Sentence Extraction System Assembling Multiple Evidence

Chikashi NOBATA† Satoshi SEKINE‡ Masaki MURATA†
Kiyotaka UCHIMOTO† Masao UTIYAMA† Hitoshi ISAHARA†
†Keihanna Human Info-Communication Research Center
Communications Research Laboratory
2-2-2 Hikari-dai, Seika-cho, Soraku-gun Kyoto 619-0289 JAPAN
{nova,murata,uchimoto,mutiyama,isahara}@crl.go.jp
‡Computer Science Department, New York University
715 Broadway, 7th floor, New York, NY 10003 USA
sekine@cs.nyu.edu

Abstract

We have developed a sentence extraction system that estimates the significance of sentences by integrating four scoring functions that use as evidence sentence location, sentence length, TF/IDF values of words, and similarity to the title. Similarity to a given query is also added to the system in the summarization task for information retrieval. Parameters for scoring functions were optimized experimentally using dry run data of the TSC. Results from the TSC formal run showed that our method was effective in the sentence extraction task.

1 Introduction

In recent years, there has been an accumulation of vast amounts web-based documents and machine-readable text. To obtain useful information from these documents efficiently, there have been several ongoing research studies of natural language processing for use in tasks such as information retrieval (IR), information extraction (IE), automatic summarization.

Sentence extraction is a method used for automatic text summarization [6]. Various clues have been used for sentence extraction. The lead-based method, which is simple but still effective, uses the sentence location in a given document. Statistical information, like word frequency and document frequency, has also been used to estimate the significance of sentences. Linguistic clues that indicate the structure of a document are also useful for extracting important sentences.

Edmundson [2] proposed a method of integrating several clues to extract sentences. He manually assigned parameter values to integrate evidence for estimating the significance score of sentences. On the other hand, Aone et al. [1], and Nomoto and Matsumoto [5] generated a decision tree [7] for sentence extraction from training data.

Our system uses several pieces of evidence to estimate the sentence significance in a uniform way. Each piece of evidence is integrated using parameters; the values of the parameters are set using training data. Suitable parameter sets can be selected for each information of section and compression ratio.

In the following sections, we first explain the methods used in our system, then show and discuss the evaluation results from the TSC, Text Summarization Challenge, which was held by the National Information Institute.

2 Methods

In this section, we introduce our sentence extraction system. First, we explain the scoring functions used in the system, and then discuss the other parts, such as the threshold types, patterns, and parameters that the system uses.

2.1 Score function

Our system uses four types of metrics to estimate the significance of sentences: sentence location, sentence length, term frequency, and similarity to the title. In task B of the TSC, summarization for IR task, the system also uses similarity to a given query. The significance of sentences is given by the sum of the values of the above metrics with parameters. Each metric will be explained in the following subsections.

2.1.1 Sentence location

Our system has a function which uses sentence location to set significance of sentences. In this function, there are three different methods to handle sentence location. The first method is to give a score of 1 to the first N sentences and 0 to the others, where N is a given threshold for the number of sentences. That is, the score of the ith sentence *Score*(S_i) is:

$$\begin{aligned} \textit{Score}_{\text{loc}}(S_i)(1 \leq i \leq n) &= 1(\text{if} \quad i < N) \\ &= 0(\text{otherwise}) \end{aligned} \tag{1}$$

where n is the number of sentences in a given article. The second method is to give the reciprocal of the sentence location; the score of ith sentence *Score*(S_i) is

$$\textit{Score}_{\text{loc}}(S_i) = \frac{1}{i} \tag{2}$$

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 location from the end of the article as well as the beginning:

$$\textit{Score}_{\text{loc}}(S_i) = \max(\frac{1}{i}, \frac{1}{n-i+1}) \tag{3}$$

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.

2.1.2 Sentence length

The second scoring function is to use sentence length to set the significance of sentences. The length here means the number of characters in the sentence. In this function, there are two methods used to handle sentence length. One method just returns the length of each sentence (L_i), and the other sets the score to a negative value as a penalty when the sentence is shorter than a certain length (C), like:

$$\begin{aligned} \textit{Score}_{\text{len}}(S_i) &= L_i \quad (\text{if} \quad L_i \geq C) \\ &= L_i - C \quad (\text{otherwise}) \end{aligned}$$

2.1.3 TF/IDF

The third scoring function is based on term frequency and document frequency. The hypothesis here is that the more words that are specific to an article are in a sentence, the more important the sentence is. The target words are nouns (excluding temporal or adverbial nouns). For each of these nouns in a sentence, the system calculates the TF/IDF score. The total score is the significance of the sentence. Word segmentation is performed by JUMAN [4].

When a set of documents is given in advance, our system counts the term frequency (*tf*) and the (document frequency *df*) for each word w, then calculates the TF/IDF score as follows [8]:

$$\text{TF/IDF}(w) = \frac{tf}{1+tf} \log \frac{DN}{df}$$

where *DN* is the number of given documents. We used all the articles in the Mainichi newspaper in 1994 and 1995 to count document frequency.

2.1.4 Headline

The fourth scoring function is to use the headline of an article to set the significance of sentences. The hypothesis here is that sentences related to a headline are significant for use in summarizing an article. This function estimates the relevance between a headline and a sentence using the TF/IDF values of words in the headline. The target words are the same as those mentioned in Section 2.1.3. When the nouns in a headline appear in a sentence, the TF/IDF values of the nouns are calculated and added to the sentence score. After this calculation is finished, the system normalizes the sentence score by using that of the headline. This scoring function can be applicable for the headline itself, and the score is always larger than that of the sentences in the article, therefore the range of the sentence score is 0 to 1.

We also evaluated this scoring function using named entities instead of the nouns. Named entities were annotated by a named entity extraction program based on a maximum entropy model [9]. For named entities, only the term frequency was used, because we judged that the document frequency for entities was usually quite small and thereby making the difference between entities negligible.

2.1.5 Query

In task B, IR queries are given in addition to target documents. The fifth scoring function is to use these queries to set the significance of sentences. This function estimates the relevance between a given query and each sentence in an article, using the TF/IDF values of words in the query. The target words are the same as those mentioned in Section 2.1.3.

One query contains shorter one("DESCRIPTION") and longer one("NARRATIVE"). We assumed that the nouns that appeared in both the DESCRIPTION and the NARRATIVE parts were more important than the other nouns, therefore this function doubled the TF/IDF values of these nouns before adding them to the sentence score.

We have submitted two summary sets for task B. One summary set, *Sum1*, was intended to improve the measure

of "Precision" when an IR task was performed. The parameters were the ones used in task A1, when the compression ratio was 10%. While the compression ratio was set to 10%, our system extracted at least three sentences from each article regardless of the length of an article. We tried to utilize the headline information for this task; if a headline shares some words with the query, the parameter for the query function is doubled. Moreover, we set a threshold for the total score for sentences. A sentence was not extracted when the score was lower than the threshold.

The other summary set, *Sum2*, was intended to supply sufficient information for the IR task and to improve the measure of "Recall". Therefore, the compression ratio was set to 50%, and at least three sentences were extracted as in *Sum1*

2.2 Threshold

Our system calculates a significance score for all of the sentences, and sets the ranking of each sentence in descending order of score. To determine how many sentences are to be extracted from these ranked ones, our system can use three types of thresholds: the number of sentences, the number of characters, and the score of the sentence. Regardless of the threshold type, the order of the extracted sentences is the same as in the original articles.

When the number of sentences N is given as a threshold, the system outputs the top N sentences in the rank. When the number of characters is given, the system converts it to the number of sentences; the maximum number of sentences within a given number of characters is calculated by accumulating the number of characters of the sentences in ascending order of rank. After the number of sentences is calculated, the system uses the result as the threshold. When the threshold score is given, the system outputs the sentences that have scores higher than the threshold score.

2.3 Patterns

Our system applied patterns to shorten sentences in task A2 of the TSC. We intended that the summary generated include as many sentences as possible by applying transformation patterns. There have been several research studies on the use of transformation patterns or rules for summarization. Wakao et al. [10] manually created patterns for the subtitles of TV news programs. Katoh and Uratani [3] proposed a method to acquire transformation rules automatically from TV news text and teletext.

We created 20 rules manually by looking at dry run data. Some examples of the patterns used in task A2 are shown in Figure 1. The automatic acquisition of such rules will be the focus of the future work.

Parenthetic:	`(  *  )`	$\rightarrow$	ϕ
Beginning:	`Shikashi  [,]`	$\rightarrow$	ϕ
End:	`bekide␣ha␣nai␣ka␣.`	$\rightarrow$	`bekida␣.`

Figure 1. Examples of patterns used in task A2

2.4 Parameters

Our system uses parameters to integrate the results of each scoring function in order to calculate the total score of a sentence. The total score of a sentence (S_i) is set using scoring functions ($Score_j()$) and parameters (α_j) as follows:

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

We approximated the optimal values of these parameters with data used in task A1 of the TSC dry run. After the range of each parameter was set manually, the system changed the values of the parameters within the range and performed a summarization on the dry run data. Each score was recorded whenever the parameter values were changed, and the parameter values having the best score were stored.

The dry run data we used was comprised of 30 newspaper articles and the manually created summary. These summaries were created for every compression ratio (10, 30, and 50%) and the 30 summaries were available at each compression ratio. We split the summaries into two sets, i.e., 15 editorials and 15 articles. We assumed that the characteristics of the editorials were different from those of articles. That is, we divided summaries into six classes by the compression ratio and the section information, and set the parameter values for each summary class. The types of location and length functions were also selected at each class.

On the other hand, two compression ratios were set in task A2: 20% and 40%. We applied the parameter set of task A1 to task A2; the parameter set for 10% in task A1 was applied to 20% in task A2, and that for 30% to 40%.

3 Results and discussion

In this section, we show the evaluation results of our system for each task of the TSC formal run and discuss the actual failures of generated summaries.

3.1 Task A1: Sentence extraction

Table 1 shows the evaluation results of our system and the base-line systems in task A1, sentence extraction task. The figures in Table 1 are points of F-measure. Since all of the participants output sentences as many as the upper

Table 1. Evaluation results of task A1

Ratio	10%	30%	50%	Avg.
Our system	0.363	0.435	0.589	0.463
Lead-based	0.284	0.432	0.586	0.434
TF-based	0.276	0.367	0.530	0.391

limit, the values of recall, precision, and F-measure were the same.

Our system obtained better results than the baseline systems, especially when the compression ratio was 10%. The average performance was the second among 10 participants.

We analyzed the causes of the errors our systems made. One type of a sentence that was missing was the short sentence. Since our system considers shorter sentences less significant by the length scoring function, short sentences without a contribution from the other scoring functions do not appear in the summary. For example, in the 30% summary of the other sections, 42% (29/69) of the missing sentences had less than 25 characters, and in the 50% summary of other sections, 64% (33/85) of the missing sentences had less than 20 characters.

In addition, the TF/IDF and headline functions gave higher scores to the sentences that described specific facts than they did to the abstract expressions. On the other hand, the key summaries, which a human annotater generated, included more abstract and shorter expressions. Table 2 shows the system's performance when one of the features was ignored. We can see the contribution of each feature in task A1 from the table. The length, TF/IDF, and headline function showed a negative or zero contribution in each compression ratio. These results show the difference between the key summaries and the our system's outputs, as we mentioned. To extract the sentences missing in the summaries generated by our system, we will need to develop another feature to the system.

The separation created between editorials and other general articles was effective in improving the performance, especially for selecting the type of location function. Table 3 shows the performance of each location function. A location type in this table corresponds to the equation number, i.e., Loc. 1 is the scoring method described in Equation 1. The results of Loc. 1 and Loc. 2 were gathered because they had the same results when the location function was used alone. As shown in Table 3, Loc. 1 and Loc. 2 are more suitable for other articles, and Loc.3 is good for editorials. Our system used each type of location function for suitable articles, which was responsible for most of the system's performance in task A1.

Table 3. Evaluation results of the types of location function

Loc. 1, Loc. 2				
Ratio	10%	30%	50%	Avg.
Editorials	0.158	0.256	0.474	0.293
Others	0.394	0.478	0.586	0.486
All	0.276	0.367	0.530	0.391
Loc. 3				
Editorials	0.323	0.360	0.557	0.413
Others	0.356	0.436	0.544	0.445
All	0.339	0.398	0.550	0.429
Mixed				
All	0.359	0.419	0.572	0.450

3.2 Task A2: Free summarization

In task A2, free summarization task, we submitted the summaries generated by sentence extraction with patterns to shorten the sentences. The content-based evaluation results of task A2 are shown in Table 4. "FREE" summary means a summary a human wrote freely with a restriction of the number of characters. "PART" summary means a summary generated by extracting phrases from articles.

In the content-based evaluation, the results of our system were much the same as those of TF-based system, and worse than the baseline systems' results for the 40%. One of the reasons was the parameters the system used. As we mentioned in the previous section, we applied the parameter sets for task A1 to this task; these parameters were probably insufficient. We should have approximated the optimal values of the parameters for task A2 as well as those for task A1.

The evaluation results obtained by human experts are shown in Table 5. The "R" beside compression ratios means the readability evaluation of summaries, and "C" means the evaluation of the contents. The values in the table are the average rankings of the four systems: the FREE summary, PART summary, lead-based summary, and the system being evaluated. While the evaluation of the contents did not show the difference between these two system, our system had a better result than the TF-based one did in readability. This is probably because the summaries were generated using sentence extraction, and also the applied patterns were created to preserve the readability of sentences.

The objective of applying patterns is to shorten sentences so that the generated summary can include as many sentences as possible. Table 6 shows how the average number of sentences in an article was changed by applying patterns. While the contribution of the patterns in the 20% summary was virtually nothing, the number of sentences increased in

Table 2. Evaluation results of task A1 when one feature was ignored

Ratio	10%	30%	50%	Avg.
All features	0.363	0.435	0.589	0.463
(ALL)−Location	0.326(−.037)	0.394(−.041)	0.575(−.014)	0.432(−.031)
(ALL)−Length	0.372(+.009)	0.472(+.037)	0.600(+.011)	0.481(+.018)
(ALL)−TF/IDF	0.372(+.009)	0.439(+.004)	0.582(−.007)	0.464(+.001)
(ALL)−Headline:Word	0.403(+.040)	0.449(+.014)	0.589(±.000)	0.480(+.017)
(ALL)−Headline:NE	0.381(+.018)	0.438(+.003)	0.589(±.000)	0.469(+.006)

Table 4. Content-based evaluation results of task A2

Comparison with the FREE summary			
Ratio	20%	40%	Avg.
Our system	0.452	0.566	0.509
TF-based	0.437	0.596	0.516
Lead-based	0.383	0.580	0.481
Comparison with the PART summary			
Our system	0.507	0.611	0.559
TF-based	0.476	0.622	0.549
Lead-based	0.421	0.605	0.513

Table 5. Evaluation results of task A2 by human experts

Ratio	20%R	20%C	40%R	40%C
Our system	3.07	3.33	2.60	3.07
TF-Based	3.20	3.27	2.77	3.07

one third of the articles in the 40% summary.

3.3 Task B: Summarization for information retrieval

We submitted two summaries in task B, summarization for IR. We call them *Sum1* and *Sum2* in the following description. *Sum1* is the summary where the compression ratio was basically 10%, and *Sum2* is the summary where the compression ratio was basically 50%. Table 7 shows the evaluation results of the summaries. "Answer level" means the level of the correct answer with relevance to a given query in the IR task. When the answer level is *A*, only the articles judged *A* are the correct answers of the IR task. When the answer level is *B*, the articles judged *A* or *B* are the correct answers of the IR task. Both summaries have better evaluation results for answer level *B* than for answer level *A*, compared with the summaries submitted by the other participants. From these results, we can say that our summaries had more information to distinguish articles of the level *A* or *B* from non-relevant articles than the other summaries.

Table 6. Average number of sentences in task A2

Ratio	20%	40%
w/o Pattern	4.53 (136/30)	8.93 (268/30)
with Pattern	4.63 (139/30)	9.27 (278/30)

Figures 2 and 3 show the evaluation results of all systems with their average time in task B. Figure 2 shows the results when the answer level was *A*, and Figure 3 shows the result when the level was *B*. While the evaluation of *Sum1* was lower than that of *Sum2* in the both figures, the average time was shorter than that of *Sum2*. Considering that the difference of F-measure is small, *Sum1* is more suitable for the summarization for the IR task than *Sum2*.

The average time for *Sum2* was greater than that for the summaries of any other participant. The compression ratio of *Sum2* was based on 50%, while that of the other participants seemed to set smaller compression ratio. However, the result using *Sum2* was better than that using full texts on both recall and precision with the shorter average time, which demonstrated the effectiveness of *Sum2* for the IR task.

4 Conclusion

Our sentence extraction system that estimates the significance of sentences by integrating some scoring functions participated in all of the tasks of the TSC formal run. The results showed that our method was effective in the sentence extraction task. While the results were not very good in the free summarization task, the transformation patterns worked for abbreviating sentences and increasing the amount of summaries. In the summarization for IR, both of the summary sets we submitted had good results when the relevance level was wide. One of our summary sets was

Table 7. Evaluation results of task B

Answer level: A			
Measurement	Recall	Precision	F
Sum1	0.833	0.728	0.761
Sum2	0.899	0.717	0.785
Full text	0.843	0.711	0.751
TF-based	0.798	0.724	0.738
Lead-based	0.740	0.766	0.731
Answer level: B			
Sum1	0.741	0.921	0.808
Sum2	0.793	0.904	0.828
Full text	0.736	0.888	0.773
TF-based	0.700	0.913	0.776
Lead-based	0.625	0.921	0.712

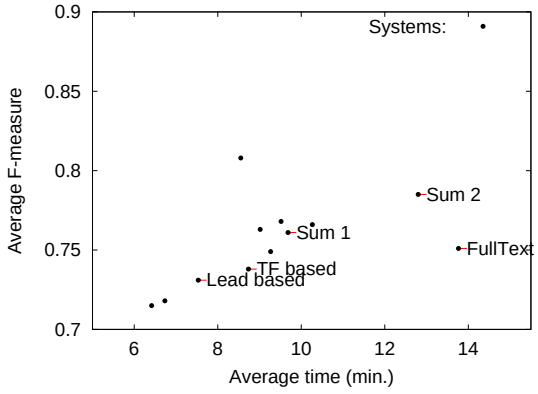

Figure 2. Evaluation results of all systems with average time (answer level A)

longer than those of any other participant, but the result was still better than that using full texts in both average time and performance.

References

[1] Chinatsu Aone, Mary Ellen Okurowski, James Gorlinsky, and Bjornal Larsen. A Scalable Summarization System Using Robust NLP. In *Proceedings of the ACL Work shop on Intelligent Scalable Text Summarization*, pages 66–73, 1997.

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

[3] Naoto Katoh and Noriyoshi Uratani. A new approach to acquiring linguistic knowledge for locally summarizing Japanese news sentences (in Japanese). *Journal of Natural Language Processing*, 6(7):73–92, 1999.

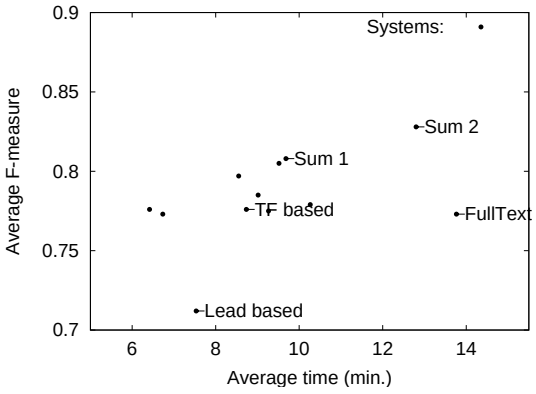

Figure 3. Evaluation results of all systems with average time (answer level B)

[4] Sadao Kurohashi and Makoto Nagao. *Japanese Morphological Analyzing System: JUMAN version 3.61*. Kyoto University, 1999.

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

[6] Manabu Okumura and Hidetsugu Nanba. Automated Text Summarization: A Survey (in Japanese). *Journal of Natural Language Processing*, 6(6):1–26, 1999.

[7] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, Inc., San Mateo, California, 1993.

[8] S. E. Robertson and S. Walker. Some simple effective approximations to the 2-poisson model for probabilistic weighted retreival. In *Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1994.

[9] Kiyotaka Uchimoto, Qing Ma, Masaki Murata, Hiromi Ozaku, and Hitoshi Isahara. Named Entity Extraction Based on A Maximum Entropy Model and Transformation Rules. In *Proceedings of the 38th Annual Meeting of Association for Computational Linguistics (ACL2000)*, pages 326–335, October 2000.

[10] Takahiro Wakao, Terumasa Ehara, and Katsuhiko Shirai. Summarization Methods Used for Captions in TV News Programs (in Japanese). In *IPSJ-NL 122-13*, pages 83–89, July 1997.