<table>
<thead>
<tr>
<th>項目</th>
<th>詳細</th>
</tr>
</thead>
<tbody>
<tr>
<td>論文</td>
<td>Chiori Hori, Sadaoki Furui</td>
</tr>
<tr>
<td>発行日</td>
<td>2003, 9</td>
</tr>
<tr>
<td>権利情報</td>
<td>(c)2003 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.</td>
</tr>
</tbody>
</table>
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 summarizing 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
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

\[
S(V) = \sum_{m=1}^{M} \left[ I(v_m | \ldots | v_{m-1}) + \lambda_I I(v_m) + \lambda_C C(v_m) + \lambda_T T(v_{m-1}, v_m) \right]
\]

where \( \lambda_I, \lambda_C, \) and \( \lambda_T \) 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:

\[
I(w_i) = f_i \log \frac{F_A}{F_i}
\]

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).

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 \( L(v_m | \ldots | v_{m-1}) \) indicates goodness of word strings in a summarized sentence, and is measured by a trigram probability \( P(v_m | v_{m-2}v_{m-1}) \) [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

\[
C(w_{k,l}) = \log \frac{\alpha_k P_{ac}(w_{k,l}) P_{lg}(w_{k,l}) \beta_l}{G}
\]

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 \( w_{k,l} \);
- \( \alpha_k \) forward probability from the beginning node \( S \) to node \( k \);
- \( \beta_l \) 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 \( w_{k,l} \);
- \( G \) forward probability from the beginning node \( S \) to the end node \( T \) (\( = \alpha_T \)).
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:

$$
\alpha \rightarrow \beta \alpha \quad \text{(right-headed)}
$$

$$\alpha \rightarrow \alpha \beta \quad \text{(left-headed)}
$$

$$\alpha \rightarrow w
$$

where $\alpha$ and $\beta$ 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 $w_\alpha$ modifies $w_\beta$ 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) $w_1 \ldots w_k$ is derived from $\beta$; 3) $w_\alpha$ is derived from $\beta$; 4) $w_{k+1} \ldots w_j$ is derived from $\alpha$; and 5) $w_j$ is derived from $\alpha$.

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 DCFG. The probability that the $w_\alpha$ and $w_\beta$ 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 $w_\alpha$ and $w_\beta$ is denoted by $d(w_\alpha, w_\beta, i, k, j)$, where $i, k$ are the indices of the initial and final words derived from $\beta$, and $j$ is the index of the final word derived from $\alpha$.

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 $\beta$ can be directly connected with a head derived from $\alpha$ in a summarized sentence. In addition, the modifier can be also connected with each word which modifies the head. The word concatenation probability between $w_\alpha$ and $w_\beta$ is defined as a sum of the dependency probabilities between $w_\alpha$ and $w_\gamma$ and between $w_\alpha$ and each of $w_{y+1} \ldots w_z$. Using the dependency probabilities $d(w_\alpha, w_\gamma, i, k, j)$, the word concatenation score is calculated as a logarithmic value of the word concatenation probability given by

$$
T(w_{\gamma}, w_\alpha) = \log \sum_{i=1}^{x} \sum_{k=1}^{y-i} \sum_{j=1}^{z-i-y} \sum_{z=y}^{L} d(w_\alpha, w_\gamma, i, k, j). \tag{4}
$$

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

\[
\begin{align*}
\alpha & \rightarrow \beta \alpha \\
\alpha & \rightarrow P
\end{align*}
\]

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

\[
\begin{align*}
\alpha & \rightarrow \beta w \\
\alpha & \rightarrow w
\end{align*}
\]

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 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.
   \[
   \{C_{13}C_{13}\}, \{C_{13}C_{12}C_{13}\}
   \]
4) The final content word or content word sequence are attached to all function words.
   \[
   \{C_{13}F_{11}F_{12}\}, \{C_{13}C_{13}F_{11}F_{12}\}, \{C_{11}C_{12}C_{13}F_{11}F_{12}\}
   \]
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 \( P_z \) (\( 1 \leq x \leq z \leq M \)) is defined as \( d_{p}(P_x, P_z, m, l, n) \) by converting a word dependency probability as shown in Fig. 4 in Section 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

\[
T_{p}(P_x, P_y) = \log \sum_{m=1}^{x} \sum_{l=1}^{x} \sum_{n=y}^{x} d_{p}(P_x, P_z, m, l, n). \quad (5)
\]

Since Japanese sentences can be represented only by the rule of \( \alpha \rightarrow \beta \alpha \), the final phrase \( P_n \), in a phrase string \( P_1, \ldots, P_n \), derived from \( \beta \), is always derived from the same nonterminal symbol \( \beta \). The final phrase \( P_n \), in a phrase strings \( P_{n+1}, \ldots, P_n \) derived from \( \alpha \), is also derived from the same nonterminal symbol \( \alpha \). 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:

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

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:

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

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 \( P_{\text{ph}(w_x)} \) and \( P_{\text{ph}(w_y)} \) respectively, where \( \text{ph}(w) \) denotes an index of a phrase including a word \( w \). A word concatenation score of \( w_x \) and \( w_y \) within a phrase \( \text{ph}(w_x) = \text{ph}(w_y) \) is calculated using the intraphrase word concatenation rule \( R(w_x, w_y) = 0 \). On the other hand, the word concatenation score when \( w_x \) and \( w_y \) occur in different phrases \( \text{ph}(w_x) \neq \text{ph}(w_y) \) is calculated using a dependency probability between \( P_{\text{ph}(w_x)} \) and \( P_{\text{ph}(w_y)} \) 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:

\[
T(w_x, w_y) = \left\{ \begin{array}{ll}
T_p(P_{\text{ph}(w_x)}, P_{\text{ph}(w_y)}), & \text{if } \text{ph}(w_x) < \text{ph}(w_y) \\
\log R(w_k, w_l), & \text{if } \text{ph}(w_x) = \text{ph}(w_y).
\end{array} \right.
\]

**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:

The definition of symbols and variables:

- \( \langle s \rangle \) : beginning symbol of a sentence
- \( \langle f \rangle \) : ending symbol of a sentence
- \( P(w_n | w_k w_l) \) : linguistic score
- \( I(w_n) \) : word significance score
- \( C(w_n) \) : confidence score
- \( T(w_i, w_n) \) : word concatenation score
- \( s(k, l, n) \) : summarization score of each word
- \( m(w_l, w_n) \) : summarization score of a subsequence \( \langle s \rangle, \ldots, w_l, w_n \) consisting of \( m \) words, beginning from \( \langle s \rangle \), and ending \( w_l, w_n (0 \leq l < n \leq N) \)
- \( B(m, l, n) \) : back pointer

1. For \( m = 2 \) to \( M \)
   - For \( n = m \) to \( N - m + 1 \)
     - For \( l = m - 1 \) to \( n - 1 \)
       - \( g(m, l, n) = \max_{k<l} \{ g(m - 1, k, l) + s(k, l, n) \} \)
       - \( B(m, l, n) = \arg \max_{k<l} \{ g(m - 1, k, l) + s(k, l, n) \} \)

2. Select the optimal path
   The best complete hypothesis consisting of \( M \) words is decided by selecting the last two words \( (w_l, w_n) \).

   \[
   S(V) = \max_{N-M-1 \leq k \leq N-1} \{ g(M, l, u) + \log P(f \mid w_l w_n) \}
   \]

   \[
   (l, u) = \arg \max_{N-M-1 \leq k \leq N-1} \{ g(M, l, u) + \log P(f \mid w_l w_n) \}
   \]

3. 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
   \[
   v_m = w_n
   \]
   \[
   l' = B(m, l, n)
   \]
   \[
   \hat{n} = i
   \]
   \[
   l = l'
   \]

4. Initialization
   - Summarization score is calculated for each subsequence 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).

5. The DP process
   A dynamic programming recursion is applied for each pair of the last two words \( (w_l, w_n) \) of each subsequence hypothesis consisting of \( m \) words.
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 $M$ ($M < \sum N_j$) 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
   
   $s_j(k, l, n) = \log P(w_{jn} | w_k \cdots w_{jl}) + \lambda_t I(w_{jn}) + \lambda_C C(w_{jn}) + \lambda_T T(w_{jl}, w_{jn})$
   
   $g_j(m, l, n)$: local optimal score of $(s)$, $w_{j1}, \ldots, w_{jl}, w_{jn}$ consisting of $m$ words beginning with $(s)$ of the sentence $S_j$ and ending with $w_{jl}, w_{jn}$ in the sentence $S_j (0 < l < n \leq N_j)$

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

   $b_j(m, l, n) : \text{back pointer}$

   $B_j(m) : \text{back pointer of the end of a sentence}$

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 $n - 1$

   $g_j(m, l, n) = \max_{0 \leq k < l} \{ g_j(m-1, k, l) + s_j(k, l, n) \}$

   $b_j(m, l, n) = \arg\max_{0 \leq k < l} \{ g_j(m-1, k, l) + s_j(k, l, n) \}$

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

   $j = J$

   $m = M$

   $G_j(m, 0, n) = \begin{cases} G_{j-1}(m-1) + \log P(w_{jn} | (s)) + \lambda_t I(w_{jn}) + \lambda_C C(w_{jn}), & \text{if } 1 \leq n \leq N_j \\ -\infty, & \text{otherwise} \end{cases}$

   $b_j(m, 0, n) = \phi.$
while \( m > 0 \)

\[
\begin{align*}
\nu_m &= w_n \\
\ell' &= b_j(m, \ell, n) \\
n &= \ell \\
\text{if } \ell' \neq \phi & \text{ then} \\
\ell &= \ell' \\
m &= m - 1 \\
\text{else} \\
\nu_{m-1} &= (/s) \\
\nu_{m-2} &= (s) \\
(l, n) &= \mathcal{B}_{j-1}(m - 2) \\
m &= m - 3 \\
j &= j - 1
\end{align*}
\]

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:

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

where

- \( \text{Sub} \) number of substitution errors;
- \( \text{Ins} \) number of insertion errors;
- \( \text{Del} \) number of deletion errors;
- \( \text{Len} \) 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 rescued 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>
<thead>
<tr>
<th>Target</th>
<th>Symbol</th>
<th>Manual Transcription (TRS)</th>
<th>RDM</th>
<th>IL</th>
<th>ILT</th>
<th>SUBTRS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manual summarization</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Significance score (I)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>Linguistic score (L)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>Word concatenation score (T)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>Random word selection</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**TABLE II**

<table>
<thead>
<tr>
<th>Target</th>
<th>Symbol</th>
<th>Automatic Transcription (REC)</th>
<th>RDM</th>
<th>IL</th>
<th>ILT</th>
<th>SUBREC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manual summarization</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Significance score (I)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>Linguistic score (L)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>Confidence score (C)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>Word concatenation score (T)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>O</td>
<td></td>
</tr>
<tr>
<td>Random word selection</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**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 RLC.

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 summarization of automatically transcribed utterances at 70% summarization ratio was also evaluated (SUB, REC). To ensure 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) **Summary of Each Utterance**: Figs. 10–13 show summarization accuracy of both manual transcription (TRS) and automatic transcription (REC) at 40%, 60%, 70%, and 80% summarization 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.
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 automatic transcription (SUBTRS), it can achieve the performance comparable to the manual summarization of the recognition result (SUBREC).

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:

$$S \rightarrow P_1 \ldots P_m \ldots P_M$$

$P_m$ is defined as follows:

$$P_m = w_{m:c}w_{m,f_1}w_{m,f_2} \ldots w_{m,f,K_m}$$

where

- $M$ number of phrases in a sentence;
- $w_{m:c}$ content word of the $m$-th phrase;
- $w_{m,f,i}$ ith function word on $m$th phrase;
- $K_m$ number of functions words in $m$th phrase.

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

1) Initialization

$$a(\alpha \rightarrow \beta\alpha) = \text{given a flat probability and } b(\alpha \rightarrow \omega), e(\alpha \rightarrow \beta\omega_f) \text{ are given random values.}$$

2) Calculation for intra phrase forward probability

The probability of deriving $w_{m:e}w_{m,f_1} \ldots w_{m,f,i}$ from $\alpha$ in the $i$th phrase is calculated by the forward probability illustrated in Fig. 15(a):

$$h(m, i, \alpha) = P(\alpha \rightarrow w_{m:e}w_{m,f_1} \ldots w_{m,f,i})$$

$$= \begin{cases} b(\omega \rightarrow w_{m:c}), & \text{if } i = 0 \\ \sum_{\beta} h(m, i - 1, \beta)e(\alpha \rightarrow \beta w_{m:f_i}), & \text{if } i > 0. \end{cases}$$

(10)
3) Calculation of the interphrase inside probability

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

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

4) Calculation of the interphrase outside probability

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

\[
f(m, n | \alpha) = P(S \rightarrow P_1 \ldots P_{m-1} \alpha P_{n+1} \ldots P_M)
\]

5) Calculation of the interphrase backward probability

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

\[
r(m, i, \alpha)
= P(S \rightarrow P_1 \ldots P_{m-1} \alpha w_{m+1} w_{i+1} \ldots w_M, K_m | P_{m+1} \ldots P_M)
\begin{cases}
f(m, m, \alpha), & \text{if } i = K_m \\
\sum_{\beta} c(\beta \rightarrow \alpha w_i) r(m, i+1, \beta), & \text{if } i < K_m.
\end{cases}
\]
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

\[
\hat{a}(\alpha \rightarrow \beta \alpha) = \frac{\sum_{m=1}^{M-1} \sum_{n=m+1}^{M} \sum_{l=1}^{n-1} g(l, m, n; \alpha \rightarrow \beta \alpha)}{e(1, M | S)}
\]

(14)

\[
\hat{b}(\alpha \rightarrow \omega_c) = \frac{\sum_{m=1}^{M} \sum_{n=1}^{m} b(\alpha \rightarrow \omega_c)^r(m, 0, \alpha)}{e(1, M | S)}
\]

(15)

\[
\hat{c}(\alpha \rightarrow \beta \omega_f) = \frac{\sum_{m=1}^{M} \sum_{n=1}^{K} c(\alpha \rightarrow \beta \omega_f)^r(m, i, \alpha)}{e(1, M | S)}.
\]

(16)

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


