Results of CRL/NYU System at DUC-2003 and an Experiment on Division of Document Sets

Chikashi Nobata† and **Satoshi Sekine**‡
† Communications Research Laboratory
3-5 Hikari-dai, Seika-cho, Soraku-gun, Kyoto 619-0289 Japan
nova@crl.go.jp
‡ Computer Science Department, New York University
715 Broadway, 7th floor, New York, NY 10003 USA
sekine@cs.nyu.edu

Abstract

We participated in three multi-document summarization tasks at the DUC-2003 formal run and evaluated the performance of our summarization system. Our summarization system based on sentence extraction also incorporated a module to estimate similarity between sentences for multi-document summarization. The similarity information was used for selecting the representative sentence among similar sentences. We also conducted a preliminary experiment to divide document sets into two groups corresponding to the distribution of key sentences.

1 Introduction

We have developed our summarization system for the single document summarization at DUC-2001. Our system is based on sentence extraction technique, which has been used to summarize documents with various features (Edmundson, 1969; Kupiec et al., 1995; Watanabe, 1996; Aone et al., 1998; Nomoto and Matsumoto, 1997). For multi-document summarization tasks in DUC-2003, we have modified our system and also incorporated another module to estimate similarity between sentences.

2 System overview

In this section, we briefly explain scoring functions used for extracting key sentences, which are revised forms of those used in our previous system at DUC-2001. The full description of our previous system is presented at (Nobata et al., 2002). We also introduce an added module to measure sentence similarity.

2.1 Scoring function

Our system uses several types of metrics to estimate the significance of sentences. Each metric is explained below.

2.1.1 Sentence position

Our system has a function that uses sentence position to establish the significance of sentences. In this function, three methods are used to handle sentence position. The first is to give a score of 1 to the first N sentences and 0 to the others when N is given as a threshold for the number of sentences of the summary. That is, the score of the ith sentence (S_i) is:

$$
\begin{aligned}
\mathit{Score}_{\mathrm{pst}}(S_i)(1 \le i \le n) &= 1 \quad (\text{if} \quad i < N) \\
&= 0 \quad (\text{otherwise})
\end{aligned}
$$

where n is the number of sentences in a given document. The second method is to give a score that is the reciprocal of the sentence position, and the third method is a modified version of the second in that it checks the sentence position from the end of the document as well as from the beginning. The first method performed best at the training stage.

2.1.2 Sentence length

The second scoring function uses sentence length to establish the significance of sentences. The length here means the number of words in the sentence. Two methods are defined here. The first returns the length of each sentence (L_i) relative to the maximum length of the sentence. The second method sets the score to a negative value as a penalty when a sentence is shorter than a certain length (L_{min}):

$$
\begin{aligned}
\mathit{Score}_{\mathrm{len}}(S_i) &= 0 \quad (\text{if} \quad L_i \ge L_{min}) \\
&= \frac{L_i - L_{min}}{L_{min}} \quad (\text{otherwise}).
\end{aligned}
$$

Since we set L_{min} to 10, a sentence with 10 or fewer words received a penalty score. The second method performed better at the training stage.

2.1.3 Tf*idf

The third scoring function is based on term frequency (*tf*) and document frequency (*df*). The score of a sentence (S_i) is the average of the tf*idf scores of each word (w) in the sentence:

$$\mathit{Score}_{\text{tf*idf}}(S_i) = \frac{1}{|S_i|} \sum_{w \in S_i} \text{tf*idf}(w)$$

We have three different scoring functions for tf*idf, and the following method performed best at the training stage:

$$\text{tf*idf}(w) = \frac{\mathit{tf(w)}\text{-}1}{\mathit{tf(w)}} \log \frac{\mathit{DN}}{\mathit{df(w)}}$$

where *DN* is the number of given documents. We used all documents in the Wall Street Journal in 1994 and 1995 to count document frequencies.

2.1.4 Headline

The fourth scoring function is to use the headline of a document to establish the significance of sentences. The basic idea is that the greater the number of words in a sentence that match those in the headline, the more important the sentence is likely to be. This function estimates the relevance between a headline (H) and a sentence (S_i) using the tf*idf values of words (w) (except for the stop words) in the headline.

We also evaluated this scoring function using only named entities (NEs) instead of the nouns. An NE tagger developed by NYU Proteus group performed NE tagging, based on extended named entity categories (Sekine et al., 2002). For NEs, only the term frequency was used because we expected the document frequency for entities (e) to usually be quite small, thereby making the difference between entities negligible. The second method performed better at the training stage, and the equation is as follows:

$$\mathit{Score}_{\text{hl}}(S_i) = \frac{\displaystyle\sum_{e \in H \cap S_i} \frac{\mathit{tf(e)}}{\mathit{tf(e)}+1}}{\displaystyle\sum_{e \in H} \frac{\mathit{tf(e)}}{\mathit{tf(e)}+1}}$$

2.1.5 Query

The fifth scoring function is to use additional information about each document set such as topics in Task 2, viewpoints in Task 3, and questions in Task 4. Here we call these information 'Queries.' These queries were used in the similar way to the headline, but the method with tf*idf values of words was used for queries. The equation of the method is as follows:

$$\text{tf*idf}_q(w) = \frac{\mathit{tf(w)}}{\mathit{tf(w)}+1} \log \frac{\mathit{DN}}{\mathit{df(w)}}$$

$$\mathit{Score}_q(S_i) = \frac{\displaystyle\sum_{w \in Q \cap S_i} \text{tf*idf}_q(w)}{\displaystyle\sum_{w \in Q} \text{tf*idf}_q(w)}$$

Since we have query information only in TREC Novelty track data at training stage, we applied the same parameter weight for the headline scoring function to the query at the formal run.

2.2 Parameters

Our system uses parameters to integrate the results of each scoring function in order to calculate the total score for a sentence. The total score for a sentence (S_i) is determined using a scoring function ($\mathit{Score}_j()$) and a parameter (α_j) as follows:

$$\text{Total-Score}(S_i) = \sum_j \alpha_j \mathit{Score}_j(S_i)$$

Our system calculates a score for all of the sentences and sets the ranking of each sentence in descending order of score. The order of the extracted sentences is the same as in the original documents when the system outputs a summary.

After the range of each parameter was set manually, the system changed the values of the parameters within the range and performed a summarization based on the training data. Each score was recorded whenever the parameter values were changed, and the parameter values resulting in the best score were stored.

The training data we used at the DUC-2003 formal run are DUC-2001, DUC-2002, and TREC Novelty track data. We used 30 document sets (299 documents) from DUC-2001 training data, 58 document sets (556 documents) from DUC-2002 extracts data, and 49 document sets (1087 documents) from TREC Novelty track data. As for DUC-2001 data, our group created our own extract data manually. Since DUC-2002 data and TREC Novelty track data include extracts, we used them for setting parameters.

Table 1 shows the contribution of each feature that was the basis of a scoring function. The greater the standard deviation is, the greater effect the scoring function has for the score of each sentence, and our system multiplies values of each scoring function by given parameter weights to calculate the score of each sentence. The value of contribution was therefore the product of the optimized parameter weight and the standard deviation of the score. The parameters were normalized by the norm of all pa-

Table 1: Contribution of features

Feature	Contribution
Position	0.507
Length	0.004
tf*idf	0.488
Headline	0.001

rameters; i.e.

$$\sum_j \alpha_j = 1.$$

We can see that features influential in sentence scores are that of sentence position and tf*idf. Contribution from the other features is relatively small.

2.3 Similarity between sentences

We added a module to estimate the similarity between sentences, which are used to select one key sentence from among semantically similar sentences. The system calculates the Dice coefficient as a similarity measure based on the number of words between two sentences S_x and S_y. For the words $\{x_i|x_i \in S_x\}$ and $\{y_j|y_j \in S_y\}$, the similarity between S_x and S_y is calculated as

$$Dice(S_x, S_y) = \frac{2\sum_{x_i=y_j} ave(f(x_i), f(y_j))}{\sum_i f(x_i) + \sum_j f(y_j)}.$$

The function $f()$ gives the weight for each word. Three types of weight can be selected:

Binary: if the word appears in the sentence, the weight is set to 1. Otherwise, the weight is set to 0.

Tf: the tf value of the word.

Tf*idf: the tf*idf value of the word.

The system uses one of the weights to calculate similarities. We used the Dice coefficient with the binary weight at the DUC-2003 formal run, because we didn't observe any significant difference among the weights in the experiment using the training data.

Sentence pairs that have a coefficient value higher than a threshold are regarded as similar to each other. The value of the threshold is set to 0.5 at the training stage.

The obtained similarity values are used in selecting only the one sentence that has the highest sentence score among similar sentences. When this sentence is selected, other sentences similar to the selected one are discarded. This is intended to remove redundancy from the generated summary. For example, in documents about a criminal case, the established facts of the case are repeatedly described. Such repetition should be removed.

Table 2: Evaluation results of multi-document summarization task at DUC-2003

	Coverage	MLAC	MNQQ
Task 2			
System	0.262 (4/16)	0.166 (7/16)	2.333 (9/16)
Ave.	0.217	0.147	2.379
Task 3			
	Coverage	MLAC	MNQQ
System	0.199 (1/11)	0.123 (3/11)	2.100 (7/11)
Ave.	0.160	0.110	2.315
Task 4			
	Coverage	MLAC	MNQQ
System	0.138 (7/9)	0.085 (7/9)	1.667 (3/9)
Ave.	0.157	0.105	1.962

3 Evaluation results

In this section, we show the evaluation results of our system at three multi-document summarization tasks in DUC-2003. The overview of our evaluation is shown in Table 2. 'System' rows correspond to our system's result, and 'Ave.' rows the average of evaluation results among all participants. The ranks of our system out of participants are also shown in parenthesis with each evaluation result.

Our system was good in Coverage and Mean Length-Adjusted Coverage (MLAC) at Task 3, and in Mean Number of Quality Questions (MNQQ) at Task 4. Table 3 shows the average of answer categories of each quality question. We can see that the answer categories of Q1 has large impact to degrade results of our system in every task. It seems strange considering that Q1 is the question about capitalization error, and our system didn't include a headline into an output summary. One possible reason is that uppercased words in the first sentence of a document, such as a location name at which the document was written and a name of a news agency, were not removed in an output summary.

4 Experiment on division of document sets

For multi-document summarization, suitable summarization method for a given document set varies according to the characteristic of the document set. For example, summarizing a document set related with a single event is considered different from summarizing a document set related with multiple events. McKeown et al. (McKeown et al., 2001) reported classification of document sets by the contents for summarization at DUC-2001. Classifying document sets based on the contents, however, did not guarantee improvement of automatic summarization.

In this section, we report our ongoing experiment for dividing document sets into two groups from a different

Table 3: Average answer categories of quality questions

Q#	Task 2 Sys.	Task 2 Ave.	Task 3 Sys.	Task 3 Ave.	Task 4 Sys.	Task 4 Ave.
Q01	1.200	0.394	1.033	0.361	0.500	0.226
Q02	0.033	0.144	0.033	0.161	0.000	0.052
Q03	0.033	0.121	0.000	0.133	0.000	0.033
Q04	0.033	0.202	0.067	0.242	0.100	0.216
Q05	0.267	0.208	0.300	0.233	0.167	0.201
Q06	0.000	0.044	0.067	0.055	0.033	0.015
Q07	0.033	0.160	0.033	0.188	0.067	0.130
Q08	0.133	0.344	0.100	0.355	0.133	0.454
Q09	0.033	0.044	0.000	0.018	0.000	0.011
Q10	0.000	0.029	0.000	0.045	0.000	0.030
Q11	0.300	0.252	0.167	0.118	0.133	0.100
Q12	0.500	0.652	0.567	0.636	0.700	0.588

viewpoint. We tried to classify document sets based on how these should be summarized so that the classification could be helpful for improving automatic summarization.

In the experiments of DUC-2003, we found that there were two main groups of distributions of key sentences in document sets. One group is that most of key sentences are in the beginning of each document, the other is that key sentences are scattered among the documents. Let G_1 the former group of document sets, and G_2 the latter group of document sets in the following description.

Figure 1 and Figure 2 show the distributions of key sentences in two document sets from DUC-2002 extracts data. Each row corresponds to each document in a given document set, and '`o`' indicates an extracted sentence, '`-`' indicates the other sentence. These figures show that how the distributions of key sentences can be different among document sets. Classifying document sets into these two groups could be useful for improving the performance of sentence extraction, because the feature of sentence position, often used in sentence extraction, is effective for the former group of document set but not for the latter. We found that document sets related with multiple events at different places such as ship sinking and coal miner's strikes were mostly type of Figure 1. On the other hand, document sets related with a single event such as a particular case of shooting spree are mostly type of Figure 2.

4.1 Division of document sets

To divide given document sets, we set a heuristic index which connotes how much key sentences in a document set are gathered in the beginning of each document. When a document set that contains (D) documents and has (K) key sentences, the index (I) is defined as follows:

$$I = \max(\frac{k}{D}, \frac{k}{K})$$

where:

Table 4: Division of document sets

Data	G_1	G_2	Total
DUC-2001	20	10	30
DUC-2002	13	45	58
Novelty	23	26	49
Total	58	79	137

k = the number of key sentences appeared within the first N sentences of a document
N = the range of sentence position regarded as the beginning of the document

We use the index (I) to divide given document sets. When a document set d is given, the division for d is selected by a threshold T for the index I:

$$\begin{aligned} d &\in G_1 \text{ (if } I > T) \\ &\in G_2 \text{ (otherwise)} \end{aligned}$$

The value of the threshold T was set to 0.3 for the data we used at DUC-2003 to train our summarization system. The number of document sets at each division is shown in Table 4.

4.2 Features for automatic division of document sets

We used a time span and NEs in a document set as features for automatic division of document sets. We used the following six NE classes here: Event, Facility, Location, Organization, Person, and Product.

Timespan of the document set: We expected that a time span was related with the distribution of key sentences. If a time span of a document set is wider, the document set is likely to be in G_1 as it is likely to have new fact in the beginning. Whereas a time span of a document set is short, similar information is often repeated in the beginning of documents and new information is likely to be scattered, so that the document set is likely to be in G_2. Timespan throughout the given document set is calculated from the document ID and the dateline of each document. The timespan information t is converted to binary features, such as:

Is_within_week: 1 (if $t \leq 7$) or 0 (otherwise)
Is_within_month: 1 (if $7 < t \leq 31$) or 0 (otherwise)
Is_within_year: 1 (if $31 < t \leq 365$) or 0 (otherwise)
Is_over_a_year: 1 (if $t > 365$) or 0 (otherwise)

Frequency and document frequency of NE tokens: The frequency and document frequency of the most frequent word at each NE class are given as features, because we expected that the presence of the salient

```
TOPIC: d075be.200e
               0    5     10    15    20    25    30    35    40    45
 AP880428-0041 :o----:-----:-----:-----:-----:
 AP880818-0088 :-----:-----:-----:-----:--
 AP880829-0222 :-o---:-----:-----:-----:-----:-----:-----:--
 AP881115-0113 :-----:-----:-----:-----:-----:-----:
 AP890115-0014 :o----:----o:-----:-----:-----:-----:--
 AP900322-0112 :-----:-----:-----:-----:-----:-----:
 AP900705-0149 :o----:-----:-----:---
 AP901003-0006 :o----:-----:-----:-----:--
WSJ880603-0129 :-----:-----:-----:
WSJ910418-0105 :o----:-----:-----:-----:-----:-----:-----:-----:-----:--
```

Figure 1: Document set in which key sentences are in the beginning of each document (G_1)

```
TOPIC: d108gh.200e
               0    5     10    15    20    25    30    35    40
 AP900802-0118 :-----:-----:
 AP900802-0180 :-----:-----:----o:o-
 AP900803-0008 :-----:-----:-----:--
 AP900807-0029 :o----:-----:-----:-----:---
 AP900807-0093 :-----:-----:--o--:--
 AP900808-0030 :----o:-----:-----:---o-:--
 LA080290-0037 :---o-:-----:--oo-:o-
 LA080290-0233 :-----:-----:-----:-----:-----:-----:-----:-----:--
 LA080690-0111 :-----:-----:-----:-----:-
 LA080890-0078 :-----:----o:--
```

Figure 2: Document set in which key sentences are scattered over documents (G_2)

NEs throughout document sets were related with the distribution of key sentences. Since the frequency is normalized with the total number of NEs, the range of the feature is [0, 1]. The document frequency is used to find salient words over the given document set. It is therefore counted in a given document set, not in other document database like calculating tf*idf values. The document frequency is normalized with the number of documents:

Person_word_TF: 0.44
Person_word_DF: 0.89
Location_word_TF: 0.56
Location_word_DF: 1.00
...

Frequency of NE classes: Other than each word, the frequency and document frequency of each NE class are used as features. These features are used to find salient NE classes even when there are no salient NE words.

Variation of NE tokens: The variation of NE tokens indicates how different NE tokens are in the NE class. When the frequency of an NE class is F and the distinct number of NE tokens is D, the following equation is used to calculate the feature value V, so that the range is within [0, 1].

$$V = \frac{F - D}{F - 1}$$

For example, if 'New York' appears twice and 'Tokyo' appears once in a given document set, 'Location_variation' is $(3 - 2)/(3 - 1) = 0.5$.

Person_variation: 0.33
Location_variation: 0.5
...

A module to divide document sets uses the above features with machine learning technique. We applied here a Support Vector Machine (SVM) (TinySVM, 2002) for machine learning.

4.3 Experimental results

Table 5 shows experimental results of dividing document sets. We have done 3-fold cross validation for the 137 document sets. The accuracy of automatic division of document sets is about 54%, which is comparable to the random division. It shows the features we used were not sufficient for dividing document sets into two groups effectively.

Some of the features, however, are promising for the division of document sets. We checked binary features by Fisher's exact test and real-valued features by Wilcoxon's

Table 5: Evaluation results of automatic division of document sets

	Fold 1	Fold 2	Fold 3	Ave.
Accuracy	57.8%	50.0%	54.3%	54.04%

Table 6: Features that have different distributions in two document set groups

Feature	Feature type	p-value
Location_variation	continuous	0.0007
Timespan ≤ 7	binary	0.0003
Timespan > 365	binary	0.0042
Organization_Variation	continuous	0.0404

rank sum test. The features shown in Table 6 have different distributions in two document set groups with the significant level of 95% ($p < 0.05$). This result shows that among the features, the variation in an NE class could be effective in showing the difference of document sets. Time span information also has potential to show the difference between two groups of document sets. Actually, 26 out of 30 document sets that had a time span within a week were included in G_2.

5 Concluding remarks

We have presented evaluation results of our summarization system at DUC-2003. Our system incorporated a module that estimated similarity between sentences for multi-document summarization. The similarity between sentences was measured by Dice coefficient, and the similarity information was used for selecting a representative sentence among similar sentences. We participated in three multi-document summarization tasks at DUC-2003, and the evaluation results showed our system had good results in coverage at Task 3, and in quality questions at Task 4.

We also found that there were two main groups of distributions of key sentences in document sets, and conducted a preliminary experiment to divide document sets into the two groups. We are planning to continue the experiment to improve the performance of automatic division by introducing another features.

References

Chinatsu Aone, Mary Ellen Okurowski, and James Gorlinsky. 1998. Trainable, Scalable Summarization Using Robust NLP and Machine Learning. In *Proc. of the 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics*, pages 62–66.

H. Edmundson. 1969. New methods in automatic abstracting. *Journal of ACM*, 16(2):264–285.

Julian Kupiec, Jan Pedersen, and Francine Chen. 1995. A Trainable Document Summarizaer. In *Proc. of the Fourteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 68–73.

K. R. McKeown, R. Barzilay, D. Evans, V. Hatzivassilogou, M. Yen Kan, B. Schiffman, and S. Teufel. 2001. Columbia Multi-Document Summarization: Approach and Evaluation. In *Online Proc. of DUC2001*.

Chikashi Nobata, Satoshi Sekine, Kiyotaka Uchimoto, and Ralph Grishman. 2002. Summarization System Integrated with Named Entity Tagging and IE pattern Discovery. In *Proceedings of the LREC-2002 Conference*, pages 1742–1745, May.

T. Nomoto and Y. Matsumoto. 1997. The Reliability of Human Coding and Effects on Automatic Abstracting (in Japanese). In *IPSJ-NL 120-11*, pages 71–76, July.

Satoshi Sekine, Kiyoshi Sudo, and Chikashi Nobata. 2002. Extended Named Entity Hierarchy. In *Proceedings of the LREC-2002 Conference*, pages 1818–1824.

TinySVM. 2002. http://cl.aist-nara.ac.jp/˜taku-ku/software/TinySVM/.

Hideo Watanabe. 1996. A Method for Abstracting Newspaper Articles by Using Surface Clues. In *Proc. of the 16th International Conference on Computational Linguistics*, pages 974–979.