Evaluation of Features for Sentence Extraction on Different Types of Corpora

Chikashi Nobata[†], **Satoshi Sekine**[‡] and **Hitoshi Isahara**[†]
† Communications Research Laboratory
3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0289, Japan
{nova, 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 report evaluation results for our summarization system and analyze the resulting summarization data for three different types of corpora. To develop a robust summarization system, we have created a system based on sentence extraction and applied it to summarize Japanese and English newspaper articles, obtained some of the top results at two evaluation workshops. We have also created sentence extraction data from Japanese lectures and evaluated our system with these data. In addition to the evaluation results, we analyze the relationships between key sentences and the features used in sentence extraction. We find that discrete combinations of features match distributions of key sentences better than sequential combinations.

1 Introduction

Our ultimate goal is to create a robust summarization system that can handle different types of documents in a uniform way. To achieve this goal, we have developed a summarization system based on sentence extraction. We have participated in evaluation workshops on automatic summarization for both Japanese and English written corpora. We have also evaluated the performance of the sentence extraction system for Japanese lectures. At both workshops we obtained some of the top results, and for the speech corpus we obtained results comparable with those for the written corpora. This means that the features we use are worth analyzing.

Sentence extraction is one of the main methods required for a summarization system to reduce the size of a document. Edmundson (1969) proposed a method of integrating several features, such as the positions of sentences and the frequencies of words in an article, in order to extract sentences. He manually assigned parameter values to integrate features for estimating the significance scores of sentences. On the other hand, machine learning methods can also be applied to integrate features. For sentence extraction from training data, Kupiec et al. (1995) and Aone et al. (1998) used Bayes' rule, Lin (1999) and Nomoto and Matsumoto (1997) generated a decision tree, and Hirao et al. (2002) generated an SVM.

In this paper, we not only show evaluation results for our sentence extraction system using combinations of features but also analyze the features for different types of corpora. The analysis gives us some indication about how to use these features and how to combine them.

2 Summarization data

The summarization data we used for this research were prepared from Japanese newspaper articles, Japanese lectures, and English newspaper articles. By using these three types of data, we could compare two languages and also two different types of corpora, a written corpus and a speech corpus.

2.1 Summarization data from Japanese newspaper articles

Text Summarization Challenge (TSC) is an evaluation workshop for automatic summarization, which is run by the National Institute of Informatics in Japan (TSC, 2001). Three tasks were presented at TSC-2001: extracting important sentences, creating summaries to be compared with summaries prepared by humans, and creating summaries for information retrieval. We focus on the first task here, i.e., the sentence extraction task. At TSC-2001, a dry run and a formal run were performed. The dry run data consisted of 30 newspaper articles and manually created summaries of each. The formal run data consisted of another 30 pairs of articles and summaries. The average number of sentences per article was 28.5 (1709 sentences / 60 articles). The newspaper articles included 15 editorials and 15 news reports in both data sets. The summaries were created from extracted sentences with three compression ratios (10%, 30%, and 50%). In our analysis, we used the extraction data for the 10% compression ratio.

In the following sections, we call these summarization data the "TSC data". We use the TSC data as an example of a Japanese written corpus to evaluate the performance of sentence extraction.

2.2 Summarization data from Japanese lectures

The speech corpus we used for this experiment is part of the *Corpus of Spontaneous Japanese (CSJ)* (Maekawa et al., 2000), which is being created by NIJLA, TITech, and CRL as an ongoing joint project. The CSJ is a large collection of monologues, such as lectures, and it includes transcriptions of each speech as well as the voice data. We selected 60 transcriptions from the CSJ for both sentence segmentation and sentence extraction. Since these transcription data do not have sentence boundaries, sentence segmentation is necessary before sentence extraction. Three annotators manually generated sentence segmentation and summarization results. The target compression ratio was set to 10%. The results of sentence segmentation were unified to form the key data, and the average number of sentences was 68.7 (4123 sentences / 60 speeches). The results of sentence extraction, however, were not unified, but were used separately for evaluation.

In the following sections, we call these summarization data the "CSJ data". We use the CSJ data as an example of a Japanese speech corpus to evaluate the performance of sentence extraction.

2.3 Summarization data from English newspaper articles

Document Understanding Conference (DUC) is an evaluation workshop in the U.S. for automatic summarization, which is sponsored by TIDES of the DARPA program and run by NIST (DUC, 2001). At DUC-2001, there were two types of tasks: single-document summarization (SDS) and multi-document summarization (MDS). The organizers of DUC-2001 provided 30 sets of documents for a dry run and another 30 sets for a formal run. These data were shared by both the SDS and MDS tasks, and the average number of sentences was 42.5 (25779 sentences / 607 articles). Each document set had a topic, such as "Hurricane Andrew" or "Police Misconduct", and contained around 10 documents relevant to the topic. We focus on the SDS task here, for which the size of each summary output was set to 100 words. Model summaries for the articles were also created by hand and provided. Since these summaries were abstracts, we created sentence extraction data from the abstracts by word-based comparison.

In the following sections, we call these summarization data the "DUC data". We use the DUC data as an example of an English written corpus to evaluate the performance of sentence extraction.

3 Overview of our sentence extraction system

In this section, we give an overview of our sentence extraction system, which uses multiple components. For each sentence, each component outputs a score. The system then combines these independent scores by interpolation. Some components have more than one scoring function, using various features. The weights and function types used are decided by optimizing the performance of the system on training data.

Our system includes parts that are either common to the TSC, CSJ, and DUC data or specific to one of these data sets. We stipulate which parts are specific.

3.1 Features for sentence extraction

3.1.1 Sentence position

We implemented three functions for sentence position. The first function returns 1 if the position of the sentence is within a given threshold N from the beginning, and returns 0 otherwise:

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

The threshold N is determined by the number of words in the summary.

The second function is the reciprocal of the position of the sentence, i.e., the score is highest for the first sentence, gradually decreases, and goes to a minimum at the final sentence:

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

These first two functions are based on the hypothesis that the sentences at the beginning of an article are more important than those in the remaining part.

The third function is the maximum of the reciprocal of the position from either the beginning or the end of the document:

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

This method is based on the hypothesis that the sentences at both the beginning and the end of an article are more important than those in the middle.

3.1.2 Sentence length

The second type of scoring function uses sentence length to determine the significance of sentences. We implemented three scoring functions for sentence length. The first function only returns the length of each sentence (L_i):

$$\text{L1. } \mathit{Score}_{\text{len}}(S_i) = L_i$$

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

$$\text{L2. } \mathit{Score}_{\text{len}}(S_i) = 0 \quad (\text{if} \quad L_i \ge C)$$
$$L_i - C \quad (\text{otherwise})$$

The third function combines the above two approaches, i.e., it returns the length of a sentence that has at least a certain length, and otherwise returns a negative value as a penalty:

$$\text{L3. } \mathit{Score}_{\text{len}}(S_i) = L_i \quad (\text{if} \quad L_i \ge C)$$
$$= L_i - C \quad (\text{otherwise})$$

The length of a sentence means the number of letters, and based on the results of an experiment with the training data, we set C to 20 for the TSC and CSJ data. For the DUC data, the length of a sentence means the number of words, and we set C to 10 during the training stage.

3.1.3 Tf*idf

The third type of scoring function is based on term frequency (*tf*) and document frequency (*df*). We applied three scoring functions for tf*idf, in which the term frequencies are calculated differently. The first function uses the raw term frequencies, while the other two are two different ways of normalizing the frequencies, as follows, where *DN* is the number of documents given:

$$\text{T1. tf*idf}(w) = \mathit{tf}(w) \log \frac{\mathit{DN}}{\mathit{df}(w)}$$
$$\text{T2. tf*idf}(w) = \frac{\mathit{tf}(w)\text{-}1}{\mathit{tf}(w)} \log \frac{\mathit{DN}}{\mathit{df}(w)}$$
$$\text{T3. tf*idf}(w) = \frac{\mathit{tf}(w)}{\mathit{tf}(w)+1} \log \frac{\mathit{DN}}{\mathit{df}(w)}$$

For the TSC and CSJ data, we only used the third method (T3), which was reported to be effective for the task of information retrieval (Robertson and Walker, 1994). The target words for these functions are nouns (excluding temporal or adverbial nouns). For each of the nouns in a sentence, the system calculates a Tf*idf score. The total score is the significance of the sentence. The word segmentation was generated by Juman3.61 (Kurohashi and Nagao, 1999). We used articles from the Mainichi newspaper in 1994 and 1995 to count document frequencies.

For the DUC data, the raw term frequency (T1) was selected during the training stage from among the three tf*idf definitions. A list of stop words were used to exclude functional words, and articles from the Wall Street Journal in 1994 and 1995 were used to count document frequencies.

3.1.4 Headline

We used a similarity measure of the sentence to the headline as another type of scoring function. The basic idea is that the more words in the sentence overlap with the words in the headline, the more important the sentence is. The function estimates the relevance between a headline (H) and a sentence (S_i) by using the tf*idf values of the words (w) in the headline:

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

We also evaluated another method based on this scoring function by using only named entities (NEs) instead of words for the TSC data and DUC data. Only the term frequency was used for NEs, because we judged that the document frequency for an entity was usually quite small, thereby making the differences between entities negligible.

3.1.5 Patterns

For the DUC data, we used dependency patterns as a type of scoring function. These patterns were extracted by pattern discovery during information extraction (Sudo et al., 2001). The details of this approach are not explained here, because this feature is not among the features we analyze in Section 5. The definition of the function appears in (Nobata et al., 2002).

3.2 Optimal weight

Our system set weights for each scoring function in order to calculate the total score of a sentence. The total score (S_i) is defined from the scoring functions ($Score_j()$) and weights (α_j) as follows:

$$\text{TotalScore}(S_i) = \sum_j \alpha_j Score_j(S_i) \quad (1)$$

We estimated the optimal values of these weights from the training data. After the range of each weight was set manually, the system changed the values of the weights within a range and summarized the training data for each set of weights. Each score was recorded after the weights were changed, and the weights with the best scores were stored.

A particular scoring method was also selected in the cases of features with more than one defined scoring methods. We used the dry run data from each workshop as TSC and DUC training data. For the TSC data, since the 30 articles contained 15 editorials and 15 news reports, we estimated optimal values separately for editorials and news reports. For the CSJ data, we used 50 transcriptions for training and 10 for testing, as mentioned in Section 2.2.

Table 1: Evaluation results for the TSC data.

Ratio	10%	30%	50%	Avg.
System	0.363 (1)	0.435 (5)	0.589 (2)	0.463 (2)
Lead	0.284	0.432	0.586	0.434

4 Evaluation results

In this section, we show our evaluation results on the three sets of data for the sentence extraction system described in the previous section.

4.1 Evaluation results for the TSC data

Table 1 shows the evaluation results for our system and some baseline systems on the task of sentence extraction at TSC-2001. The figures in Table 1 are values of the F-measure[1]. The 'System' column shows the performance of our system and its rank among the nine systems that were applied to the task, and the 'Lead' column shows the performance of a baseline system which extracts as many sentences as the threshold from the beginning of a document. Since all participants could output as many sentences as the allowed upper limit, the values of the 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 second among the nine systems.

4.2 Evaluation results for the DUC data

Table 2 shows the results of a subjective evaluation in the SDS task at DUC-2001. In this subjective evaluation, assessors gave a score to each system's outputs, on a zero-to-four scale (where four is the best), as compared with summaries made by humans. The figures shown are the average scores over all documents. The 'System' column shows the performance of our system and its rank among the 12 systems that were applied to this task. The 'Lead'

[1] The definitions of each measurement are as follows:

Recall (REC) = COR / GLD
Precision (PRE) = COR / SYS
F-measure = 2 * REC * PRE / (REC + PRE),

where COR is the number of correct sentences marked by the system, GLD is the total number of correct sentences marked by humans, and SYS is the total number of sentences marked by the system. After calculating these scores for each transcription, the average is calculated as the final score.

Table 2: Evaluation results for the DUC data (subjective evaluation).

	System	Lead	Avg.
Grammaticality	3.711 (5)	3.236	3.580
Cohesion	3.054 (1)	2.926	2.676
Organization	3.215 (1)	3.081	2.870
Total	9.980 (1)	9.243	9.126

Table 3: Evaluation results for the CSJ data.

	Annotators			
	A	B	C	Avg.
REC	0.407	0.331	0.354	0.364
PRE	0.416	0.397	0.322	0.378
F	0.411	0.359	0.334	0.368

column shows the performance of a baseline system that always outputs the first 100 words of a given document, while the 'Avg.' column shows the average for all systems. Our system ranked 5th in grammaticality and was ranked at the top for the other measurements, including the total value.

4.3 Evaluation results for the CSJ data

The evaluation results for sentence extraction with the CSJ data are shown in Table 3. We compared the system's results with each annotator's key data. As mentioned previously, we used 50 transcriptions for training and 10 for testing.

These results are comparable with the performance on sentence segmentation for written documents, because the system's performance for the TSC data was 0.363 when the compression ratio was set to 10%. The results of our experiments thus show that for transcriptions, sentence extraction achieves results comparable to those for written documents, if the are well defined.

4.4 Contributions of features

Table 4 shows the contribution vectors for each set of training data. The contribution here means the product of the optimized weight and the standard deviation of the score for the test data. The vectors were normalized so that the sum of the components is equal to 1, and the selected function types for the features are also shown in the table. Our system used the NE-based headline function (HL (N)) for the DUC data and the word-based function (HL (W)) for the CSJ data, and both functions for the TSC data. The columns for the TSC data show the contributions when the compression ratio was 10%.

Table 4: Contribution (weight × s.d.) of each feature for each set of summarization data.

	TSC			
Features	Editorial	Report	DUC	CSJ
Pst.	P3. 0.446	P1. 0.254	P1. 0.691	P3. 0.055
Len.	L3. 0.000	L3. 0.000	L2. 0.020	L2. 0.881
Tf*idf	T3. 0.169	T3. 0.185	T1. 0.239	T3. 0.057
HL (W)	0.171	0.292	-	0.007
HL (N)	0.214	0.269	0.045	-
Pattern	-	-	0.005	-

We can see that the feature with the biggest contribution varies among the data sets. While the position feature was the most effective for the TSC and DUC data, the length feature was dominant for the CSJ data. Most of the short sentences in the lectures were specific expressions, such as "This is the result of the experiment." or "Let me summarize my presentation.". Since these sentences were not extracted as key sentences by the annotators, it is believed that the function giving short sentences a penalty score matched the manual extraction results.

5 Analysis of the summarization data

In Section 4, we showed how our system, which combines major features, has performed well as compared with current summarization systems. However, the evaluation results alone do not sufficiently explain how such a combination of features is effective. In this section, we investigate the correlations between each pair of features. We also match feature pairs with distributions of extracted key sentences as answer summaries to find effective combination of features for sentence extraction.

5.1 Correlation between features

Table 5 shows Spearman's rank correlation coefficients among the four features. Significantly correlated feature pairs are indicated by '$*$'($\alpha = 0.001$). Here, the word-based feature is used as the headline feature. We see the following tendencies for any of the data sets:

- "Position" is relatively independent of the other features.
- "Length" and "Tf*idf" have high correlation[2].

Table 5: Rank correlation coefficients between features.

TSC Report			
Features	Length	Tf*idf	Headline
Position	0.019	-0.095	-0.139
Length	–	0.546*	0.338*
Tf*idf	–	–	0.696*
TSC Editorial			
Features	Length	Tf*idf	Headline
Position	-0.047	-0.099	0.046
Length	–	0.532*	0.289*
Tf*idf	–	–	0.658*
DUC Data			
Features	Length	Tf*idf	Headline
Position	-0.130*	-0.108*	-0.134*
Length	–	0.471*	0.293*
Tf*idf	–	–	0.526*
CSJ Data			
Features	Length	Tf*idf	Headline
Position	-0.092*	-0.069*	-0.106*
Length	–	0.460*	0.224*
Tf*idf	–	–	0.533*

- "TF*idf" and "Headline " also have high correlation.

These results show that while combinations of these four features enabled us to obtain good evaluation results, as shown in Section 4, the features are not necessarily independent of one another.

5.2 Combination of features

Tables 6 and 7 show the distributions of extracted key sentences as answer summaries with two pairs of features: sentence position and the tf*idf value, and sentence position and the headline information. In these tables, each sentence is ranked by each of the two feature values, and the rankings are split every 10 percent. For example, if a sentence is ranked in the first 10 percent by sentence position and the last 10 percent by the tf*idf feature, the sentence belongs to the cell with a position rank of 0.1 and a tf*idf rank of 1.0 in Table 6.

Each cell thus has two letters. The left letter is the number of key sentences, and the right letter is the ratio of key sentences to all sentences in the cell. The left letter shows how the number of sentences differs from the average when all the key sentences appear equally, regardless of the feature values. Let T be the total number of key sentences, $M(=\frac{T}{100})$ be the average number of key sentences in each range, and S be the standard deviation of the number of key sentences among all cells. The number of key sentences for cell $T_{i,j}$ is then categorized according to one of the following letters:

A:	$T_{i,j} \geq M + 2S$
B:	$M + S \leq T_{i,j} < M + 2S$
C:	$M - S \leq T_{i,j} < M + S$
D:	$M - 2S \leq T_{i,j} < M - S$
E:	$T_{i,j} < M - 2S$
O:	$T_{i,j} = 0$
–:	No sentences exist in the cell.

Similarly, the right letter in a cell shows how the ratio of key sentences differs from the average ratio when all the key sentences appear equally, regardless of feature values. Let N be the total number of sentences, $m(=\frac{T}{N})$ be the average ratio of key sentences, and s be the standard deviation of the ratio among all cells. The ratio of key sentences for cell $t_{i,j}$ is then categorized according to one of the following letters:

a:	$t_{i,j} \geq m + 2s$
b:	$m + s \leq t_{i,j} < m + 2s$
c:	$m - s \leq t_{i,j} < m + s$
d:	$m - 2s \leq t_{i,j} < m - s$
e:	$t_{i,j} < m - 2s$
o:	$t_{i,j} = 0$
–:	No sentences exist in the cell.

When key sentences appear uniformly regardless of feature values, every cell is defined as 'Cc'. We show both the range of the number of key sentences and the ratio of key sentences, because both are necessary to show how effectively a cell has key sentences. If a cell includes many sentences, the number of key sentences can be large even though the ratio is not. On the other hand, when the ratio of key sentences is large and the number is not, the contribution to key sentence extraction is small.

Table 6 shows the distributions of key sentences when the features of sentence position and tf*idf were combined. For the DUC data, both the number and ratio of key sentences were large when the sentence position was ranked within the first 20 percent and the value of the tf*idf feature was ranked in the bottom 50 percent (i.e., Pst. $\leq$ 0.2, Tf*idf $\geq$ 0.5). On the other hand, both the number and ratio of key sentences were large for the CSJ data when the sentence position was ranked in the last 10 percent and the value of the tf*idf feature was ranked

[2]Here we used equation T1 for the tf*idf feature, and the score of each sentence was normalized with the sentence length. Hence, the high correlation between "Length" and "Tf*idf" is not trivial.

Table 6: Distributions of key sentences based on the combination of the sentence position (Pst.) and tf*idf features.

DUC data										
	Tf*idf									
Pst.	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
0.1	Cc	Cc	Cc	Cb	Ba	Ba	Aa	Aa	Aa	Aa
0.2	Cd	Cc	Cc	Cc	Cc	Cc	Bb	Bb	Bb	Bb
0.3	Cd	Cc	Cc	Cc	Cc	Cc	Cc	Cc	Cc	Cc
0.4	Dd	Cc	Cc	Cc	Cc	Cc	Cc	Cc	Cc	Cc
0.5	Dd	Cc	Cc	Cc	Cc	Cc	Cc	Cc	Cc	Cc
0.6	Dd	Cc	Cc	Cc	Cc	Cc	Cc	Cc	Cc	Cc
0.7	Dd	Cc	Cc	Cc	Cc	Cc	Cc	Cc	Cc	Cc
0.8	Cd	Dd	Cc	Cc	Cc	Cc	Cc	Cc	Cc	Cc
0.9	Dd	Dd	Cc	Cc	Cc	Cc	Cc	Cc	Cc	Cc
1.0	Dd	Dd	Cc	Cc	Cc	Cc	Cc	Cc	Cc	Cc
CSJ data										
	Tf*idf									
Pst.	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
0.1	Cc	Cc	Cc	Cc	Bc	Cc	Ab	Bb	Bb	Bb
0.2	Oo	Oo	Cc	Cc	Cc	Cc	Cc	Bb	Bc	Cc
0.3	Oo	Cc	Cc	Cc	Cc	Cc	Cc	Cc	Cc	Cc
0.4	Cc	Cc	Cc	Cc	Oo	Cc	Cc	Cc	Cc	Cc
0.5	Oo	Cc	Oo	Oo	Cc	Oo	Cc	Cc	Cc	Cc
0.6	Cc	Cc	Cc	Cc	Cc	Cc	Cc	Cc	Cc	Cc
0.7	Oo	Cc	Cc	Cc	Cc	Cc	Cc	Cc	Cc	Cc
0.8	Oo	Cc	Cc	Oo	Oo	Cc	Cc	Cc	Cc	Cc
0.9	Cc	Cc	Cc	Cc	Cc	Cc	Cc	Cc	Cc	Bb
1.0	Cc	Cc	Ba	Bb	Bb	Aa	Aa	Bb	Aa	Aa

after the first 30 percent (i.e., Pst. $= 1.0$, Tf*idf ≥ 0.3),. When the tf*idf feature was low, the number and ratio of key sentences were not large, regardless of the sentence position values. These results show that the tf*idf feature is effective when the values are used as a filter after the sentences are ranked by sentence position.

Table 7 shows the distributions of key sentences with the combination of the sentence position and headline features. About half the sentences did not share words with the headlines and had a value of 0 for the headline feature. As a result, the cells in the middle of the table do not have corresponding sentences. The headline feature cannot be used as a filter, unlike the tf*idf feature, because many key sentences are found when the value of the headline feature is 0. A high value of the headline feature is, however, a good indicator of key sentences when it is combined with the position feature. The ratio of key sentences was large when the headline ranking was high and the sentence was near the beginning (at Pst. ≤ 0.2, Headline ≥ 0.7) for the DUC data. For the CSJ data, the ratio of key sentences was also large when the headline ranking was within the top 10 percent (Pst. $= 0.1$, Headline $= 1.0$), as well as for the sentences near the ends of speeches.

These results indicate that the number and ratio of key sentences sometimes vary discretely according to the changes in feature values when features are combined for sentence extraction. That is, the performance of a sentence extraction system can be improved by categorizing feature values into several ranges and then combining ranges. While most sentence extraction systems use sequential combinations of features, as we do in our system based on Equation 1, the performance of these systems can possibly be improved by introducing the categorization of feature values, without adding any new features. We have shown that discrete combinations match the distributions of key sentences in two different corpora, the DUC data and the CSJ data. This indicates that discrete combinations of corpora are effective across both different languages and different types of corpora. Hirao et al. (2002) reported the results of a sentence extraction system using an SVM, which categorized sequential feature values into ranges in order to make the features binary. Some effective combinations of the binary fea-

Table 7: Distributions of key sentences based on the combination of the sentence position (Pst.) and headline features.

DUC data							
	Headline						
Pst.	0.1	0.2–0.5	0.6	0.7	0.8	0.9	1.0
0.1	Ab	−−	−−	Ca	Ba	Ba	Aa
0.2	Ac	−−	−−	Cb	Cc	Ca	Ca
0.3	Ac	−−	−−	Cc	Cc	Cb	Cb
0.4	Ac	−−	−−	Cc	Cc	Cc	Cb
0.5	Ac	−−	−−	Cc	Cc	Cc	Cc
0.6	Bc	−−	−−	Cc	Cc	Cc	Cc
0.7	Bc	−−	−−	Cc	Cc	Cc	Cc
0.8	Ac	−−	−−	Cd	Cc	Cc	Cc
0.9	Bd	−−	−−	Cd	Cc	Cc	Cc
1.0	Bd	−−	−−	Cd	Cc	Cc	Cc
CSJ data							
	Headline						
Pst.	0.1	0.2–0.5	0.6	0.7	0.8	0.9	1.0
0.1	Bc	−−	Cc	Cc	Bb	Cc	Aa
0.2	Bc	−−	Cc	Cb	Cc	Cc	Bb
0.3	Cc	−−	Cc	Cc	Cc	Cc	Cc
0.4	Cc	−−	Oo	Cc	Cc	Cc	Cc
0.5	Cc	−−	Oo	Cc	Oo	Cc	Cc
0.6	Cc	−−	Cc	Cc	Cc	Cc	Cc
0.7	Cc	−−	Oo	Cc	Cc	Cc	Cc
0.8	Cc	−−	Cc	Cc	Cc	Cc	Cc
0.9	Ac	−−	Ca	Cc	Cc	Cc	Cb
1.0	Ab	−−	Ca	Aa	Ba	Ba	Ba

tures in that report also indicate the effectiveness of discrete combinations of features.

6 Conclusion

We have shown evaluation results for our sentence extraction system and analyzed its features for different types of corpora, which included corpora differing in both language (Japanese and English) and type (newspaper articles and lectures). The system is based on four major features, and it achieved some of the top results at evaluation workshops in 2001 for summarizing Japanese newspaper articles (TSC) and English newspaper articles (DUC). For Japanese lectures, the sentence extraction system also obtained comparable results when the sentence boundary was given.

Our analysis of the features used in this sentence extraction system has shown that they are not necessarily independent of one another, based on the results of their rank correlation coefficients. The analysis also indicated that the categorization of feature values matches the distribution of key sentences better than sequential feature values.

There are several features that were not described here but are also used in sentence extraction systems, such as some specific lexical expressions and syntactic information. In our future work, we will analyze and use these features to improve the performance of our sentence extraction system.

References

C. Aone, M. E. Okurowski, and J. Gorlinsky. 1998. Trainable, Scalable Summarization Using Robust NLP and Machine Learning. In *Proc. of COLING-ACL'98*, pages 62–66.

DUC. 2001. http://duc.nist.gov. Document Understanding Conference.

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

T. Hirao, H. Isozaki, E. Maeda, and Y. Matsumoto. 2002. Extracting Important Sentences with Support Vector Machines. In *Proc. of COLING-2002*.

J. Kupiec, J. Pedersen, and F. Chen. 1995. A Trainable Document Summarizaer. In *Proc. of SIGIR'95*, pages 68–73.

S. Kurohashi and M. Nagao, 1999. *Japanese Morphological Analyzing System: JUMAN version 3.61*. Kyoto University.

Chin-Yew Lin. 1999. Training a selection function for extraction. In *Proc. of the CIKM'99*.

K. Maekawa, H. Koiso, S. Furui, and H. Isahara. 2000. Spontaneous Speech Corpus of Japanese. In *Proc. of LREC2000*, pages 947–952.

C. Nobata, S. Sekine, H. Isahara, and R. 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.

S. E. Robertson and S. Walker. 1994. Some simple effective approximations to the 2-poisson model for probabilistic weighted retreival. In *Proc. of SIGIR'94*.

K. Sudo, S. Sekine, and R. Grishman. 2001. Automatic pattern acquisition for japanese information extraction. In *Proc. of HLT-2001*.

TSC. 2001. Proceedings of the Second NTCIR Workshop on Research in Chinese & Japanese Text Retrieval and Text Summarization (NTCIR2). National Institute of Informatics.