Term Weighting Method based on Information Gain Ratio for Summarizing Documents retrieved by IR systems

Tatsunori Mori Miwa Kikuchi Kazufumi Yoshida
Div. of Electrical and Computer Eng., Yokohama National University
79-5 Tokiwadai, Hodogaya, Yokohama 240-8501, Japan
{mori,miwa,kazu}@forest.dnj.ynu.ac.jp

Abstract

This paper proposes a new term weighting method for summarizing documents retrieved by IR systems. Unlike query-biased summarization methods, our method utilizes not the information of query, but the similarity information among original documents by hierarchical clustering. In order to map the similarity structure of the clusters into the weight of each word, we adopt the information gain ratio (IGR) of probabilistic distribution of each word as a term weight. If the amount of information of a word in a cluster increases after the cluster is partitioned into sub-clusters, we may consider that the word contributes to determine the structure of the sub-clusters. The IGR is a measure to express the degree of such contribution. We will show the effectiveness of our method based on the IGR by comparison with other systems.

1 Introduction

Information retrieval (IR) become widely used in daily life to search for a variety of information. One of the most popular type of services is search engine for documents on the Internet. Those systems usually show not only the titles of documents but also the small pieces of document, namely "summary." Such summary information is expected to be helpful for users to judge the relevance of each (original) document to users' information need. Therefore, the quality of summaries in IR tasks may be measured by the degree of consistency between the relevance judgment about summaries and that about original documents.

In generally, however, most of search engines adopt simple strategies like showing the first several sentences of documents, presenting several portions of document which include the keywords in queries. Quality of summaries generated such simple strategies is not usually enough for users to judge the relevance. We need more sophisticated summarization method to cope with the problem.

The most basic and main way of automatic document summarization is the extraction of important sentences, which is firstly proposed by Luhn[4]. A system based on this methods extracts important sentence and arranges the extracted sentences in the original order. The importance of each sentence may be calculated from combination of several factors, like importance of each word (e.g. frequency, clue words etc.), position of the sentence in the document, the role of sentence(e.g. title etc.), and so forth[5, 6]. Especially, the sentence extraction based on importance of words is one of primary ways to summarize documents.

The term frequency is widely used to sentence extraction, because it can be easily calculated within each document. However, in order to improve the quality of summaries, we have to consider not only such information, but also other types of information available in the process of summarization of retrieved documents. The mainstream of methodology, which adopts a part of such information, is the query-biased summarization[10]. The method of summarization uses user's query to give weight to the words or phrases in the query. Although the method which lays emphasis on queries is very intuitive and works well, there are a drawback that it does not use the information derived from the set of retrieved documents, which is expected to be an important clue for summarization as well as queries.

In this paper, we propose a novel way to utilize the information lying in the set of retrieved documents in order to summarize the documents. Unlike query-biased summarization methods, our method utilizes not the information of query, but the similarity information among original documents by hierarchical clustering. In order to map the similarity structure of documents into the weight of each word, we adopt the information gain ratio (IGR) of the probabilistic distribution of each word as a term weight. If the amount of information of a word in a cluster increases after the cluster is partitioned into sub-clusters, we may consider that the word contributes to determine the structure of the sub-clusters. IGR is a measure to express the degree of such contribution. We will show the effectiveness of our method based on IGR by comparison with other systems.

2 Term Weighting Method based on Information Gain Ratio

In contrast to general document summarization, summarization of documents retrieved by an IR system has the feature that the following extra information is given:

- A query,
- A set of documents to be summarized.

  Similarity among retrieved documents is expected to be higher than the average similarity among all documents, because they are supposed to be retrieved according to the relevance to the query.

Both of those types of information are expected to be good clues in summarization. In this section, we consider incorporating those types of information into the process of term weighting.

First choice is the query-biased summarization[10]. This methodology is based on the intuition that the words or phrases in the query express users' information needs directly, and summaries should include those expressions. Although it is very intuitive and works well, there are the following drawbacks:

- Since the expressions in the query are usually used as they are, efforts in search engines would not be reflected on summary. For example, feedbacks and query expansion modify the original query to improve effectiveness.
- Search engines retrieve not only the documents relevant to the query but also irrelevant documents. Since irrelevant documents scarcely contain the expression in the query, the summarization of such documents falls into the summarization of single document.

Therefore, we adopt second choice, namely, the term weighting method based on the information extracted from the set of retrieved documents. We expect that, if the quality of the result of IR is not so poor, the set of retrieved documents implicitly contains the information corresponding to the query, and the information can be extracted by some suitable way. However, it would be easily imagined that the effectiveness is not improved with simple methods like extracting words shared by almost all documents, because irrelevant documents are included in the set of retrieved documents and, moreover, the quality of retrieval depends on the IR system.

Based on the consideration above, we propose the scheme shown in Figure 1 which consists of the following steps:

1. Make a hierarchical clustering structure to obtain the information of the similarity among the retrieved documents.

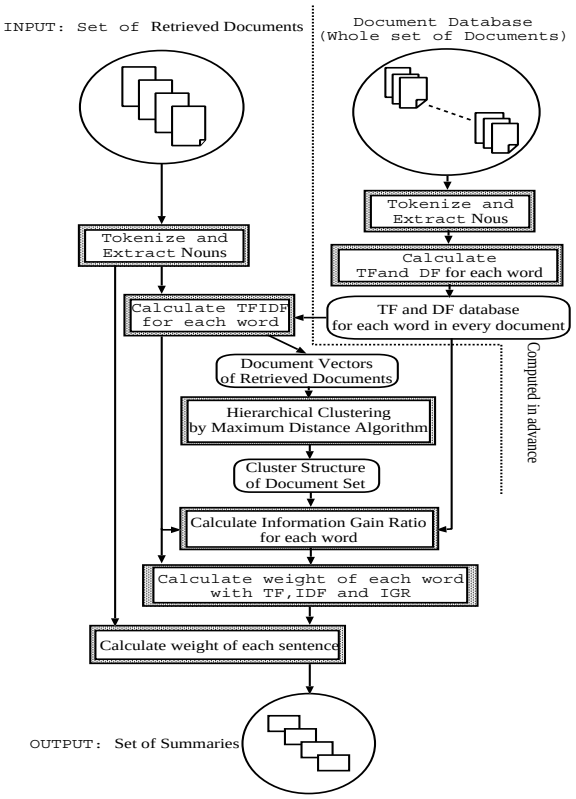

Figure 1. Overview of our scheme

2. Calculate the weight of each word according to the structure of document clusters and the probabilistic distribution of the word.

Through the step 1, it is expected that the set of retrieved documents are partitioned into clusters according to similarity, and documents relevant to query and irrelevant documents are separately organized into different clusters. We may obtain information to weight terms from the hierarchical structure of clusters. Note that we have to take account of the documents which are not retrieved but exist in the document database. By comparing those two types of documents, we can obtain the information what factors really contribute to retrieving the set of documents. Therefore, as shown in Figure 2 we introduce another layer of cluster, which corresponds to the set of the whole document database. The cluster consists of two sub-clusters. One sub-cluster is the cluster of retrieved documents, which will be partitioned into smaller clusters. The other one is the cluster of the rest of database.

The similarity structure given by document clustering describes the relations among documents. In order to generate summaries by important sentence extraction, we have to map the information about similarity among documents into the information about constituents of sentences like words. The step 2 makes the mapping.

In this paper, for the step 1, we adopt a hierarchical clustering of documents based on the similarity

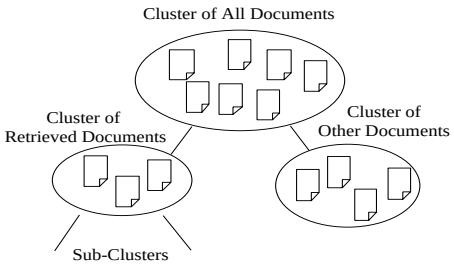

Figure 2. Clustering Retrieved Documents

of document vectors. Maximum distance clustering algorithm[11] is used to perform it. As for the step 2, we introduce a way to estimate the contribution of each word to deciding to partition a cluster into sub-clusters. It is based on a measure, called *information gain ratio* (IGR), about the probabilistic distribution of each word.

By combining the weight based on IGR with the term frequency (TF) and the inverse document frequency (IDF), we assign an composite weight to each word in documents. TF and IDF are typical measure of importance of words in the area of information retrieval and the area of summarization. Note that those three types of weight have different features as follows. Therefore, we expect that the combination of those weights is an overall weight suitable for summarization of retrieved documents.

- Term frequency of each word in a document(TF):

  is a weight which depends on the distribution of each word in documents. It expresses the importance of the word in the document.

- Information gain ratio of the probabilistic distribution of each word in partitioning a cluster(IGR):

  is the weight which depends on the structure of document clusters. It expresses the importance of a word in the cluster.

- Inverse document frequency of each word in the document database(IDF):

  is a weight which depends on the distribution of each word in the document database. It expresses the importance of each word in the document database.

2.1 Hierarchical Clustering by Maximum Distance Algorithm

In order to analyze the similarity among the retrieved documents, we need the definition of distance between two documents and the method to organize documents according to the similarity. Although there are many choices, we adopt the vector space model with the TFIDF term weighting to define distance of documents, and a hierarchical clustering method.

Among hierarchical clustering methods, the hierarchical agglomerative clustering method is commonly used. However, this method squeezes a cluster structure into a binary tree, and consequently discards the information of absolute value of distance between documents. Therefore we employ the maximum distance clustering method[11]. Although the method is originally a non-hierarchical algorithm, we recursively apply it to sub-clusters.

This recursive version of the method produces more general cluster structures, in which one cluster may have more than two sub-clusters according to distance among documents.

2.1.1 Distance between Documents

Based on the vector space model, we represent each document D_i as a point in an n-dimensional vector space $(weight_{i1}, weight_{i2}, \ldots, weight_{in})$, where $weight_{ik}$ is the weight assigned to the word w_k in D_i. In our experiment described later, we will take account of only nouns as (key)words. We adopt the TFIDF value of w_k for the weight $weight_{ik}$. The distance d between the documents D_i and D_j is defined as the following Euclidean distance:

$$
\begin{aligned}
d(D_i, D_j) &= \sqrt{\sum_k (weight_{ik} - weight_{jk})^2} \quad (1)\\
weight_{ik} &= tf(D_i, w_k) idf(w_k),\\
tf(D_i, w_k) &= \frac{freq(D_i, w_k)}{|D_i|},\\
idf(w_k) &= \log_2 \frac{N}{df(w_k)},
\end{aligned}
$$

where

$freq(D_i, w_k)$:	Frequency of the word w_k in D_i
$\|D_i\|$:	Number of morphemes in D_i
$df(w_k)$:	Document frequency of the word w_k
N :	Total number of documents in the database

In our experiment described later, we use JUMAN-3.61[3] to extract nouns from documents. The values of $df(w_k)$ and N are obtained from all of Mainichi Shimbun Newspaper articles in 1994, 1995, 1997 and 1998, which are target documents in our experiment.

2.1.2 Maximum Distance Clustering Algorithm

The maximum distance clustering algorithm firstly selects more than one cluster centers from the document set, then assigns other documents to the nearest cluster.

The main part of the algorithm is the selection process of cluster centers, which consists of the following steps.

1. Remove the most distant pair of documents from the document set DS and put them into the set of cluster centers C.

2. Calculate the distance d_{max} between the most distant pair of cluster centers in C.

3. For each document D_i in DS, calculate the distance $d(D_i, C)$ between D_i and the set of cluster centers as follows:

$$d(D_i, C) = \min_j d(D_i, C_j).$$

Select the most distant document D_d from the set of cluster centers as follows:

$$D_d = argmax_{D_d \in DS} d(D_d, C).$$

4. If $d(D_d, C) \geq \alpha \cdot d_{max}$, then put D_d into C, else terminate the procedure.

where α is a constant, and $0.5 \leq \alpha < 1.0$. In our experiment, α is set to 0.5.

Although the algorithm described above is originally one of the non-hierarchical clustering methods, we recursively apply it to sub-clusters to obtain hierarchical tree structure.

2.2 Clustering Documents according to Result of Retrieval

In order to get the factors of similarity in the set of retrieved documents, we need to compare the set of retrieved documents with the rest of document database. On the other hand, we do not need the information of similarity among documents which are not retrieved. Thus we obtain the structure of document cluster as shown in Figure 2 by the following two steps.

1. Introduce one cluster which has all documents in the document database, and partition it into two clusters. One is the cluster of retrieved documents. The other is the cluster of the rest of document database.

2. Apply the clustering algorithm recursively to the cluster of retrieved documents.

Through the steps, we obtain a tree structure, which represents the part-of relations between clusters. The root node and a leaf of the tree correspond to the document database and a retrieved document.

2.3 Term Weighting based on Information Gain Ratio

Each inner node of the tree of clusters represents the partition of a cluster. Each partition of cluster is performed based on the similarity among documents in the cluster. Therefore, we can map the information about similarity among documents into the weight of words in documents, if we introduce a method which reflects each structure of partition of cluster into the weight of each words.

As such a method, we propose a new method which consists of the following steps.

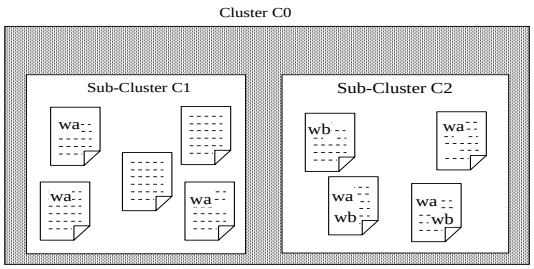

Figure 3. Word Distribution and Partitioning of Cluster

1. For each cluster, calculate the weight of each word according to the structure of its sub-clusters.

2. Since each document is specified by the series of partitions from the root node to a leaf of the cluster tree, the total weight of each word in a document is calculated by integrating the weights of each word for all of the partitions.

The step 1 is the most important part of our method. The basic idea is that we assign a higher weight to a word, if the word makes more contribution to determine the structure of the sub-clusters. In this paper, we measure the degree of contribution by the consistency between the distribution of a word and the partition of a cluster.

For example, let us consider partitioning the cluster into two sub-clusters in Figure 3. We suppose that the word w_b appears only in a sub-cluster C_1, on the other hand, the word w_a appears in both of two sub-clusters C_1 and C_2. In this case, we can conclude that the word w_b has more contribution to determine the partition than w_a because we can select a specific cluster by examine whether the word w_b appears.

2.3.1 Information Gain Ratio

In this section, we propose the utilization of the IGR to represent the degree of the consistency between the distribution of a word and the partition of a cluster. The IGR is the measure used in the decision tree leaning algorithm C4.5 to select the best attribute to be tested[7]. It represents how precisely the test by the attribute predict the distribution of classes. By regarding a cluster structure of documents as a decision tree, we may use the IGR under the correspondence shown in Table 1.

The information gain ratio $gain\_r(w, C)$ of the word w in the cluster C is calculated as follow:

$$\begin{aligned} gain\_r(w, C) &= \frac{gain(w, C)}{split\_info(C)} \\ gain(w, C) &= entropy(w, C) - entropy_p(w, C) \\ entropy(w, C) &= -p(w|C) \log_2 p(w|C) \end{aligned} \quad (2)$$

Table 1. Comparison of our method with the decision tree leaning algorithm C4.5

Our method	C4.5
Partition of a cluster	Test by an attribute
Probabilistic distribution of a word	Probabilistic distribution of classes

$$
\begin{aligned}
 & -(1-p(w|C))\log_2(1-p(w|C)) \\
p(w|C) &= freq(w,C)/|C| \\
entropy_p(w,C) &= \sum_i \frac{|C_i|}{|C|} entropy(w,C_i) \\
split\_info(C) &= -\sum_i \frac{|C_i|}{|C|}\log\frac{|C_i|}{|C|} \\
freq(w,C) &= \text{Frequency of the word } w \text{ in } C \\
C_i &: \text{The } i\text{-th sub-cluster of } C \\
|C_i| &: \text{Number of words in } C_i
\end{aligned}
$$

The *information gain* $gain(w,C)$ is the amount of decrease of the entropy about the probabilistic distribution of the word w. The *split information* $split\_info(C)$ is the entropy about partitioning the cluster C. The IGR $gain\_r(w,C)$ is defined as the ratio of the information gain to the split information.

2.3.2 Weighting Terms based on Information Gain Ratio

For each word in every document, we can collect a set of IGR values by pursuing the path in the cluster tree from the root node to the leaf corresponding to the document. There would be several ways to use the set of IGR values according to the design of the user interfaces.

For instance, let us consider the interactive user interface where the system shows the user the structure of a cluster and then the user selects sub-cluster(s) by referring to the the summary of each sub-cluster. In this case, the weight of each word can be calculated based on the IGR at the cluster.

On the other hand, we need to integrate the set of IGR values into a value if we adopt a list-style user interface where all summaries of retrieved documents are shown to the user at once. There would be several ways of integration, e.g., summation of all values, product of all values, the maximum value, the value of the cluster in a certain depth, and so on.

In this paper, we suppose the list-type user interface and adopt the summation shown in (3) and Figure 4. This integration method take account of every IGR equally.

$$
\begin{aligned}
igr(w,D) &= \sum_{C\in Cset(D)} gain\_r(w,C) \qquad (3) \\
Cset(D) &= \text{the set of all clusters to which the document } D \text{ belongs}
\end{aligned}
$$

With this weight $igr(w,D)$, we define the weight $weight(w,D)$ of the word w in the document D. As

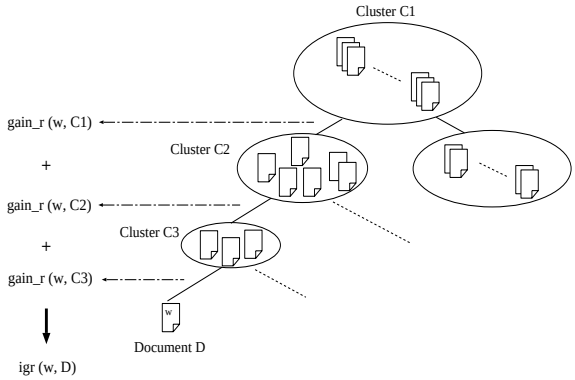

Figure 4. Weighting Terms by Information Gain Ratio

described before, we suppose that the effective weight of word should be the combination of three types of fundamental weight, TF, IDF and IGR. Therefore we use the product of those three values as the weight of word.

$$
weight(w,D) = igr(w,D)\cdot tf(w,D)\cdot idf(w) \qquad (4)
$$

3 Evaluation

In this section, we will show the experimental result of our system in the IR task of NTCIR2 Text Summarization Challenge(TSC).

3.1 Summarization by Extracting Important Sentences

Since the term weighting is the most basic component of summarization methods, we expect that our weighting method can be integrated into various summarization schemes. However, our aim is to show that our weighting method is effective in summarizing retrieved documents. Therefore, we use the most fundamental scheme of summarization which is only based on the term weighting. The scheme consists of the following steps.

1. Let the importance $s\_imp(s,D)$ of the sentence s in the document D be the average weight of

keywords in the sentence. That is,

$$s\_imp(s, D) = \frac{\sum_{w \in keyw(s)} weight(w, D)}{|keyw(s)|} \quad (5)$$

$$keyw(s) = \text{the bag of keywords in the sentence } s.$$

2. Extract sentences with higher importance from the original document, until the total length of selected sentences exceeds a certain predetermined length of summary.

3. Put the selected sentences in order of original document to obtain the summary.

Our experiment is performed under the following conditions:

- Keywords are nouns.
- When the summaries are shown in a list at once, uniformity of the length of each summary is expected to increase readability. Therefore, we use not a certain compression ratio but a cutoff length. The cutoff length is 150 words in our experiment.
- If the original document is shorter than 150 words, the system does not perform summarization and returns the original document.
- In generating summaries, the system inserts '$\cdots$' to indicate that the omission is at that point. The system also inserts a 'newline' at the end of paragraph.

3.2 Experimental Result of Summarization for IR Tasks

We will evaluate the effectiveness of our term weighting method with the result of summarization for IR tasks in NTCIR2 TSC. The data set distributed by TSC committee has 12 topics. Each topic has one query and 50 retrieved documents. Those documents are retrieved from the data base of Mainichi Shimbun Newspaper articles in 1994, 1995 1997 and 1998.

Every participant made a summary for each document with his/her system and submitted 600 summaries to the TSC committee. TSC committee evaluated the summaries by presenting the queries and the summaries to 36 subjects(36 students). Three subjects were assigned to one topic and they judged the relevance between the query and a summary. The quality of summaries are evaluated by comparing subjects' relevance judgments for summaries and the relevance judgment for the original documents, which is carefully assigned by TSC committee. If those two relevance judgments are highly consistent with each other, we may conclude that the system is very effective in summary generation for retrieved documents.

The relevance of each original document is graded either 'A(relevant)', 'B(related)' or 'C(not relevant)'.

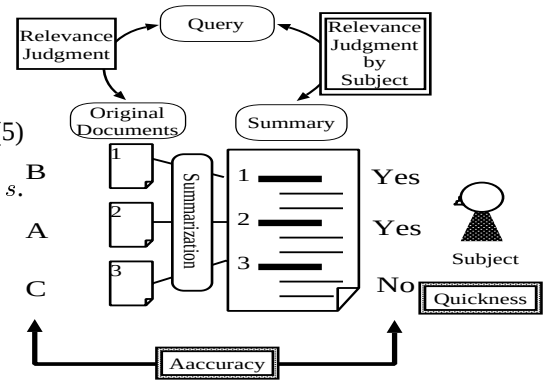

Figure 5. Evaluation of Summaries in IR task

On the other hand, each subject answers the question about the relevance of each summary with a simple answer, 'Yes' or 'No'. Therefore, we can consider the following criteria for evaluating consistency between the relevance judgment of original documents and subjects' judgment of summary.

- Answer Level A:

  documents of the grade A are regarded as 'relevant'.

- Answer Level B:

  documents of either the grade A or the grade B are regarded as 'relevant'.

The experimental result of our system in TSC is shown in Table 2 along with the results of other participating systems and the baseline systems. In this table, the following measures are used for evaluation.

- Average time to accomplish the relevance judgment of one task (50 summaries).
- Average length of Summaries.
- Averages of Recall, Precision and F-measure about relevance judgment.

4 Discussion

In this section, we evaluate the effectiveness of our method by comparing it with eight other participating systems and three baseline systems shown in Table 2. At this moment, the detail of other participating systems are not given. Three baseline systems are named 'Fulltext', 'TF' and 'Lead' in the table. The system of the method 'Fulltext' just returns the orignal documents. Thus, the compression ratio is 100%. The system 'TF' generates summaries with TF-based sentence extraction. The weight of words in the query is doubled and the compression ratio is 20%.
The system 'Lead' returns the lead of a document as as a summary and the compression ratio is 20%.

Table 2. Experimental Result in TSC

	Our System	Sys 1	Sys 2	Sys 3	Sys 4	Sys 6	Sys 7	Sys 8	Sys 9	Fulltext	TF	Lead
Recall (Ans. A)	**0.907**	0.833	0.899	0.793	0.818	0.858	0.831	0.824	0.849	0.843	0.798	0.740
Precision (Ans. A)	0.751	0.728	0.717	0.685	0.674	0.718	0.739	0.738	0.741	0.711	0.724	**0.766**
F-Measure (Ans. A)	**0.808**	0.761	0.785	0.715	0.718	0.763	0.766	0.749	0.768	0.751	0.738	0.731
Recall (Ans. B)	0.754	0.741	**0.793**	0.715	0.737	0.745	0.719	0.719	0.752	0.736	0.700	0.625
Precision (Ans. B)	0.897	0.921	0.904	0.898	0.875	0.892	0.908	0.913	**0.923**	0.888	0.913	0.921
F-Measure (Ans. B)	0.797	0.808	**0.828**	0.776	0.773	0.785	0.779	0.775	0.805	0.773	0.776	0.712
TIME	8:33	9:41	12:48	**6:25**	6:44	9:01	10:16	9:16	9:31	13:46	8:44	7:32
LENGTH	234.4	297.8	585.7	**89.5**	136.4	288.4	292.9	266.1	262.5	819.4	253.6	174.5

Sys 1 to 9: Other participating systems.
Ans.A, Ans. B: Answer Level A and Answer Level B
Fulltext: The system which just returns the orignal documents.
TF: The system which generates summaries with TF-based sentence extraction. The weight of words in the query is doubled. Compression ratio is 20%.
Lead: The system which returns the lead of document. Compression ratio is 20%.

In the summarization for retrieved documents, it is important to improve both of the accuracy of the judgments of relevance and the time required to make the judgments, simultaneously. On the other hand, there is a trade-off relation between them. For example, if a longer summary is shown to a user, the required time would become longer but the judgment would be more accurate. Although we need a certain measure which integrates them into one appropriate value, no good measure has been proposed so far.

Therefore, we evaluate them separately. Firstly, we will beliefly examine the time required to make the judgments, then we will consider the accuracy of the tasks.

4.1 Time required to Make Judgments

The average time for the relevance judgment of our summaries is 8:33 (8 minutes and 33 seconds) per topic. It is in third position among all participating systems, and the average time of all participating systems is 9:08 per topic. The average time of our system is shorter than the average of all participating systems. Since the summaries generated by our system are relatively shorter than the others, we do not consider the time for judgment in the following discussion about accuracy and we directly compare the values of evaluation measures.

4.2 Accuracy of Performance of Task

4.2.1 Answer Level A

In this section, we consider the evaluation of 'Answer Level A'. Our system outperformed other participating systems in terms of all of measures, the average precision, the average recall and the average F measure. Although the precision of the 'Lead' method is 1.5 point higher than our system, our system outperforms all baseline systems in other measures.

The F measure of 'Lead' method is 7.7 point lower than our system, because the precision of the method is the lowest. The 'Lead' can be regard as the method in order to attach importance to precision.

In comparison with 'TF' method, our system is 10.9 point higher in the recall, 2.7 point higher in the precision and 7.0 point higher in the F-measure. It would shows that we can make effective summaries with retrieved documents even if we do not use the information of query.

From the consideration described above, we may conclude that our weighting method of word is very effective for the summarization of the retrieved documents.

4.2.2 Answer Level B

In this section, we consider the evaluation of 'Answer Level B'. Since the number of relevant documents increases in 'Answer Level B', the precision grows and the recall decreases. If a system has high precision in 'Answer Level A,' the recall in 'Answer Level B' will remarkably fall. On the other hand, a system will gain in precision if the main cause of error in 'Answer Level A' is that the document of the relevance level B is judged as relevant.

Although the recall of our system decreases from 0.907 to 0.754, our system is still in second place among participating systems. Thus, our system generate more summaries which are judged correctly as relevant than other systems. The first-ranking system generates longer summaries and the average time per one task (12:48) was also longer than our average time 8:33.

On the other hand, the precision does not grow as other systems do and is degraded to the seventh place. It shows that our system generates more inappropri-

ate summaries, which are originally the grade-C documents but are judged as relevant, than other systems.

From the consideration described above, we may conclude that our system can be regarded as the method to attach importance to recall.

5 Related Works

As described in Section 2, summarization of documents retrieved by an IR system has the features that the following types of extra information is given:

1. A query,
2. A set of documents to be summarized.

Although we only use the information of (2) in this paper, there, of course, are several proposals of utilization of the information (1). This type of method is called query-biased summarization. Tombros et al.[10] and Shiomi et al.[8] independently propose the method to give the higher weight to the terms in queries and confirm the effectiveness of it. Carbonell et al.[1] introduce a notion called 'Maximum Marginal Relevance' for re-ordering retrieved documents and producing summaries so as to minimize redundancy according to the similarity between documents and queries. Although those methods, which use queries directly, are very intuitive and works well, there are the drawbacks as described in Section 2.

In the same way as ours, Eguchi et al.[2] and Fukuhara et al.[9] use the information (2). Eguchi et al.[2] propose an IR system based on some kind of relevance feedback. The system partitions the set of retrieved documents into clusters. Then, it represent a "summary" of each cluster in order for the user to select a relevant cluster and feedback it to the system. The summary contains the title of representative document and the keywords, which appear frequently in the cluster. The system proposed by Fukuhara et al.[9] also makes clusters of retrieved documents, then, extracts topic words in terms of the notions of 'skewness' and 'kurtosis'. The summaries are generated by linking up sentences which have relevant topics.

Both of those systems uses the clustering of documents only to find groups of similar documents. On the other hand, we use the structure of clusters more effectively in order to weight terms.

6 Concluding Remarks

In this paper, we proposed a novel way to utilize the information lying in the set of retrieved documents in order to summarize the documents. Our method utilizes the similarity information among original documents by hierarchical clustering. In order to map the similarity structure of documents into the weight of each word, we adopt the information gain ratio of the probabilistic distribution of a word as a term weight. In the experiments of TSC, we showed that our term weighting method is very effective in summarization of retrieved documents.

In future work, we plan to investigate the utilization of our IGR-based term weighting method in an interactive user interface of IR. In this paper, every part of cluster structure is uniformly reflected in the weight of each term. On the other hand, as an interactive user interface of IR, we can imagine a system where the user selects one sub-cluster recursively to reach a desired document. In this case, words may be weighted according to the cluster structure presented to the user.

References

[1] J. Carbonell and J. Goldstein. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. In *Proceedings of the 21st Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 335–336, 1998.

[2] K.Eguchi, H.Ito, A.Kumamoto, and Y.Kanata. Adaptive Query Expansion Based on Clustering Search Results. *Transaction of Information Processing Society of Japanese*, 40(5):2439–2449, 1999.

[3] T. Kurohashi and M. Nagao. *Japanese Morphological Analysis System JUMAN version 3.61 Manual*. Kyoto University, 1998. (in Japanese).

[4] H. Luhn. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159–165, 1958.

[5] M. Okumura and H. Nanba. Automated text summarization: A survey. *Journal of Natural Language Processing*, 6(6):1–26, 1999. (in Japanese).

[6] M. Okumura and H. Nanba. Recent advanced in automated text summarization. Technical Memorandum IS-TM-2000-001, School of Information Science, Japan Advanced Institute Of Science and Technology, 7 2000.

[7] J. R. Quinlan. *C4.5: progarms for machine learning*. Morgan Kaufmann Publishers, May 1993.

[8] T. Shiomi, K. Tokuda, S. Aoyama, and K. Kakigahara. An Abstraction Method Using Viewpoints. In *Proceedings of The 56-th Annual Meeting of Information Processing Society of Japan*, volume 3, pages 104–105, 1998. (in Japanese).

[9] T.Fukuhara, H.Takeda, and T.Nishida. Multiple-text Summarization for Collective Knowledge Formation. In *Proceedings of Workshop on Social Aspects of Knoledge and Memmory*. IEEE Systems, Man and Cyberbetics Conference, 1999.

[10] A. Tombros and M. Sanderson. Advantages of Query Biased Summaries in Information Retrieval. In *Proceedings of the 21st Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 2–10, 1998.

[11] J. T. Tou and R. C. Gonzalez. *Pattern recognition principles*. Addison-Wesley Pub. Co., 1974.