CRL/NYU Summarization System at DUC-2004

Chikashi Nobata† and **Satoshi Sekine**‡
† National Institute of Information and Communications Technology*
3-5 Hikari-dai, Seika-cho, Soraku-gun, Kyoto 619-0289 Japan
nova@nict.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 two multi-document summarization tasks (Task 2 and Task 5) at the DUC-2004 formal run and evaluated the performance of our summarization system. Our system based on sentence extraction also uses a module to estimate similarity between sentences. The similarity information was used for either selecting the representative sentence among similar sentences or gathering key sentences that have similar structures but different contents. We also incorporated a module which categorized document sets into two groups corresponding to the distribution of key sentences.

1 Introduction

For multi-document summarization tasks in DUC-2004, we have modified our summarization system based on sentence extraction technique(Mani, 2001). We made revisions in a module for similarity between sentences, and also incorporated categorization of document sets corresponding to the distribution of key sentences.

We participated in two multi-document summarization tasks (Task 2 and Task 5) at the DUC-2004 formal run and evaluated the performance of our system. The formal run data were 50 TDT clusters in Task 2 and 50 TREC clusters in Task 5. Task 2 is to make short summaries at each document cluster, and Task 5 is to make short summaries focused by questions of the form "Who is X?", where X is the name of a person.

In the following sections, we explain an overview of our system and the evaluation results at DUC-2004 formal run.

*Formerly known as Communications Research Laboratory (CRL).

2 System overview

In this section, we briefly explain scoring functions used for extracting key sentences and other modules used in our summarization system.

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:

$$\text{P1. } \mathit{Score}_{\text{pst}}(S_i)(1 \leq i \leq n) = 1 \quad (\text{if} \quad i < N)$$
$$= 0 \quad (\text{otherwise})$$

where n is the number of sentences in a given document. The second method is to give the reciprocal of the sentence position; the score of ith sentence $\mathit{Score}(S_i)$ is

$$\text{P2. } \mathit{Score}_{\text{pst}}(S_i) = \frac{1}{i}.$$

These two methods are based on the hypothesis that the sentences in the beginning of the article are more important than those in the other part.

The third method is a modified version of the second one; the method checks the sentence position from the end of the article as well as the beginning:

$$\text{P3. } \mathit{Score}_{\text{pst}}(S_i) = \max(\frac{1}{i}, \frac{1}{n-i+1}).$$

The method is based on the hypothesis that the sentences in both the beginning and the end of the article are more

important than those in the middle. The second method (P2) performed best at the training stage.

2.1.2 Sentence length

The second feature used to set the significance of sentences is 'Sentence length.' The length here means the number of words in the sentence. The first function returns the relative length of each sentence (L_i) to the maximum length of the sentence (L_{max}). Because we would like to set it uniformly in the whole document sets, we fixed the value of (L_{max}) to 100 in advance.

$$\text{L1. } Score_{\text{len}}(S_i) = \frac{L_i}{L_{max}} \quad (\text{if} \quad L_i \leq L_{max}) \\ = 1 \quad (\text{otherwise}).$$

The second function sets the score to a negative value as a penalty when the sentence is shorter than a certain length (L_{min}):

$$\text{L2. } Score_{\text{len}}(S_i) = 0 \quad (\text{if} \quad L_i \geq L_{min}) \\ = \frac{L_i - L_{min}}{L_{min}} \quad (\text{otherwise}).$$

Since we set L_{min} to 10 in the following evaluation, a sentence with 10 or fewer words received a penalty score. The second method (L2) 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 sentence score with tf*idf values of words is calculated with normalization(Madani,). When a document D is given, our system calculates the Euclidean norm of tf*idf values for all words in D (D_{norm}):

$$D_{\text{norm}} = \sqrt{\sum_{w \in D} \text{tf*idf}(w)^2}.$$

Then, the score for the ith sentence (S_i) in D is calculated as follows:

We have three functions for tf*idf, where term frequencies were calculated differently. The first one uses the raw term frequencies, and the others are two ways of normalizing the figure:

$$\text{T1. tf*idf}(w) = tf(w) \log \frac{DN}{df(w)}$$

$$\text{T2. tf*idf}(w) = \frac{tf(w)\text{-}1}{tf(w)} \log \frac{DN}{df(w)}$$

$$\text{T3. tf*idf}(w) = \frac{tf(w)}{tf(w)+1} \log \frac{DN}{df(w)}$$

where *DN* is the number of given documents. We used all the articles in the Wall Street Journal in 1994 and 1995 to count document frequencies. The third function (T3) was selected at the training stage.

2.1.4 Headline (Task 5)

The fourth scoring function is to use the headline of a document to establish the significance of sentences. Our system used this feature only at Task 5, because documents at Task 2 didn't have headlines. 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:

$$\text{H1. } Score_{\text{hl}}(S_i) = \frac{\sum_{w \in H \cap S_i} \frac{tf(w)}{tf(w)+1} \log \frac{DN}{df(w)}}{\sum_{w \in H} \frac{tf(w)}{tf(w)+1} \log \frac{DN}{df(w)}}.$$

We also evaluated the 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:

$$\text{H2. } Score_{\text{hl}}(S_i) = \frac{\sum_{e \in H \cap S_i} \frac{tf(e)}{tf(e)+1}}{\sum_{e \in H} \frac{tf(e)}{tf(e)+1}}.$$

The second method (H2) performed best at the training stage.

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 ($Score_j()$) and a parameter (α_j) as follows:

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

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

Table 1: Contribution of features

	Contribution	
Feature	Task 2	Task 5
Position	0.372	0.450
Length	0.006	0.003
Tf*idf	0.623	0.547
Headline	-	0.001

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 are 30 document sets (299 documents) from DUC-2001 training data and 58 document sets (556 documents) from DUC-2002 extracts data. Table 1 shows 'Contribution' of each feature that was the basis of a scoring function. The value of contribution here means the product of the optimized parameter weight and the standard deviation of the score, because 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 parameters were normalized by the norm of all parameters; i.e.

$$\sum_j \alpha_j = 1.$$

We can see that the most influential feature in sentence scores was tf*idf, and the second most was sentence position.

2.3 Similarity between sentences

Our summarization system uses a module to estimate the similarity between sentences. Similarity values are used to either select one key sentences among semantically similar sentences or output a set of similar sentences with high sentence scores.

There are two kinds of similarity functions to check if a given sentence is redundant or necessary. The assumption here is:

1. If two sentences are similar and have no NEs:
   the sentence pair has the same contents.

2. If two sentences are similar and share NEs:
   the sentence pair has the same contents.

3. If two sentences are similar and both have different NE tokens of the same types:
   the sentence pair has the similar structure but different contents.

For example, in articles about one criminal case the preceding facts of the case are described repeatedly. These repetition should be removed. On the other hand, when each article describes an earthquake that occurred at different place in a given document set, expressions in the articles are typical and similar, but tell us different information. These expression should be included in the summary of the document set. Based on that assumption, our system calculates two similarity functions; one is based on content words (SimW), and the other similarity function is based on NEs (SimN).

The system uses Dice, Jaccard, or cosine coefficients as a similarity measure based on the number of words between two sentences. When two sentences are represented as word vectors S_x and S_y, each coefficient between them is calculated as follows(Manning and Schuetze, 2000):

$$\textit{Dice}(S_x, S_y) = \frac{2|S_x \cap S_y|}{|S_x| + |S_y|}$$
$$\textit{Jaccard}(S_x, S_y) = \frac{|S_x \cap S_y|}{|S_x \cup S_y|}$$
$$\textit{cosine}(S_x, S_y) = \frac{|S_x \cap S_y|}{\sqrt{|S_x| \times |S_y|}}$$

The following three types of weights at each word can also be selected from the following:

Binary: if the word appears on the sentence, the weight is set to 1. The weight is set to 0 otherwise.
Tf: the term frequency of the word
Tf*idf: the tf*idf value of the word

The system uses one of coefficients with one of the weights to calculate similarities. For calculating both values of SimW and SimN, we used cosine coefficient with the binary weight from results at the training stage.

One of our system (SysA) assumes that a given sentence pair is similar each other when the coefficient between the sentences is higher than a threshold. That is, two sentences are regarded as similar when SimW is greater than a threshold T_{sw}, and regarded as sharing NEs when SimN is greater than another threshold T_{sn}. The values of both thresholds T_{sw} and T_{sn} were set to 0.4 at the training stage. The other system (SysB) uses only SimW, i.e. the system uses the similarity value for selecting the one sentence that has the highest sentence score among similar sentences. The comparison of these two system is mentioned in Section 3.

2.4 Division of document sets

We incorporated a module for dividing document sets into two groups from a different viewpoint. 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.

We tried to classify document sets based on how these should be summarized. 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. 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 in the type of G_1. On the other hand, document sets related with a single event such as a particular case of shooting spree were mostly in the type of G_2.

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

Time Span 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. Time span throughout the given document set is calculated from the document ID and the dateline of each document. The time span information t is converted to binary features. We selected following two features that used time span information.

Is_within_week: 1 (if $t \leq 7$) 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 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 $(3-2)/(3-1) = 0.5$.

Location_variation: 0.5
Organization_variation: 0.33
...

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.

2.5 Person names (Task 5)

Since the questions of document sets in Task 5 is "who is X?", there are given person names at each document set. In Task 5, our system uses the information for screening sentences. A coreference module is used to find pronouns and anaphora expressions related with a given person name in the document set, and our system extracts sentences that have the person name or anaphoric expressions.

3 Evaluation results

We participated in Task 2 and Task 5 of multi-document summarization tasks in DUC-2004. Evaluation results of our system are shown in Table 2 and Table 3. 'Sys.' corresponds to our system's results, 'Ave.' the average of evaluation results among all participants, and 'BL' the

Table 2: DUC-2004 Evaluation results in Task 2

	Sys.	Ave.	BL
Q01	2.3	3.1	1.4
Q02	2.1	2.8	2.3
Q03	1.6	1.7	1.3
Q04	1.2	2.2	1.3
Q05	1.5	1.3	1.0
Q06	1.2	1.7	1.3
Q07	1.4	1.7	1.3
MC	0.24	0.21	0.20
ROUGE-1	0.304	0.337	0.324
ROUGE-2	0.076	0.069	0.064
ROUGE-3	0.030	0.022	0.020
ROUGE-4	0.015	0.009	0.007
ROUGE-L	0.325	0.346	0.346
ROUGE-W	0.114	0.119	0.119

Table 3: DUC-2004 Evaluation results in Task 5

	Sys.	Ave.	BL
Q01	2.9	3.5	1.6
Q02	2.3	3.0	2.2
Q03	1.7	1.6	1.4
Q04	1.2	2.2	1.4
Q05	2.1	1.7	1.1
Q06	1.2	1.7	1.4
Q07	1.4	1.8	1.8
MC	0.18	0.20	0.19
ROUGE-1	0.263	0.324	0.314
ROUGE-2	0.063	0.073	0.063
ROUGE-3	0.023	0.027	0.021
ROUGE-4	0.011	0.013	0.010
ROUGE-L	0.284	0.340	0.340
ROUGE-W	0.098	0.115	0.115

baseline system's results. Evaluation results with Mean Coverage (MC) and ROUGE are also included in the tables.

Table 2 and Table 3 show that while evaluation results with ROUGE were not good, our system obtained good evaluation results in quality questions. Among systems, it was the best in questions 1, 2, and 4 on Task 2, and questions 2, 4, and 6 on Task 5.

The comparison of results between SysA and SysB is shown in Table 4. We can see that the selection of similarity functions helped improve the evaluation results of ROUGE-2, ROUGE-3, and ROUGE-4, but didn't contribute to improve those of ROUGE-1, ROUGE-L, and ROUGE-W.

4 Concluding remarks

We have presented evaluation results of our summarization system at DUC-2004. Our system incorporated a module that categorized document sets into two groups corresponding to the distribution of key sentences. A module to estimate similarity between sentences was also modified, so that the similarity information was used either for selecting a representative sentence among similar sentences or gathering key sentences that have similar structures but different contents. We participated in Task 2 and Task 5 of multi-document summarization tasks at DUC-2004. The evaluation results showed that while modified similarity module didn't work well, our system obtained good evaluation results in quality questions.

Table 4: Comparison of systems (Task 2)

	SysA	SysB
ROUGE-1	0.30420	0.32051
ROUGE-2	0.07616	0.07154
ROUGE-3	0.03033	0.02580
ROUGE-4	0.01451	0.01200
ROUGE-L	0.32471	0.33884
ROUGE-W	0.11396	0.11799

Acknowledgements

A part of this research is supported by the Defense Advanced Research Projects Agency as part of the Translingual Information Detection, Extraction and Summarization (TIDES) program, under Grant N66001-001-1-8917 from the Space and Naval Warfare Systems Center, San Diego, and by the National Science Foundation under Grant ITS-00325657. This paper does not necessarily reflect the position of the U.S. Government. We would like to thank our colleagues at New York University and National Institute of Information and Communications Technology, who provided useful suggestions and discussions, including Prof. Ralph Grishman, Mr. Kiyoshi Sudo and Mr. Yusuke Shinyama at NYU, and Dr. Kiyotaka Uchimoto and Dr. Hitoshi Isahara at NICT.

References

Madani. http://classes.seattleu.edu/computer_science/csse470/Madani/ABCs.html. ABCs of Text Categorization.

I. Mani. 2001. *Automatic Summarization*. John Benjamins Publishing Company.

Christopher D. Manning and Hinrich Schuetze. 2000. *Foundations of Statistical Natural Language Processing*. The MIT Press.

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*.

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/.