NTT's Text Summarization System for DUC-2002

Tsutomu HIRAO, Yutaka SASAKI, Hideki ISOZAKI and **Eisaku MAEDA**
NTT Communication Science Laboratories
2-4, Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0237 Japan
{hirao,isozaki,sasaki,maeda}@cslab.kecl.ntt.co.jp

Abstract

We participated in the Document Understanding Conference 2002 (DUC-2002) in order to confirm the effectiveness of our summarization system based on an important sentence extraction technique. Our system employs the machine learning algorithm, Support Vector Machines, to classify a sentence into an important or an unimportant sentence. The result of the Single-Document Summarization task shows that our system's performance achieved a high grade in coverage metrics.

1 Introduction

A summary made by an important sentence extraction system may lack coherence, but still contain useful information. Therefore, this technique plays an important role in automatic text summarization.

Conventionally, an important sentence extraction method focus on sentence features and define significance scores. The features include key words, sentence position, and certain linguistic clues. Sekine and Nobata (2001) proposed scoring functions to integrate heterogeneous features and showed the effectiveness of the method at DUC-2001. However, it is hard to determine the optimal parameter values manually.

When a large quantity of training data is available, tuning can be effectively realized by machine learning. Aone et al. (1998) and Kupiec et al. (1995) employed Bayesian classifiers, Mani et al. (1998), Lin (1999) used decision tree learning.

We have already applied Support Vector Machines (SVMs) (Vapnik, 1995) to Japanese Single-Document Summarization. We confirmed the effectiveness of our systems(Hirao et al., 2002). In order to confirm performance of our system for English documents, we participated in the Single-Document Summarization task at the Document Understanding Conference 2002 (DUC-2002).

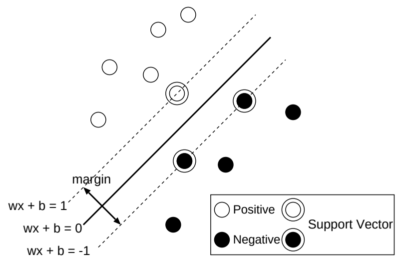

Figure 1: Support Vector Machines.

The remainder of this paper is organized as follows. Section 2 describes our system based on Support Vector Machines. In Section 3, we show the evaluation results at DUC-2002. Finally, Section 4 concludes this paper.

2 Description of our system

2.1 Support Vector Machines (SVMs)

SVM is a supervised learning algorithm for two-class problems. Figure 1 shows the conceptual structure of SVM.

Training data is given by

$$(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_u, y_u), \quad \mathbf{x}_j \in \mathbf{R}^n,\ y_j \in \{+1, -1\}.$$

Here, $\mathbf{x}_j$ is a feature vector of the j-th sample; y_j is its class label, positive $(+1)$ or negative (-1). SVM separates positive and negative examples by a hyperplane given by

$$\mathbf{w} \cdot \mathbf{x} + b = 0,\ \mathbf{w} \in \mathbf{R}^n, b \in \mathbf{R}, \tag{1}$$

In general, such a hyperplane is not unique. The SVM determines the optimal hyperplane by maximizing the margin. The margin is the distance between negative examples and positive examples; the distance between $\mathbf{w} \cdot \mathbf{x} + b = 1$ and $\mathbf{w} \cdot \mathbf{x} + b = -1$. The examples on $\mathbf{w} \cdot \mathbf{x} + b = \pm 1$ are called the Support Vector which represents both positive or negative examples.

Here, the hyperplane must satisfy the following constraints:

$$y_i(\mathbf{w} \cdot \mathbf{x}_j + b) - 1 \geq 0.$$

Hence, the size of the margin is $2/||\mathbf{w}||$. In order to maximize the margin, we assume the following objective function:

$$\begin{aligned} &\underset{\mathbf{w},b}{\text{Minimize}} \quad J(\mathbf{w}) = \frac{1}{2}||\mathbf{w}||^2 \\ &\text{s.t.} \quad y_j(\mathbf{w} \cdot \mathbf{x}_j + b) - 1 \geq 0. \end{aligned} \tag{2}$$

By solving a quadratic programming problem, the decision function $f(\mathbf{x}) = \text{sgn}(g(\mathbf{x}))$ is derived, where

$$g(\mathbf{x}) = \sum_{i=1}^{u} \lambda_i y_i \mathbf{x}_i \cdot \mathbf{x} + b. \tag{3}$$

Since training data is not necessarily linearly separable, slack variables (ξ_j) are introduced for all $\mathbf{x}_j$. These ξ_j give a misclassification error and should satisfy the following inequalities:

$$y_i(\mathbf{w} \cdot \mathbf{x}_j + b) - (1 - \xi_j) \geq 0.$$

Hence, we assume the following objective function to maximize margin:

$$\begin{aligned} &\underset{\mathbf{w},b,\xi}{\text{Minimize}} \quad J(\mathbf{w}, \xi) = \frac{1}{2}||\mathbf{w}||^2 + C\sum_{j=1}^{u} \xi_j \\ &\text{s.t.} \quad y_j(\mathbf{w} \cdot \mathbf{x}_j + b) - (1 - \xi_j) \geq 0. \end{aligned} \tag{4}$$

Here, $||\mathbf{w}||/2$ indicates the size of the margin, $\sum_{j=1}^{u} \xi_j$ indicates the penalty for misclassification, and C is the cost parameter that determines the trade-off for these two arguments. By solving a quadratic programming problem, the decision function $f(\mathbf{x}) = \text{sgn}(g(\mathbf{x}))$ is derived in the same as linear separation (equation (3)).

The decision function depends only on support vectors ($\lambda_i \neq 0$). Training examples, except for support vectors ($\lambda_i = 0$), have no influence on the decision function.

SVMs can handle non-linear decision surfaces by simply substituting every occurrence of the inner product in equation (3) with kernel function $K(\mathbf{x}_i \cdot \mathbf{x})$. Therefore, the decision function can be rewritten as follows:

$$g(\mathbf{x}) = \sum_{i=1}^{u} \lambda_i y_i K(\mathbf{x}_i, \mathbf{x}) + b. \tag{5}$$

Note that the kernel function must satisfy the Mercer's condition.

In this paper, we use polynomial kernel functions, which have been found to be very effective in the study of other tasks in natural language processing (Joachims, 1998; Kudo and Matsumoto, 2001; Kudo and Matsumoto, 2000):

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^d. \tag{6}$$

2.2 Sentence Ranking

Important sentence extraction can be regarded as a two-class problem. However, the proportion of important sentences in training data will differ from that in test data. The number of important sentences in a document is determined by a summarization rate or word limit which is given at run-time. In the Single-Document Summarization task at DUC-2002, the word limit was 100 words. A simple solution to this problem is to rank sentences in a document, then select the top N sentences. We used $g(\mathbf{x})$, the normalized distance from the hyperplane to $\mathbf{x}$ to rank the sentences.

2.3 Features

We define the features discussed below that are associated with sentence S_i by taking past studies into account (Zechner, 1996; Sekine, 2001).

Position of sentences

We define a feature function, Posd, for the position of S_i. Posd is S_i's position in a document. The first sentence obtains the highest score, the last obtains the lowest score:

$$\text{Posd}(S_i) = 1 - \frac{BD(S_i)}{D(S_i)}.$$

Here, $|D(S_i)|$ is the number of characters in the document $D(S_i)$ that contains S_i and $BD(S_i)$ is the number of characters before S_i in $D(S_i)$.

Table 1: Evaluation Results

System-ID	Mean Coverage	Length-Adjusted Coverage	Count of Quality Questions	Mean Score for Quality Questions
15	0.332	0.232	0.986	0.551
16	0.303	0.214	1.441	0.644
17	0.082	0.299	0.758	0.408
18	0.323	0.228	0.997	0.565
19	0.389	0.293	0.698	0.448
21	0.370	0.247	0.885	0.561
23	0.335	0.272	0.582	0.425
25	0.290	0.220	3.200	1.281
Our System	**0.383**	**0.272**	**1.014**	**0.552**
28	0.380	0.261	1.013	0.537
29	0.361	0.251	1.210	0.660
30	0.057	0.339	2.637	1.040
31	0.360	0.240	1.153	0.676
Lead	0.370	0.255	0.718	0.490
Human	0.505	0.336	0.505	0.354

Length of sentences

We define a feature function that addresses the length of sentences as

$$\mathrm{Len}(S_i) = \frac{|S_i|}{\max_{S_z \in D(S_i)} |S_z|}.$$

Here, $|S_i|$ is the number of characters of sentence S_i and $\max_{S_z \in D} |S_z|$ is the maximum number of characters in a sentence that belongs to $D(S_i)$.

Weight of sentences

We defined the feature function that weights sentences based on $TF \cdot IDF$ word weighting as

$$\mathrm{Score}(S_i) = \sum_t tf(t, S_i) \cdot w(t, D(S_i)).$$

Here, $\mathrm{Score}(S_i)$ is the summation of weighting $w(t, D(S_i))$ of words that appear in sentence S_i. In addition, we define word weight $w(t, D(S_i))$ based on $TF \cdot IDF$:

$$w(t, D) = 0.5 \left(1 + \frac{tf(t, D)}{tf_{max}(D)}\right) \cdot \log\left(\frac{|DB|}{df(t)}\right).$$

Here, $tf(t, D)$ is the term frequency of t in D, $tf_{max}(D)$ is the maximum term frequency in D and $df(t)$ is the frequency of documents that contains term t. $|DB|$ is the total number of the documents in database.

We used the terms t that were judged to be noun or unknown by the Part-of-Speech tagger, TreeTagger(Schmid, 1994). The database indicates TIPSTER collection.

Similarity between Headline

We defined feature function $\mathrm{Sim}(S_i)$, which is similarity between headlines of documents that contain S_i, as follows:

$$\mathrm{Sim}(S_i) = \frac{\vec{v}(S_i) \cdot \vec{v}(H)}{\|\vec{v}(S_i)\| \, \|\vec{v}(H)\|}.$$

Here, $\vec{v}(H)$ is a boolean vector in the Vector Space Model (VSM), the elements of which represent terms in the headline, and $\vec{v}(S_i)$ is also a boolean vector the elements of which represent terms in the sentence.

Prepositions

Boolean value 1 is given to sentences that include a certain preposition. The prepositions are decided by TreeTagger.

Verbs

Boolean value 1 is also given to sentences if they include a certain verb. The verbs are also decided by TreeTagger.

3 Results

We trained classifiers by using data at DUC-2001 and classified sentences contained in test data (567 documents). Randomly chosen documents of 295 were evaluated.

Table 1 shows the results of subjective evaluation of 13 systems which participated in the Single-Document Summarization task at DUC-2002 and two reference results. In the table, "Lead" denotes the result of a lead-based baseline system and "Human" denotes the result of human subjects. "Mean Coverage" (MC) and "Length-Adjusted Coverage" (LAC) indicate content based metrics for summaries. The higher score means the better performance. "Count of Quality Questions" (CQ) and "Mean Score for Quality Questions" (MCQ) indicate readability metrics, such as grammaticality, cohesion and organization. The lower score means better performance.

Our system achieved 2nd in MC, 4th in LAC, 8th in CQ and 6th in MCQ. Moreover, our system outperformed Lead in MC and LAC, but was less successful in CQ and MCQ. This result shows that our summaries contain important information but that they have moderate readability because of the lack of cohesion.

4 Conclusion

We described our system based on Support Vector Machines, which participated in the Single-Document Summarization task at DUC-2002 and showed the evaluation results. The results confirm the effectiveness of our system in coverage metrics.

As future work, we would like to introduce other feature such as Named Entities, Modalities, and Rhetorical Relations.

Acknowledgement

We would like to thank Dr. Nobata and Dr. Sekine for helping us to use their algorithm that aligns abstracts with extracts.

References

C. Aone, M. Okurowski, and J. Gorlinsky. 1998. Trainable Scalable Summarization Using Robust NLP and Machine Learning. *Proc. of the 17th COLING and 36th ACL*, pages 62–66.

T. Hirao, H. Isozaki, M. Eisaku, and Y Matsumoto. 2002. Extraction important sentences with supprt vector machines. *Proc. of of the 19th Inter National Conference of Computational Linguistics.*

T. Joachims. 1998. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. *Proc. of ECML*, pages 137–142.

T. Kudo and Y. Matsumoto. 2000. Japane Dependency Structure Analysis Based on Suport Vector Machines. *Proc. of EMNLP and VLC*, pages 18–25.

T. Kudo and Y. Matsumoto. 2001. Chunking with Support Vector Machine. *Proc. of the 2nd NAACL*, pages 192–199.

J. Kupiec, J. Pedersen, and F. Chen. 1995. A Trainable Document Summarizer. *Proc. of the 18th ACM-SIGIR*, pages 68–73.

Chin-Yew Lin. 1999. Training a Selection Function for Extraction. *Proc. of the 18th ACM-CIKM*, pages 55–62.

I. Mani and E. Bloedorn. 1998. Machine Learning of Generic and User-Focused Summarization. *Proc. of the 15th AAAI*, pages 821–826.

G. Schmid. 1994. Treetagger – a language independent part-of-speech tagger.

S. Sekine and C. Nobata. 2001. Sentence extraction with information extraction technique. *Proc. of the DUC2001.*

V. Vapnik. 1995. *The Nature of Statistical Learning Theory.* New York.

K. Zechner. 1996. Fast Generation of Abstracts from General Domain Text Corpora by Extracting Relevant Sentences. *Proc. of the 16th COLING*, pages 986–989.