-of-speech), the dependency probability of words excluded in the learning data can be calculated based on their POS. Even if the transcription results obtained by a speech recognizer are ill-formed, the dependency structure can be robustly estimated by the SDCFG.

Computation of word concatenation score for Japanese: Suppose w_x and w_y belong to $P_{\mathrm{ph}(w_x)}$ and $P_{\mathrm{ph}(w_y)}$ respectively, where $\mathrm{ph}(w)$ denotes an index of a phrase including a word w. A word concatenation score of w_x and w_y within a phrase ($\mathrm{ph}(w_x) = \mathrm{ph}(w_y)$) is calculated using the *intraphrase word concatenation rule* ($R(w_x, w_y) = 0, 1$). On the other hand, the word concatenation score when w_x and w_y occur in different phrases ($\mathrm{ph}(w_x) < \mathrm{ph}(w_y)$) is calculated using a dependency probability between $P_{\mathrm{ph}(w_x)}$ and $P_{\mathrm{ph}(w_y)}$ based on phrase-based SDCFG. The word concatenation score $T(w_x, w_y)$ is calculated as a logarithmic value of the word concatenation probability as follows:

$$T(w_x, w_y) = \begin{cases} T_p(P_{\mathrm{ph}(w_x)}, P_{\mathrm{ph}(w_y)}), & \text{if } \mathrm{ph}(w_k) < \mathrm{ph}(w_l) \\ \log R(w_k, w_l), & \text{if } \mathrm{ph}(w_k) = \mathrm{ph}(w_l). \end{cases} \quad (8)$$

E. Dynamic Programming for Automatic Summarization

Given a transcription result consisting of N words, $W = w_1, w_2, \ldots, w_N$, the summarization is performed by extracting a set of $M(M < N)$ words, $V = v_1, v_2, \ldots, v_M$, which maximizes the summarization score given by (1). The algorithm is as follows:

1. Definition of symbols and variables

$\langle s \rangle$: beginning symbol of a sentence
$\langle /s \rangle$: ending symbol of a sentence
$P(w_n \mid w_k w_l)$: linguistic score
$I(w_n)$: word significance score
$C(w_n)$: confidence score
$T(w_l, w_n)$: word concatenation score
$s(k, l, n)$: summarization score of each word

$$s(k, l, n) = \log P(w_n \mid w_k w_l) + \lambda_I I(w_n) + \lambda_C C(w_n) + \lambda_T T(w_l, w_n)$$

$g(m, l, n)$: summarization score of a subsentence $\langle s \rangle, \ldots, w_l, w_n$, consisting of m words, beginning from $\langle s \rangle$, and ending $w_l, w_n (0 \le l < n \le N)$
$B(m, l, n)$: back pointer

2. Initialization

Summarization score is calculated for each subsentence hypothesis consisting of one word. $-\infty$ is given for each word which is never selected as the first word in the summarization sentence consisting of M words (see the equation at the bottom of the page).

3. The DP process

A dynamic programming recursion is applied for each pair of the last two words (w_l, w_n) of each subsentence hypothesis consisting of m words.

for $m = 2$ to M
 for $n = m$ to $N - m + 1$
  for $l = m - 1$ to $n - 1$

$$g(m, l, n) = \max_{k<l}\{g(m-1, k, l) + s(k, l, n)\}$$
$$B(m, l, n) = \operatorname*{argmax}_{k<l}\{g(m-1, k, l) + s(k, l, n)\}$$

4. Select the optimal path

The best complete hypothesis consisting of M words is decided by selecting the last two words $(w_{\hat{l}}, w_{\hat{n}})$.

$$S(V) = \max_{\substack{N-M<n\le N \\ N-M-1<l\le N-1}} g(M, l, n) + \log P(\langle /s \rangle \mid w_l w_n)$$
$$(\hat{l}, \hat{n}) = \operatorname*{argmax}_{\substack{N-M<n\le N \\ N-M-1<l\le N-1}} g(M, l, n) + \log P(\langle /s \rangle \mid w_l w_n)$$

5. Backtracking

We can get the word sequence $V = v_1 \ldots v_M$ of the best summarization result by backtracking the back pointers retained in 3.

for $m = M$ to 1
 $v_m = w_{\hat{n}}$
 $l' = B(m, \hat{l}, \hat{n})$
 $\hat{n} = \hat{l}$
 $\hat{l} = l'$

$$g(1, 0, n) = \begin{cases} \log P(w_n \mid \langle s \rangle) + \lambda_I I(w_n) + \lambda_C C(w_n), & \text{if } 1 \le n \le (N - M + 1) \\ -\infty, & \text{otherwise} \end{cases}$$

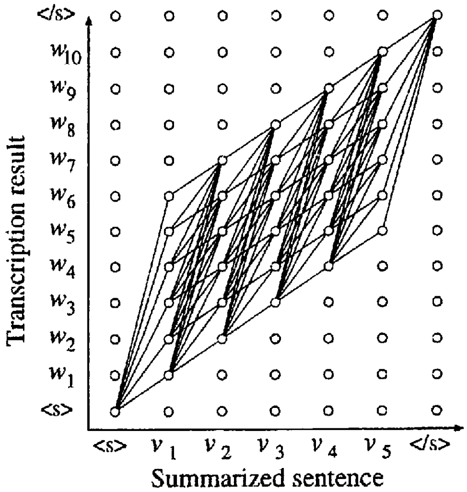

Fig. 8. Example of DP alignment for speech summarization.

The two-dimensional space for performing the dynamic programming process is shown in Fig. 8. The vertical axis indicates the transcription result consisting of ten words, and the horizontal axis indicates the summarized sentence having five words. All possible sets of five words extracted from the ten words are indicated by the paths from the bottom-left corner to the top-right corner.

III. Summarization of Multiple Utterances With Consistent Meanings

Our proposed automatic speech summarization technique for each sentence can be extended to summarize a set of multiple utterances (sentences) having consistent meanings by combining a rule which gives restrictions at sentence boundaries. As a result, original sentences including many informative words are preserved, and sentences including few informative words are deleted or shortened.

Given a transcription result consisting of J utterances, $S_1, \ldots, S_J (S_j = w_{j1}, w_{j2}, \ldots, w_{jN_j})$ the summarization is performed by extracting a set of M $(M < \sum_j N_j)$ words, $V = v_1, v_2, \ldots, v_M$ which maximizes the summarization score given by (1). The algorithm is as follows:

1. Definition of symbols and variables

$s_j(k,l,n)$: summarization score of each word

$$s_j(k,l,n) = \log P(w_{jn} \mid w_{jk} w_{jl}) + \lambda_I I(w_{jn}) + \lambda_C C(w_{jn}) + \lambda_T T(w_{jl}, w_{jn})$$

$g_j(m,l,n)$: local optimal score of $\langle s \rangle, w_{11}, \ldots, w_{jl}, w_{jn}$ consisting of m words beginning with $\langle s \rangle$ of the sentence S_1 and ending with w_{jl}, w_{jn} in the sentence$S_j (0 \leq l < n \leq N_j)$

$G_j(m)$: local optimal score at the end of the sentence, consisting of m words beginning with $\langle s \rangle$ of the sentence 1 and ending with $\langle /s \rangle$ in the sentence j

$b_j(m,l,n)$: back pointer

$B_j(m)$: back pointer of the end of a sentence

2. Initialization

$$G_0(m) = \begin{cases} 0, & m = 0 \\ -\infty, & \text{otherwise} \end{cases}$$
$$B_0(m) = \phi$$

3. The DP process

Dynamic programming recursion is applied and the summarization score is summed up through sentences $S_1 \ldots S_J$.

for $j = 1$ to J

Calculation for the beginning of a sentence: the summarization score is calculated as the score up to the preceding sentence, $G_{j-1}(m-1)$, plus the score for the first one word selected from the current sentence (see the equation at the bottom of the page).

Calculation for the inside of a sentence: DP recursion is applied for each sentence in the same manner as that of sentence-by-sentence summarization described in Section II-E:

for $m = j \times 2$ to N_j
for $n = 2$ to N_j
for $l = 1$ to $n - 1$

$$g_j(m,l,n) = \max_{0 \leq k < l} \{g_j(m-1,k,l) + s_j(k,l,n)\}$$
$$b_j(m,l,n) = \operatorname*{argmax}_{0 \leq k < l} \{g_j(m-1,k,l) + s_j(k,l,n)\}.$$

Calculation for the end of a sentence: the score of the local best hypothesis up to the end of S_j is calculated:

$$G_j(m) = \max_{\substack{0 < n \leq N_j \\ 0 \leq l \leq N_j - 1}} g_j(m,l,n) + \log P(\langle /s \rangle \mid w_{jl} w_{jn})$$
$$(\hat{l}, \hat{n}) = \operatorname*{argmax}_{\substack{0 < n \leq N_j \\ 0 \leq l \leq N_j - 1}} g_j(m,l,n) + \log P(\langle /s \rangle \mid w_{jl} w_{jn})$$
$$B_j(m) = (\hat{l}, \hat{n})$$

4. Backtracking

We can get the word sequence $V = v_1 \ldots v_M$ of the best summarization result for the multiple utterances by backtracking the back pointers retained within each sentence and at the end of each sentence, where

$j = J$
$m = M$

$$g_j(m,0,n) = \begin{cases} G_{j-1}(m-1) + \log P(w_{jn} \mid \langle s \rangle) + \lambda_I I(w_{jn}) + \lambda_C C(w_{jn}), & \text{if } 1 \leq n \leq N_j \\ -\infty, & \text{otherwise} \end{cases}$$
$$b_j(m,0,n) = \phi.$$

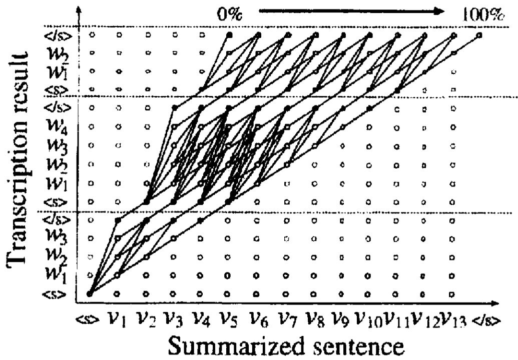

Fig. 9. Example of DP process for summarization of multiple utterances.

while $m > 0$
 $v_m = w_{\hat{n}}$
 $l' = b_j(m, \hat{l}, \hat{n})$
 $\hat{n} = \hat{l}$
 if $l' \neq \phi$ then
  $\hat{l} = l'$
  $m = m - 1$
 else
  $v_{m-1} = \langle /s \rangle$
  $v_{m-2} = \langle s \rangle$
  $(\hat{l}, \hat{n}) = B_{j-1}(m - 2)$
  $m = m - 3$
  $j = j - 1$

Fig. 9 illustrates the DP process for summarizing multiple utterances. This summarization technique can be considered as a combination of the summarization method developed in the field of natural language processing which extracts important sentences, and our sentence-by-sentence summarization method.

IV. Evaluation

A. Word Network of Manual Summarization Results for Evaluation

To automatically evaluate summarized sentences, correctly transcribed speech is manually summarized by human subjects and used as correct targets. The manual summarization results are merged into a word network, and the word accuracy of automatic summarization given by (9) is calculated using the word network as the summarization accuracy. The network approximately expresses all possible correct summarization including subjective variations:

$$\text{Accuracy} = \frac{\text{Len} - \text{Sub} - \text{Ins} - \text{Del}}{\text{Len}} \times 100 \quad [\%] \qquad (9)$$

where

- Sub number of substitution errors;
- Ins number of insertion errors;
- Del number of deletion errors;
- Len number of words in the most similar word string in the network.

The summarization accuracy is defined by the word accuracy based on the word string extracted from the word network that is most similar to the automatic summarization result. This accuracy is expected to indicate linguistic correctness and maintenance of original meanings of the utterance.

B. Evaluation Data

Japanese news speech data broadcast on TV in 1996 was used as a test set to evaluate our proposed method. The set consisted of 419 utterances by a female anchor speaker, and was manually segmented into sentences. The out-of-vocabulary (OOV) rate for the 20 k word vocabulary was 2.5% and the perplexity for the test set was 54.5. 50 utterances with word recognition accuracy above 90%, which was the average rate over the 50 utterances, were selected and used for the evaluation. The summarization ratio, the ratio of the number of characters in the summarized sentences to that in the original sentences, was set to 40, 60, 70, and 80%.

In addition, five news articles consisting of approximately five sentences each were summarized using the summarization technique for multiple utterances at 30% summarization ratio.

C. Structure of the Broadcast News Transcription System

1) Acoustic Models: The feature vector extracted from speech consists of 16 cepstral coefficients, normalized logarithmic power, and their delta features (derivatives). The total number of parameters in each vector is 34. Cepstral coefficients were normalized using the CMS (cepstral mean subtraction) method. The acoustic models used were shared-state triphone HMMs designed using tree-based clustering. The total number of states was 2106, and the number of Gaussian mixture components per state was four. They were trained using phonetically-balanced sentences and dialogues read by 53 speakers (approximately 20 h in total). They were completely different from the broadcast news task. All of the speakers were male, and so the HMMs were gender-dependent models. The total number of training utterances was 13 270 and the total length of the training data was approximately 20 hours.

2) Language Models: Broadcast-news manuscripts recorded from August 1992 to May 1996, comprising of approximately 500 k sentences consisting of 22 M words, were used for constructing language models. The vocabulary size was 20 k words. To calculate word n-gram language models, we segmented the broadcast-news manuscripts into words by using a morphological analyzer since Japanese sentences are written without spaces between words. In addition words are tagged with POS by the morphological analyzer at the same time.

3) Decoder: We used a word-graph-based 2-pass decoder for transcription. In the first pass, frame-synchronous beam search was performed using the above-mentioned HMMs and a bigram language model. A word graph was generated as a result of the first pass. In the second pass, the word graph was rescored using a trigram language model. Since each word entry is tagged with POS, e.g., cherry/noun, the/preposition, etc., in our Japanese LVCSR (Large Vocabulary Continuous Speech Recognition) system, recognition results obtained by our system are words appended POS.

TABLE I
SUMMARIZATION TYPES OF MANUAL TRANSCRIPTION

Target	Manual Transcription(TRS)			
Symbol	RDM	I_L	I_L_T	SUB_TRS
Manual summarization				○
Significance score (I)		○	○	
Linguistic score (L)		○	○	
Word concatenation score (T)			○	
Random word selection	○			

TABLE II
SUMMARIZATION TYPES OF AUTOMATIC TRANSCRIPTION

Target	Automatic Transcription(REC)				
Symbol	RDM	I_L_C	$I\_L\_C\_T$	I_L_T	SUB_REC
Manual summarization					○
Significance score (I)		○	○	○	
Linguistic score (L)		○	○	○	
Confidence score (C)		○	○		
Word concatenation score (T)			○	○	
Random word selection	○				

D. Training Data for Summarization Models

1) Word Significance Model: The same broadcast-news manuscripts used for building a language model in speech recognition was used for calculating the word significance measure for summarization.

2) Language Model: A trigram language model for summarization was built using text from Mainichi newspaper published from 1996 to 1998, comprising of 5.1 M sentences with 87 M words. We consider that the newspaper text is usually more compact and simpler than broadcast news text and therefore more appropriate for building language models for summarization than the latter. Our previous experiments confirmed that the automatically summarized sentences using word trigram based on newspaper text were much better than those by broadcast-news manuscripts [5].

3) SDCFG: SDCFG for word concatenation score was built using text from the manually parsed corpus of Mainichi newspaper published from 1996 to 1998, comprising of approximately 4 M sentences with 68 M words. The number of non-terminal symbols was 100.

E. Evaluation Results

Tables I and II respectively show the types of summarization of manual transcription (TRS) and automatic transcription (REC) investigated in this paper. In these tables the symbols of I, L, C, and T indicate the utilization of word significance score, linguistic score, confidence score and word concatenation score for summarization respectively.

In the summarization of REC, conditions with ($I\_L\_C\_T$) and without (I_L_T) the word confidence score were compared. Conditions with (I_L_T, $I\_L\_C\_T$) and without (I_L, I_L_C) the word concatenation score were compared in summarization for both TRS and REC.

To set the upper limit of the automatic summarization, manual summarization by human subjects for manual transcription (SUB_TRS) was performed. The results were evaluated using all other manual summarization results as correct summarization. In addition, as the upper bound of automatic speech summarization for transcription including speech recognition errors, manual

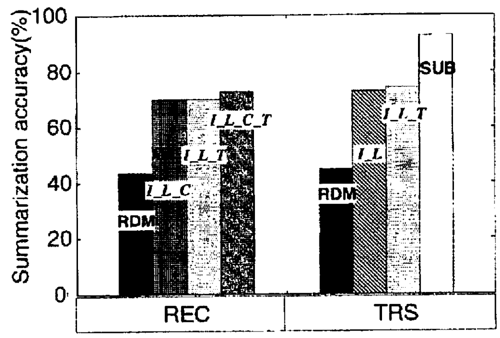

Fig. 10. Each utterance summarization results at 40% summarization ratio.

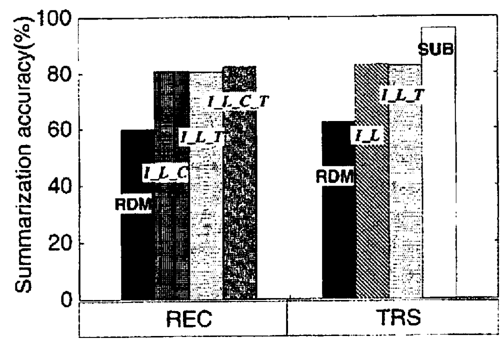

Fig. 11. Each utterance summarization results at 60% summarization ratio.

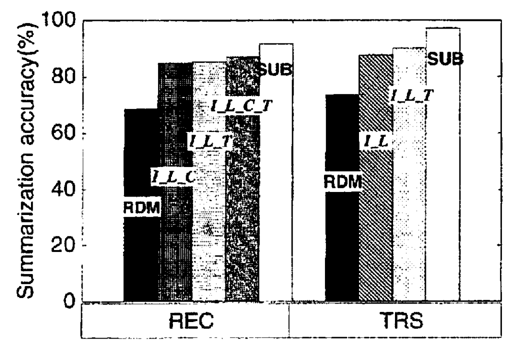

Fig. 12. Each utterance summarization results at 70% summarization ratio.

summarization of automatically transcribed utterances at 70% summarization ratio was also evaluated (SUB_REC). To insure that our method is sound, we made randomly generated summarization sentences (RDM) according to the summarization ratio and compared them with those obtained by our proposed methods.

1) Summarization of Each Utterance: Figs. 10–13 show summarization accuracy of both manual transcription (TRS) and automatic transcription (REC) at 40%, 60%, 70%, and 80% summraization ratios. These results show that our proposed automatic speech summarization technique is significantly more effective than RDM. The method using the word concatenation score (I_L_T, $I\_L\_C\_T$) can reduce meaning alteration compared with the method without using the word concatenation score (I_L, I_L_C). The better result using the word concatenation score ($I\_L\_C\_T$) compared with that without using the word concatenation score (I_L_T) shows that the summarization accuracy is significantly improved by the confidence score.

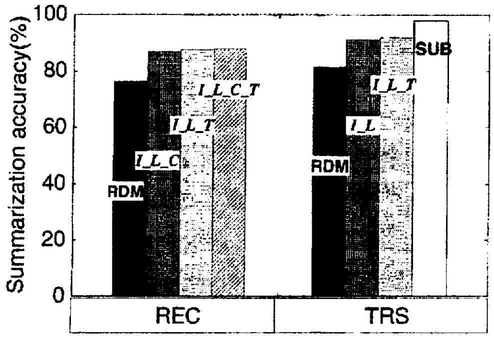

Fig. 13. Each utterance summarization results at 80% summarization ratio.

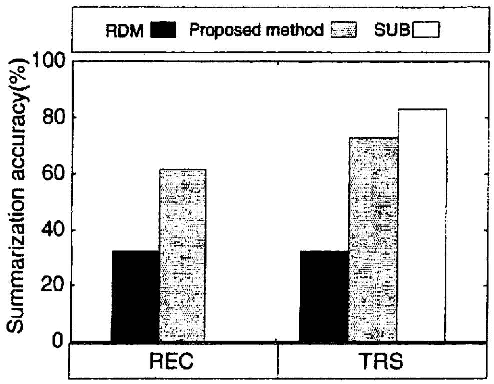

Fig. 14. Article summarization results at 30% summarization ratio.

The performance of automatic summarization of automatic transcription (REC) is comparable with that of manual transcription (TRS) under all the conditions of summarization ratio. Although automatic summarization cannot achieve the performance of the manual summarization of manual transcription (SUB_TRS), it can achieve the performance comparable to the manual summarization of the recognition result (SUB_REC).

2) Summarization of Multiple Utterances: Fig. 14 shows the summarization accuracy of summarizing articles having multiple sentences at 30% summarization ratio. These results show that our proposed automatic speech summarization technique is effective for the summarization of multiple utterances.

V. Conclusion

An automatic speech summarization method based on a word significance score, linguistic likelihood, a word confidence measure and a word concatenation probability has been proposed. A dependency structure in the original sentence given by SDCFG was used to determine the word concatenation probability. A word set maximizing the total score was extracted using dynamic programming techniques and connected to build a summarized sentence. The summarization was performed according to the users' required amount of information.

Each utterance and multiple utterances with consistent meanings of Japanese broadcast news speech was summarized by our proposed method. Experimental results show that the proposed method can effectively extract relatively important information and remove redundant and irrelevant information. A confidence score giving a penalty for acoustically as well as linguistically unreliable words could reduce the meaning alteration of summarization caused by recognition errors. A word concatenation score giving a penalty for a concatenation between words with no dependency in the original sentence could also reduce the meaning alteration of summarization.

In this study, newspaper text was used for training linguistic models for summarization. If we could use a summarization model constructed using a manual summarization corpus, the automatic summarization performance should be improved.

We proposed a new method for measuring the summarization accuracy based on a word network constructed using manual summarization results. Our future research will include task-dependent evaluation methods such as those for information retrieval. This is because summarization obtained from ill-formed speech are sometimes linguistically incorrect but semantically correct and understandable. They need to be evaluated from the viewpoint of how much the original meaning is maintained in the summarization results.

Our future work also includes the application of summarization scores to the word graph instead of transcription. This method is expected to contribute to increase the performance of speech recognition. We are also planning to apply our summarization method to making abstracts of various monologues such as lectures and presentations.

Appendix

Parameter Re-Estimation in Phrase-Based SDCFG

Parameters of a phrase-based SDCFG are estimated from a manual parsed corpus using the Inside-Outside algorithm. Since words in the corpus are tagged with POS, phrase boundaries are automatically detected based on the POS. Each phrase is made up of a content word followed by zero or more function words. In this study, content words include nouns, adjectives, verbs and adverbs, and the remaining words are included as function words. Suppose a sentence consists of M phrases:

$$S \rightarrow P_1 \dots P_m \dots P_M$$

P_m is defined as follows:

$$P_m = w_{mc} w_{mf,1} w_{mf,2} \cdots w_{mf,Km}$$

where

- M number of phrases in a sentence;
- w_{mc} content word of the m-the phrase;
- $w_{mf,i}$ ith function word on mth phrase;
- K_m number of functions words in mth phrase.

Rewrite probabilities of $\alpha \rightarrow \beta\alpha, \alpha \rightarrow w_c, \alpha \rightarrow \beta w_f$ are denoted by $a(\alpha \rightarrow \beta\alpha), b(\alpha \rightarrow w_c), c(\alpha \rightarrow \beta w_f)$ respectively. The algorithm for estimating parameters of the phrase-based SDCFG is described below. Fig. 15 indicates the estimation steps.

1) Initialization

   $a(\alpha \rightarrow \beta\alpha)$ is given a flat probability and $b(\alpha \rightarrow w_c), c(\alpha \rightarrow \beta w_f)$ are given random values.

2) Calculation for intra phrase forward probability

   The probability of deriving $w_{mc} w_{mf,1} \cdots w_{mf,i}$ from α in the mth phrase is calculated by the forward probability illustrated in Fig. 15(a):

$$h(m, i, \alpha) = P(\alpha \rightarrow w_{mc} w_{mf,1} \cdots w_{mf,i}) = \begin{cases} b(\alpha \rightarrow w_{mc}), & \text{if } i = 0 \\ \sum_{\beta} h(m, i-1, \beta) c(\alpha \rightarrow \beta w_{mf,i}), & \text{if } i > 0. \end{cases} \tag{10}$$

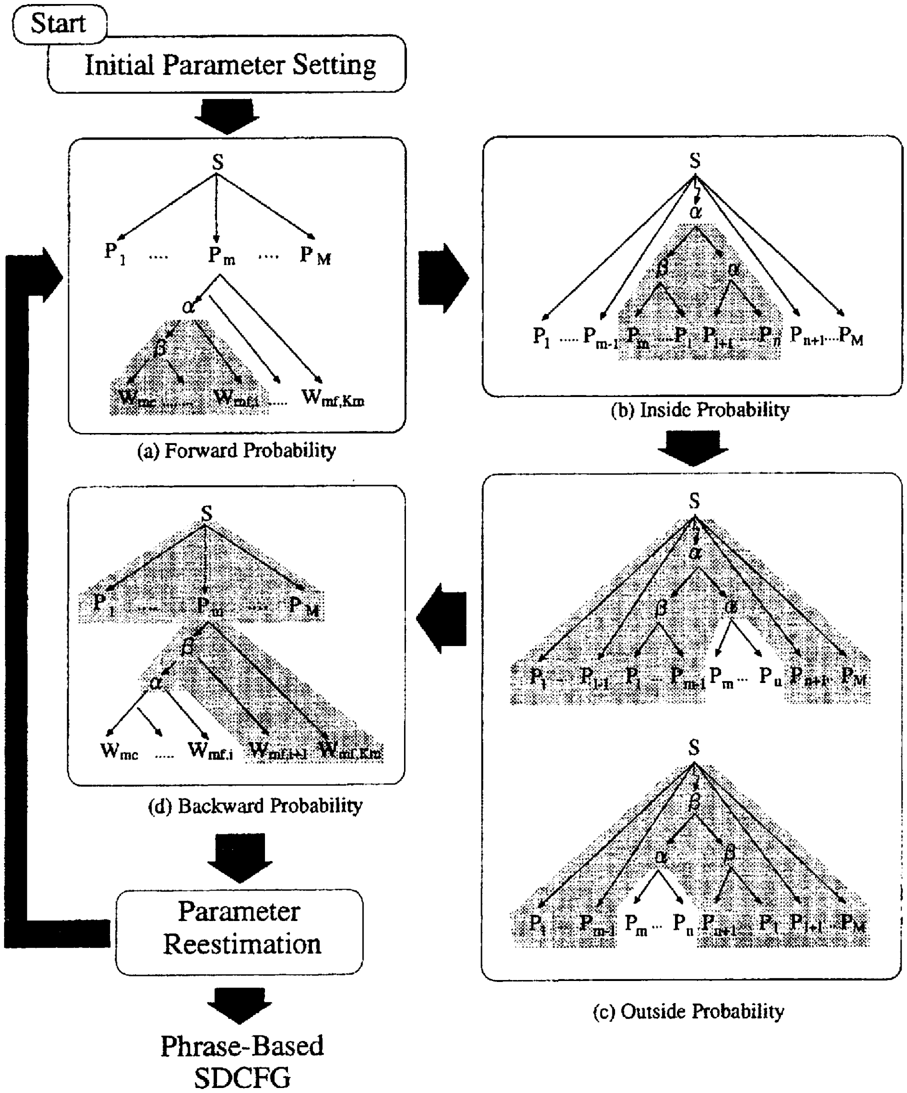

Fig. 15. Estimation algorithm of phrase-based SDCFG.

3) Calculation of the interphrase inside probability

The interphrase inside probability illustrated in Fig. 15(b) is calculated using the intraphrase forward probability:

$$\begin{aligned} e(m,n\,|\,\alpha) &= P(\alpha \rightarrow P_m \ldots P_n) \\ &= \begin{cases} h(m, K_m, \alpha), & \text{if } m = n \\ \displaystyle\sum_{l=m}^{n-1} \sum_{\beta} a(\alpha \rightarrow \beta\alpha) e(m,l\,|\,\beta) e(l+1,n\,|\,\alpha), & \text{if } m < n. \end{cases} \end{aligned} \tag{11}$$

4) Calculation of the interphrase outside probability

The interphrase outside probability illustrated in Fig. 15(c) is calculated using the interphrase inside probability:

$$\begin{aligned} f(m,n\,|\,\alpha) &= P(S \rightarrow P_1 \ldots P_{m-1} \alpha P_{n+1} \ldots P_M) \\ &= \sum_{l=1}^{m-1} \sum_{\beta} a(\alpha \rightarrow \beta\alpha) e(l, m-1\,|\,\beta) f(l,n\,|\,\alpha) \\ &\quad + \sum_{l=n+1}^{M} \sum_{\beta} a(\beta \rightarrow \alpha\beta) e(n+1, l\,|\,\beta) f(m,l\,|\,\beta). \end{aligned} \tag{12}$$

5) Calculation of the intraphrase backward probability

The intraphrase backward probability illustrated in Fig. 15(d) is calculated as follows using the interphrase outside probability:

$$\begin{aligned} r(m,i,\alpha) &= P(S \rightarrow P_1 \ldots P_{m-1} \alpha w_{mf,i+1} \ldots w_{mf,K_m} P_{m+1} \ldots P_M) \\ &= \begin{cases} f(m,m,\alpha), & \text{if } i = K_m \\ \displaystyle\sum_{\beta} c(\beta \rightarrow \alpha w_{mf,i+1}) r(m,i+1,\beta), & \text{if } i < K_m. \end{cases} \end{aligned} \tag{13}$$

$$\hat{a}(\alpha \rightarrow \beta\alpha) = \frac{\sum_{m=1}^{M-1} \sum_{n=m+1}^{M} \sum_{l=m}^{n-1} g(m, l, n; \alpha \rightarrow \beta\alpha)}{e(1, M \mid S)} \tag{14}$$

$$\hat{b}(\alpha \rightarrow w_c) = \frac{\sum_{m=1: w_{m,c}=w_c}^{M} b(\alpha \rightarrow w_c) r(m, 0, \alpha)}{e(1, M \mid S)} \tag{15}$$

$$\hat{c}(\alpha \rightarrow \beta w_f) = \frac{\sum_{m=1}^{M} \sum_{i=1: w_{m,f,i}=w_f}^{K_m} h(m, i-1, \beta) c(\alpha \rightarrow \beta w_f) r(m, i, \alpha)}{e(1, M \mid S).} \tag{16}$$

6) Estimation of parameters
The parameters are re-estimated using the probabilities obtained by the steps 2) to 5); see (14)–(16), shown at the top of the page, where

$$g(m, l, n; \alpha \rightarrow \beta\alpha) = e(m, l \mid \beta) e(l+1, n \mid \alpha) a(\alpha \rightarrow \beta\alpha) f(m, n \mid \alpha). \tag{17}$$

7) The steps from 2 to 6 are iterated until the parameters are saturated.

ACKNOWLEDGMENT

The authors would like to thank NHK (Japan Broadcasting Corporation) for providing us with the broadcast news database.

REFERENCES

[1] T. Imai, A. Kobayashi, S. Sato, H. Tanaka, and A. Ando, "Progressive 2-pass decoder for real-time broadcast news captioning," in *Proc. ICSLP2000, 6th Int. Conf. on Spoken Language Processing*, Beijing, China, 2000, pp. I-246–I-249.

[2] S. Furui, K. Maekawa, H. Isahara, T. Shinozaki, and T. Ohdaira, "Toward the realization of spontaneous speech recognition—Introduction of a japanese priority program and preliminary results," in *Proc. ICSLP2000, 6th Int. Conf. on Spoken Language Processing*, Beijing, China, 2000, pp. III-518–III-521.

[3] R. Valenza, T. Robinson, M. Hickey, and R. Tucker, "Summarization of spoken audio through information extraction," in *Proc. ESCA Workshop on Accessing Information in Spoken Audio*, 2000, pp. 111–116.

[4] I. Manu and M. Maubury, *Advances in Automatic Text Summarization*. Cambridge, MA: MIT Press, 1999.

[5] C. Hori and S. Furui, "Automatic speech summarization based on word significance and linguistic likelihood," in *Proc. ICASSP 2000*, Istanbul, Turkey, 2000, pp. 1579–1582.

[6] A. Ito, C. Hori, M. Katoh, and M. Kohda, "Language modeling by stochastic dependency grammar for japanese speech recognition," in *Proc. ICSLP 2000*, Beijing, China, 2000, pp. I-246–I-249.

[7] C. Hori and S. Furui, "Improvements in automatic speech summarization and evaluation methods," in *Proc. ICSLP2000*, Beijing, China, 2000, pp. IV-326–IV-329.

[8] T. Kemp and T. Schaaf, "Estimating confidence using word lattices," in *Proc. 5th Eurospeech*, vol. 2, Rhodes, , Greece, 1997, pp. 827–830.

[9] V. Valtchev, J. J. Odel, P. C. Woodland, and S. J. Young, "MMIE training of large vocabulary recognition systems," *Speech Commun.*, vol. 22, pp. 303–314, 1997.

[10] C. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press, 2000.

[11] K. Lari *et al.*, "The estimation of stochastic context free grammars using the inside-outside algorithm," *Comput. Speech. Lang.*, vol. 4, pp. 35–56, 1990.

Chiori Hori (M'02) received B.E. and M.E. degrees in electrical and information engineering from Yamagata University, Yonezawa, Japan, in 1994 and 1997, respectively. In April 1999, she started the doctoral course in the Graduate School of Information Science and Engineering at Tokyo Institute of Technology (TITECH), Tokyo, Japan, and received the Ph.D. degree in March 2002.

From April 1997 to March 1999, she was a Research Associate with the Faculty of Literature and Social Sciences, Yamagata University. She is currently a Researcher with NTT Communication Science Laboratories (CS Labs), Nippon Telegraph and Telephone Corporation (NTT), Kyoto, Japan, which she joined in 2002.

Dr. Hori is a member of the Acoustical Society of Japan (ASJ) and the Institute of Electronics, Information and Communication Engineers of Japan (IEICE).

Sadaoki Furui (M'79–SM'88–F'93) is currently a Professor with the Department of Computer Science, Tokyo Institute of Technology, Japan. He is engaged in a wide range of research on speech analysis, speech recognition, speaker recognition, speech synthesis, and multimodal human–computer interaction and has authored or coauthored over 350 published articles. From 1978 to 1979, he served on the staff of the Acoustics Research Department, Bell Laboratories, Murray Hill, NJ, as a Visiting Researcher working on speaker verification. He is Editor-in-Chief of the *Transactions of the IEICE*. He is also an Editorial Board member of *Speech Communication*, the *Journal of Computer Speech and Language* and the *Journal of Digital Signal Processing*. He is the author of *Digital Speech Processing, Synthesis, and Recognition* (New York: Marcel Dekker, 1989; revised 2000) in English, *Digital Speech Processing* (Tokyo, Japan: Tokai University Press, 1985), in Japanese, *Acoustics and Speech Processing* (Tokyo, Japan: Kindai-Kagaku-Sha, 1992), in Japanese, and *Speech Information Processing* (Tokyo, Japan: Morikita, 1998), in Japanese. He edited *Advances in Speech Signal Processing* (New York: Marcel Dekker, 1992) jointly with Dr. M. M. Sondhi. He has translated into Japanese *Fundamentals of Speech Recognition*, authored by Dr. L. R. Rabiner and Dr. B.-H. Juang (Tokyo, Japan: NTT Advanced Technology, 1995) and *Vector Quantization and Signal Compression*, authored by Dr. A. Gersho and Dr. R. M. Gray (Tokyo, Japan: Corona-sha, 1998).

Dr. Furui is a Fellow of the Acoustical Society of America and the Institute of Electronics, Information and Communication Engineers of Japan (IEICE). He is President of the Acoustical Society of Japan (ASJ), the International Speech Communication Association (ISCA), and the Permanent Council for International Conferences on Spoken Language Processing (PC-ICSLP). He is a on the Board of Governors of the IEEE Signal Processing Society (SPS). He has served on the IEEE Technical Committee on Speech and MMSP and on numerous IEEE conference organizing committees. He has received the Yonezawa Prize and the Paper Award from the IEICE in 1975, 1988, and 1993, and the Sato Paper Award from the ASJ in 1985 and 1987. He received the Senior Award from the IEEE ASSP Society in 1989 and the Achievement Award from the Minister of Science and Technology, Japan, also in 1989. He has received the Book Award from the IEICE in 1990. He has also received the Mira Paul Memorial Award from the AFECT, India, in 2001. In 1993, he served as an IEEE SPS Distinguished Lecturer.