

erated, the method chooses important passages by judging from the *bushiness*¹ of a node (passage), depth-first path, and segmented bushy path in the network.

Fukumoto (Fukumoto 1997) proposed a method that first chooses sentences that contain a query term of user input and sentences which have a strong similarity to the previously selected sentences. As it decides to extract sentences one-by-one by comparing similarity, it does not consider the overall network topology of sentence similarity. A reader must give a query term to determine a point (sentence) where the extraction process starts. When the reader does not have an adequate knowledge of source documents, he or she may miss important sentences that have no connection with the query, or be at a loss for the query. As with Salton's method (Salton *et al.* 1997), it uses simple vector cosine distance for measuring sentence similarity; it neglects synonym relations.

Proposed Method

Against the background of these studies, we propose a novel extraction method that ranks sentences by spreading activation with the assumption that “*Sentences which are relevant to ones of significance are also significant*”. It produces a comprehensive summary even when a reader requires a short summary. Our method differs from some studies e.g., (Mani & Bloedorn 1999; Nagao & Hasida 1998) in that ours ranks sentences directly by spreading activation through sentence similarity; it does not require a deep analysis of original text. Our method also differs from others (Salton *et al.* 1997; Fukumoto 1997) by introducing refined similarity measure of sentences.

Sentence Similarity

Sentence extraction by spreading activation, as we detail later, requires similarity of sentences. Sentence similarity can be calculated from lexical relations between terms appearing in a sentence and others. When we estimate similarity of sentences, we must consider three problems: *how to estimate similarities of terms; how to identify the meaning of terms; and how to calculate sentence similarity from them.*

Estimation of term similarity For estimating similarity of terms, we use a Japanese lexical dictionary, *Nihongo Goi Taikai*² to take synonyms or other relations into consideration. Examining the semantic tree carefully, we notice that the number of terms that exist along the path from one term to another increases exponentially in proportion to path length. In other words, the relationship between two terms is inversely exponential to path length since the number of

¹The bushiness of a node on a graph is defined as the number of links connecting it to other nodes on the graph.

²NTT Communication Science Laboratories, Iwanami Shoten. *Nihongo Goi Taikai* consists of three sub-dictionaries, “lexical system”, “word system”, and “syntactical system”. The “noun lexical system” maps nouns into a tree structure which consists of 2,710 nodes that represent semantic attributes. Because the tree has the property that a node connotes semantic attributes of descendant nodes, we can estimate similarity of terms by the distance between terms on the semantic tree.

terms on the path increases exponentially. Hence, we should define similarity of two terms, t_i and t_j , by the exponential function,

$$\text{sim}(t_i, t_j) = \gamma^{\text{distance}(t_i, t_j)}, \quad (1)$$

where $\text{distance}(t_i, t_j)$ is the path length between the terms, and an attenuation factor γ ranges $0 < \gamma < 1$. We determine γ to be 0.5 vaguely, as similarity of two terms belonging to the same semantic attribute will be 0.5 since they do not always have a synonymous relation.

When t_i and t_j are identical, we define distance to be 0; $\text{sim}(t_i, t_i)$ will be 1, consequently. In cases where t_i and t_j are not identical, introducing a_i and a_j to represent attributes to which term t_i and t_j belong respectively, we define distance as the following.

$$\text{distance}(t_i, t_j) = \begin{cases} \text{length}_p(a_i, a_j) + 1 & (\text{length} < 4) \\ \infty & (\text{length} \geq 4) \end{cases} \quad (2)$$

$\text{length}_p(a_i, a_j)$ is the path length between nodes $\#a_i$ and $\#a_j$ on the semantic tree. In case either t_i or t_j has no entry in the dictionary, distance is defined as ∞ .

Sense disambiguation of terms Although a human can determine correctly and immediately the meaning of a term which has a number of meanings in the context of a text, computers do not have such ability. We can not calculate similarity of terms without identifying meanings. We formulate the word-sense disambiguation problem as follows.

We define $\mathbf{T} = (t_1, t_2, \dots, t_n)$ as a noun term which appears in a document. We introduce A_i to enumerate possible semantic attributes of term t_i , consulting the dictionary, *Nihongo Goi Taikai*. For example, for a word ‘system’, five attributes are found: #362 (organization), #962 (machine), #1155 (institution), #2498 (structure), #2595 (unit).

$$t_1 = \text{‘system’}, A_1 = \{362, 962, 1155, 2498, 2595\}. \quad (3)$$

When t_i has no entry in the dictionary (i.e. unidentified terms), we leave A_i empty.

Then, we choose a combination of $a_i \in A_i$ (i.e. search optimal $\{a_1, a_2, \dots, a_n\}$, where $a_1 \in A_1, a_2 \in A_2, \dots$, and $a_n \in A_n$) so that it maximizes the following *score*,

$$\text{score} = \sum_{k=1}^n \sum_{l=k+1}^n \min\{4 - \text{distance}(a_k, a_l), 0\}, \quad (4)$$

where $\text{distance}(a_i, a_j)$ is the same as in Equation (2). In other words, we determine an attribute of each term adopting lexical cohesion as context of original articles through optimization (Okumura & Honda 1994).

Calculation of sentence similarity For all pairs of sentences, we calculate similarity of sentences by the following formula,

$$\text{Sim}(S_i, S_j) = \sum_{t_i \in S_i} \sum_{t_j \in S_j} \frac{\text{sim}(t_i, t_j)}{\sqrt{|S_i||S_j|}}, \quad (5)$$

where $|S_i|, |S_j|$ are the numbers of indexing terms in sentences S_i, S_j , respectively. This formula counts up all possible lexical relations in inter-sentences and normalizes the sum by the geometrical mean to satisfy similarity of the same sentences to be 1.

