Learning to Generate Coherent Summary with Discriminative Hidden Semi-Markov Model

Hitoshi Nishikawa[1], **Kazuho Arita**[1], **Katsumi Tanaka**[1],
Tsutomu Hirao[2], **Toshiro Makino**[1] and **Yoshihiro Matsuo**[1]
Nippon Telegraph and Telephone Corporation
[1] 1-1 Hikari-no-oka, Yokosuka-shi, Kanagawa, 239-0847 Japan
[2] 2-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0237 Japan
{nishikawa.hitoshi, arita.kazuho, tanaka.katsumi
hirao.tsutomu, makino.toshiro, matsuo.yoshihiro}@lab.ntt.co.jp

Abstract

In this paper we introduce a novel single-document summarization method based on a hidden semi-Markov model. This model can naturally model single-document summarization as the optimization problem of selecting the best sequence from among the sentences in the input document under the given objective function and knapsack constraint. This advantage makes it possible for sentence selection to take the coherence of the summary into account. In addition our model can also incorporate sentence compression into the summarization process. To demonstrate the effectiveness of our method, we conduct an experimental evaluation with a large-scale corpus consisting of 12,748 pairs of a document and its reference. The results show that our method significantly outperforms the competitive baselines in terms of ROUGE evaluation, and the linguistic quality of summaries is also improved. Our method successfully mimicked the reference summaries, about 20 percent of the summaries generated by our method were completely identical to their references. Moreover, we show that large-scale training samples are quite effective for training a summarizer.

1 Introduction

Single-document summarization is attracting much more attention as a key technology in providing better information access in a commercial context. The Financial Times and CNN have been providing summaries of articles in their websites to attract users, and Summly, which has been acquired by Yahoo!, provided the service of automatically summarizing articles on the Internet. Given the cost of manual summarization, we can greatly improve the information access of Internet users by creating an automatic summarizer that can approach the summarization quality of humans.

To mimic manually-written summaries, one important aspect is coherence (Nenkova and McKeown, 2011). Although coherence has been studied widely in a field of multi-document summarization (Karamanis et al., 2004; Barzilay and Lapata, 2005; Nishikawa et al., 2010; Christensen et al., 2013), it has not been studied enough in the context of single-document summarization. In this paper, we revisit the problem of coherence and employ it to produce both informative and linguistically high-quality summaries.

To obtain such summaries, we introduce a novel summarization method based on a hidden semi-Markov model. The method has the properties of both the popular single-document summarization model, the knapsack problem, which packs the sentences into the given length and the hidden Markov model, which takes summary coherence into account by determining sentence context when selecting sentences. By leveraging this, we can build a summarizer that naturally achieves coherence.

We state the novelty and contributions of this paper as follows:

- We regard single-document summarization as a combinatorial optimization problem modeled by a hidden semi-Markov model and propose an efficient decoding algorithm for the problem.
- We introduce various features related to coherence in a combinatorial formulation. We extend a hidden semi-Markov model to achieve discrimination, so our method can take advantage of many features for predicting coherence.

Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 1648–1659, Dublin, Ireland, August 23-29 2014.

- We show that our large-scale corpus greatly improves the performance of summarization.

This paper is organized as follows. In Section 2, we describe related work. In Section 3, we detail our proposed model. We also explain how the parameters in our model are optimized and how sentences are compressed. In Section 4, we explain how variants of the original sentences are generated. In Section 5, we explain the decoding algorithm for our method. In Section 6, we explain the settings of our experiments, our corpus, and compared methods. In Section 7, we show results of the experiments conducted to evaluate our method. In Section 8, we conclude this paper.

2 Related Work

2.1 Single-Document Summarization

Basically, single-document summarization can be done through sentence selection (Nenkova and McKeown, 2011) . The document to be summarized is decomposed into a set of sentences and then the summarizer selects a subset of the sentences as a summary.

McDonald (2007) pointed out that single-document summarization can be formulated as a well-known combinatorial optimization problem, the knapsack problem. Given a set of sentences together with their lengths and values, the summarizer packs them into a summary so that the total value is as large as possible but the total length is less than or equal to a given maximum summary length. Interestingly, a hidden semi-Markov model (Yu, 2010) can be regarded as a natural extension of the knapsack problem, we take advantage of this property for single-document summarization. We elaborate the relation between the knapsack problem and the hidden semi-Markov model in Section 3.

To generate coherent summaries in single-document summarization, there are two types of approaches[1] : tree-based approaches (Marcu, 1997; Daume and Marcu, 2002; Hirao et al., 2013) and sequence-based approaches (Barzilay and Lee, 2004; Shen et al., 2007). The former rely on the tree representation of a document based on the Rhetorical Structure Theory (RST) (Mann and Thompson, 1988). Basically, the former approaches (Marcu, 1997; Daume and Marcu, 2002; Hirao et al., 2013) trim the tree representation of a document by making use of nucleus-satellite relations among sentences. The advantage of RST-based approaches is that they can take advantage of global information about the documents. However, a drawback is that they depend heavily on the RST parser that is used. Performance is remarkably sensitive to the accuracy of RST parsing, and hence we have to build a good RST parser. Instead of making use of the global structure of the document, the sequence-based methods rely on and take advantage of the local coherence of sentences. As one advantage over the tree-based approaches, the sequence-based approaches do not require tools as RST parsers, and hence they are more robust. For this reason, this paper focuses on sequence-based approaches.

The previous works most closely related to our method are those proposed by Barzilay and Lee (2004) and Shen et al. (2007). Barzilay and Lee built a hidden Markov model to capture the content structure of documents and used it to identify the important sentences. Shen et al. (2007) extended the HMM-based approach to make it discriminative by making use of conditional random fields (Lafferty et al., 2001). Conditional random fields can incorporate various features to identify the importance of a sentence and they showed its effectiveness. A shortcoming of these approaches is that their model only classifies sentences into two classes, it cannot take account of output length directly. This deficiency is problematic because in practical usage the maximum length of a summary is specified by the user; hence, the summarizer should be able to control output length. In contrast to their method, our approach naturally takes the maximum summary length into account when summarizing a document.

2.2 Coherence

In the context of multi-document summarization, coherence has been studied widely. In multi-document summarization, sentences are selected from different documents, and hence some way of ordering the sentences is required. Sentence ordering (Barzilay et al., 2002; Althaus et al., 2004; Karamanis et al.,

[1] As an interesting related work, Clarke and Lapata (2007) compresses documents by making use of Centering Theory (Grosz et al., 1995). However, in their approach, the desired length of an output summary could not be specified and hence they said their method was compression rather than summarization.

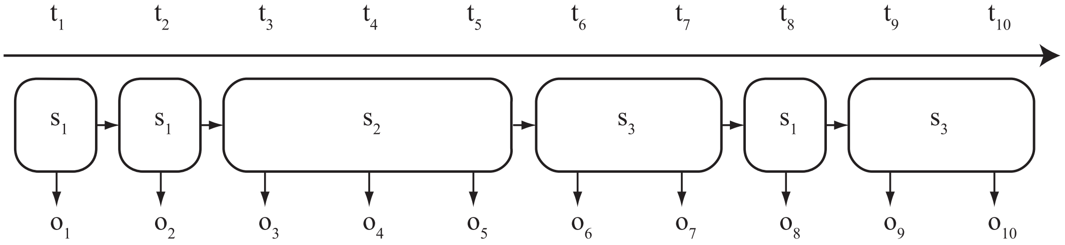

Figure 1: An example of the hidden semi-Markov model. The system observes a sequence consisting of 10 symbols $o_1 ... o_{10}$ over time $t_1 ... t_{10}$ and transitions between states $s_1 ... s_3$. Unlike the basic hidden Markov model, states can persist for a non-unit length. In this figure, state s_2 and state s_3 persist for non-unit lengths. Hence, the system traverses only 6 states despite observing 10 symbols.

2004; Okazaki et al., 2004) is a task to order extracted sentences and is closely related to coherence (Lapata, 2003; Barzilay and Lapata, 2005; Nenkova et al., 2010; Pitler et al., 2010; Louis and Nenkova, 2012). Many effective features have been found out to capture coherence and we utilize these features.

Some work proposed a model that could jointly taking the content of the summary and its coherence into account (Nishikawa et al., 2010; Christensen et al., 2013). Since extracted sentences in multi-document summarization must be ordered, a task that is NP-hard, they relied on integer linear programming (Nishikawa et al., 2010) or a local search strategy (Christensen et al., 2013). The former can locate the optimal solution at a heavy computation cost, while the latter runs quickly but there is no guarantee of locating the optimal solution. In contrast to their trade-off, our proposed algorithm, based on dynamic programming, can locate the optimal solution quickly because the single-document summarization can skip the ordering operation by reproducing the original order of the input sentences.

In this paper, we show that coherence also takes an important role in single-document summarization. We model the coherence between adjacent sentences in the summary by leveraging the hidden semi-Markov model, which can naturally capture the coherence between sentences.

3 Summarization with Hidden Semi-Markov Model

We first introduce the knapsack problem, which can naturally model single-document summarization. Next, we explain the hidden semi-Markov model and show its relationship to the knapsack problem. Then, we elaborate our summarization method.

3.1 Knapsack Problem

The knapsack problem is a type of combinatorial optimization problem (Korte and Vygen, 2008). Given a set of elements, each of which has a score and size, the problem is formulated as the task of finding the best subset in terms of maximizing the sum of their scores under the size limitation. As mentioned above, single-document summarization can be regarded as an instance of the knapsack problem. The best combination of input sentences can be found by calculating the value of each sentence and packing them into a summary through the dynamic programming knapsack algorithm.

3.2 Hidden Semi-Markov Model

The hidden semi-Markov model (HSMM) is an extension of the hidden Markov model (HMM) (Yu, 2010). In the popular hidden Markov model, each state persists for only one unit length. For example, if a system observes 10 discrete symbols, it outputs 10 hidden states. In the HSMM, each state can persist for some unit lengths through the concept of duration. For example, if a system observes 10 discrete symbols and each state persists for two unit lengths, i.e., their duration is 2, the system outputs 5 hidden states. We show an example in Figure 1. The system observes a sequence consisting of 10 symbols $o_1 ... o_{10}$ over time $t_1 ... t_{10}$ and transitions between states $s_1 ... s_3$. Unlike the basic HMM, states can persist for a non-unit length. In this figure, state s_2 and state s_3 persist for a non-unit length. Hence, the system traverses 6 states even though it observes 10 symbols. This property has been utilized for sequential tagging, such as named entity recognition (Sarawagi and Cohen, 2004), scene text recognition (Weinman et al., 2008) and phonetic recognition (Kim et al., 2011).

The hidden semi-Markov model is closely related to the knapsack problem. The length, K, of the observed symbols can be regarded as a knapsack constraint. We can consider that the system tries to *pack* the states of the model into the observed sequence of symbols by transitioning over the states under the knapsack constraint so as to maximize the likelihood. Therefore, the hidden semi-Markov can naturally be used for single-document summarization. Suppose that the document to be summarized consists of 10 sentences and the length of each of them is measured by the number of words. In this case, the system transitions over 10 states corresponding to the 10 sentences until it cannot select any further sentence due to the given length requirement. Since each state persists for the length of the corresponding sentence, the remaining length decreases every time the system transitions to a new state.

While an HMM is basically a generative model, Collins (2002) extended it to create a discriminative model. An HSMM can also be extended to become discriminative model (Sarawagi and Cohen, 2004). Our discriminative HSMM learns through the application of max-margin training.

3.3 Formulation

We consider there are n input sentences $s_1, s_2, ..., s_n$. These sentences have lengths $\ell_1, \ell_2, ..., \ell_n$ and weights $w_1, w_2, ..., w_n$. We assume that a sentence that has a high weight should be present in the output summary. We also consider each sentence, s_i, has m_i variants $s_{i,1}, s_{i,2}, ..., s_{i,m}$, each produced by some sort of sentence compression or paraphrase module. These variants also have lengths $\ell_{i,1}, \ell_{i,2}, ..., \ell_{i,m_i}$ and weights $w_{i,1}, w_{i,2}, ..., w_{i,m_i}$. For simplicity, we hereinafter note the original sentences $s_1, s_2, ..., s_n$ as $s_{1,0}, s_{2,0}, ..., s_{n,0}$. Hence we have original sentence $s_{i,0}$ and variants $s_{i,1}, s_{i,2}, ..., s_{i,m}$. Let $s_{0,0}$ and $s_{n+1,0}$ be special symbols indicating the beginning of a document and the end of a document, respectively. We define coherence $c_{g,h,i,j}$ as the coherence between sentence $s_{g,h}$ and sentence $s_{i,j}$. An output summary is described as a sequence of input sentences, g. Let G be the entire set of sequences that can be constructed from the input sentences, i.e., $g \in G$. Finally, let K be the maximum length of the summary desired. With these notations, our proposed method can be formulated as the following optimization problem:

$$g^* = \operatorname*{argmax}_{g \in G} \sum_{s_{i,j} \in sent(g)} w_{i,j} + \sum_{(s_{g,h}, s_{i,j}) \in adj(g)} c_{g,h,i,j} \qquad (1)$$

$$s.t. \sum_{s_{i,j} \in sent(g)} \ell_{i,j} \leq K, \qquad (2)$$

where $sent(g)$ and $adj(g)$ indicate a set of sentences in g and a set of adjacent sentences in g, respectively. That is, our model tries to find the best sequence of sentences under the knapsack constraint so as to maximize the sum of weights and sentence coherence. In contrast to the common knapsack problem which cannot take the variants and sentence coherence into account, our method, based on the hidden semi-Markov model, does so naturally.

3.4 Parameter Optimization

Here we elaborate how parameters in the model are optimized to achieve the desired summaries. The goal is to determine the value of $w_{i,j}$ for all i, j and $c_{g,h,i,j}$ for all g, h, i, j. We define $w_{i,j}$ and $c_{g,h,i,j}$ as follows:

$$\begin{aligned} w_{i,j} &= \mathbf{w}_w \cdot \mathbf{f}_w(s_{i,j}) & (3) \\ c_{g,h,i,j} &= \mathbf{w}_c \cdot \mathbf{f}_c(s_{g,h}, s_{i,j}), & (4) \end{aligned}$$

where $\mathbf{f}_w$ and $\mathbf{f}_c$ are d_w-dimensional and d_c-dimensional feature vectors for sentences and sentence pairs, respectively, and $\mathbf{w}_w$ and $\mathbf{w}_c$ are d_w-dimensional and d_c-dimensional parameter vectors for sentences and sentence pairs, respectively. The goal of optimization is to determine the values of both vector $\mathbf{w}_w$ and

$\mathbf{w}_c$, given feature function $\mathbf{f}_w$ and $\mathbf{f}_c$. For simplicity, let $\mathbf{s}$ be a summary, let $\mathbf{f} = \langle \mathbf{f}_w, \mathbf{f}_c \rangle$ be a $(d_w + d_c)$-dimensional feature function for the whole summary and let $\mathbf{w} = \langle \mathbf{w}_w, \mathbf{w}_c \rangle$ be a $(d_w + d_c)$-dimensional weight vector. The value that the objective function outputs for summary $\mathbf{s}$ is $\mathbf{w} \cdot \mathbf{f}(\mathbf{s})$.

To optimize the parameter, we employ the Passive-Aggressive algorithm (Crammer, 2006), a widely-used structured learning method. Since the algorithm offers online learning, it can learn the parameter quickly and is easy to implement. To learn the parameter so that the output summary is optimized to the evaluation criteria popular in document summarization research, ROUGE (Lin, 2004), we introduce ROUGE as the loss function. The parameter is estimated by solving the following formula iteratively[2]:

$$\mathbf{w}^{new} = \operatorname*{argmin}_{\mathbf{w}} \frac{1}{2} ||\mathbf{w} - \mathbf{w}^{old}||^2 \tag{5}$$

$$s.t.\ \ \mathbf{w} \cdot \mathbf{f}(\mathbf{r}) - \mathbf{w} \cdot \mathbf{f}(\mathbf{s}) \geq \text{loss}(\mathbf{s}; \mathbf{r}),$$

where $\mathbf{w}^{new}$ is the parameter vector after update, $\mathbf{w}^{old}$ is the parameter vector before update, $\mathbf{r}$ is a reference summary, and loss is the loss function. We define loss as $1 - \text{ROUGE}(\mathbf{s}; \mathbf{r})$. Among the variants of ROUGE, we used ROUGE-1 for the loss function.

3.4.1 Sentence Feature

The features introduced in this section are used to calculate the weights of sentences, $w_{i,j}$.

Term Frequency: Term frequency is a classic feature in document summarization (Luhn, 1958). We calculate the total number of times each content word occurs in the document and then, for each sentence, sum the totals of the content words that appear in the sentence as the value of this feature.
Word: We also use the words and parts-of-speech as features.
Named Entity: Named entities such as a name of person or organization are important. We use named entities and classes as features.
Length: The length of a sentence may indicate the information value of its content. We use the length of a sentence, measured by character number, as a feature.
Position: The position of a sentence is a classically important feature. We use the position of a sentence, the relative position of a sentence, whether the sentence is the first in the document and whether the sentence is the first in a paragraph, the position of the paragraph in which the sentence is, as features.

3.4.2 Coherence Feature

The features introduced in this section are used to calculate sentence coherence, $c_{g,h,i,h}$.

Lexical Transition: Lapata (2003) showed that the structure of the document can be captured by word-pairs consisting of words of two adjacent sentences. We use this feature for capturing the links between two sentences[3]. We build a set of word pairs where one occurs in a precedent sentence and the other occurs in a succeeding one, and use the elements of the set as a feature.
Lexical Cohesion: Pitler et al. (2010) showed that the similarity of two sentences is one of the strongest features for predicting coherence. We reproduce this feature for generating coherent summaries. We calculate cosine similarity between two sentences and use its value as a feature.
Entity Grid: Previous studies showed that Entity Grid (Barzilay and Lapata, 2005) is a strong feature for predicting coherence (Pitler et al., 2010). We also employ this feature for summarization. While the entity vector made from the entity grid was originally defined for whole documents, we build the entity vector for each pair of two sentences because our model is based on the Markovian assumption, and hence the coherence score is defined between two sentences.

[2] As we explain later in Section 5, computation complexity of our algorithm is pseudo-polynomial, and hence the best solution of our model can be located quickly. This is also advantageous in the learning phase because to learn parameters using structured learning, the learner has to generate a summary to calculate the loss. Since our algorithm can quickly find the best solution and generate a summary, it can also contribute to shortening the time required for learning.

[3] It is expected that this feature will also contribute to sentence selection. Barzilay and Elhadad (1997) showed that a closely related word-pair was a good indicator for sentence selection. This feature captures this property by learning.

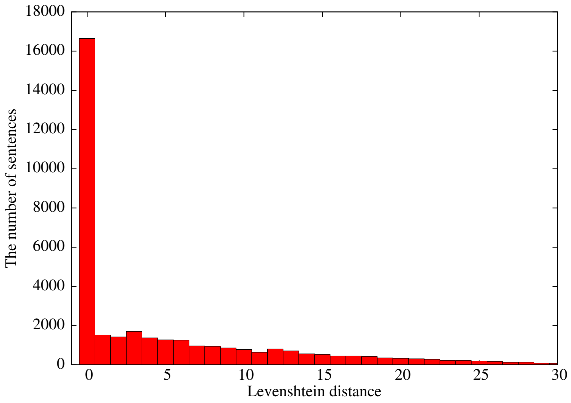

Figure 2: Distribution of Levenshtein distance in the aligned sentences. Among the 36,413 sentences in the references, 16,643 were identical (Levenshtein distance is 0) to the aligned sentences in the input documents.

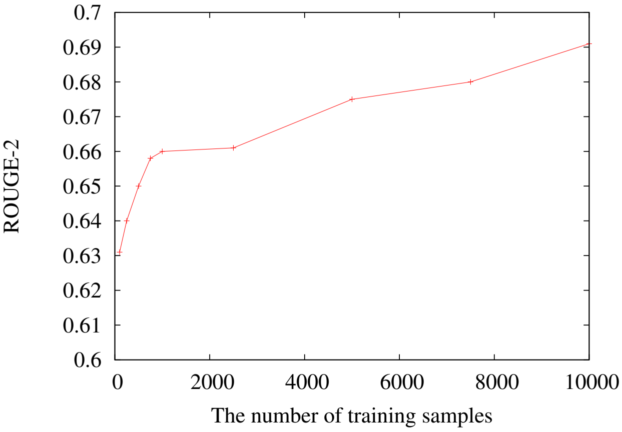

Figure 3: Learning curve of HSMM.

4 Generating Sentence Variants

Since our model can take the variants of an original sentence in the input document as in the multi-candidate reduction framework (Zajic et al., 2007), we incorporate sentence compression.

We generate a few variants of each original sentence by trimming the dependency tree of the sentence; this simple operation is sufficient for reproducing reference summaries. By aligning sentences in a reference summary with those in the corresponding input document[4], we found that human summaries were quite conservative. Among the 36,413 sentences in the references, 16,643 were identical to the aligned sentences in the input documents. Furthermore, most remaining sentences were virtually identical to the original sentences; revisions were minor, and can be reproduced by simple operations. Few sentences exhibited paraphrasing or more sophisticated operations. We plot the distribution of Levenshtein distance in the aligned sentences in Figure 2. According to this observation, we produce the following types of variants by sentence compression:

1. Removing information in parentheses. Some sentences contain parentheses containing additional information for readers. The first type of variant deletes text in parentheses.

2. Shortening sentences by trimming their dependency trees. Basically this method follows the sentence trimmer proposed by Nomoto (2008). While using his method, we keep the predicate and its obligatory arguments in the sentences to keep the sentences grammatical. If a predicate is trimmed, its obligatory arguments are also trimmed and vice versa. Since there are an exponential number of subtrees in one tree, we use only n-best subtrees by ranking them according to n-gram language likelihood and dependency-based language likelihood. We used the dependency parser proposed by Imamura et al (Imamura et al., 2007) to acquire the dependency tree.

5 Decoding with Dynamic Programming

To solve Equation 1 under the constraints of Equation 2, we use dynamic programming. Algorithm 1 shows the pseudo code of the decoding algorithm. Line 1 to Line 7 initializes the variables used in the algorithm. Vector $\mathbf{x} = \langle x_0, ..., x_{n+1} \rangle$ stores which sentence and which variants are included in the output summary. If $x_3 = 2$, $s_{3,2}$ is included in the summary. V, P and S are two-dimensional arrays, each of which is used as a dynamic programming table. They store the process of dynamic programming.

[4] Alignment proceeds in two steps: first, we calculate the Levenshtein distance between sentences in the document and its reference, and then we align sentences so as to minimize the distance between them.

Algorithm 1 Decoding Algorithm: Filling Table

```
1: $\mathbf{x} = \langle x_0, ..., x_{n+1} \rangle$
2: for $i = 0$ to $n + 1$ do
3:     $x_i = -1$
4:     $V[0][i] \leftarrow -1$
5:     $P[0][i] \leftarrow -1$
6:     $S[0][i] \leftarrow 0$
7: $V[0][0] = 0$
8: for $k = 1$ to $K$ do
9:     for $i = 1$ to $n$ do
10:        $V[k][i] \leftarrow V[k-1][i]$
11:        $P[k][i] \leftarrow P[k-1][i]$
12:        $S[k][i] \leftarrow S[k-1][i]$
13:        for $v = 0$ to $m_i$ do
14:            if $\ell_{i,v} \leq k$ then
15:                for $h = 0$ to $i - 1$ do
16:                    $u = V[k - \ell_{i,v}][h]$
17:                    if $u \neq -1 \wedge S[k - \ell_{i,v}][h] + w_{i,v} + c_{h,u,i,v} \geq S[k][i]$ then
18:                        $V[k][i] \leftarrow v$
19:                        $P[k][i] \leftarrow h$
20:                        $S[k][i] \leftarrow S[k - \ell_{i,v}][h] + w_{i,v} + c_{h,u,i,v}$
21: $V[K+1][n+1] \leftarrow 0$
22: $P[K+1][n+1] \leftarrow 0$
23: $S[K+1][n+1] \leftarrow 0$
24: for $h = 1$ to $n$ do
25:     $u = V[K][h]$
26:     if $S[K][h] + c_{h,u,n+1,0} \geq S[K+1][n+1]$ then
27:         $P[K+1][n+1] \leftarrow h$
28:         $S[K+1][n+1] \leftarrow S[K][h] + c_{h,u,n+1,0}$
```

	Document	Reference
Avg. # of characters	476.2	142.0
Avg. # of words	298.6	88.3
Avg. # of sentences	9.7	2.9

Table 1: The statistics of our corpus.

$V[k][i]$ stores which variants are used at time k, i. If $V[k][i] = 0$, original sentence $s_{i,0}$ is selected at time k, i. If $V[k][i] = -1$, no sentence is selected at time k, i. $P[k][i]$ stores a pointer to the sentence connected to the front of the current sentence. $S[k][i]$ stores the value of the objective function at time k, i. Line 8 to Line 36 locates the best sequence of sentences based on the following recurrence formula:

$$S[k][i] = \begin{cases} \max_{h=0...i-1, v=0...m} S[k - \ell_{i,v}][h] + w_{i,v} + c_{h,V[k-\ell_{i,v}][h],i,v} & \text{(A)} \\ S[k-1][i] & \text{(B)}, \end{cases} \tag{6}$$

where case A is: $\ell_{i,v} \leq k \ \wedge S[k-1][i] \leq S[k - \ell_{i,v}][h] + w_{i,v} + c_{h,V[k-\ell_{i,v}][h],i,v}$ and case B is: $otherwise$. This recurrence formula means that at time k, i the best variant to be selected as can be determined at time $k - \ell_{i,v}, h$. Hence, for all $k \in 1...K$ and $i \in 1...n$, the algorithm finds the best sequence of sentences at time k, i. After Algorithm 1 locates the best sequence of sentences by filling the tables, the best sequence can be restored by backtracing along the pointers stored in P. Finally, the algorithm outputs $\mathbf{x}$, which stores which sentences and variants are used in the best sequence. Since this algorithm is based on a dynamic programming knapsack algorithm (Korte and Vygen, 2008), it runs in pseudo-polynomial time. This is a significant advantage over the methods that rely on integer linear programming solvers due to their substantial computation cost.

6 Experiments

6.1 Data

We prepared 12,748 pairs of Japanese newspaper articles and their manually-written reference summaries. This is one of the largest corpus available for single-document summarization research. The length of all references is within 150 characters. All references in the corpus were written by a specialist staff in a Japanese newspaper company and the company sold these summaries for commercial purposes.

We list the statistics of our corpus in Table 1. As shown, the task is to summarize the document in about a third of its original length in terms of the number of words.

6.2 Evaluation Criteria

ROUGE; ROUGE is an automatic evaluation method for automatic summarization proposed by Lin (2004). We used ROUGE-1 and ROUGE-2 to evaluate the summaries. Since our document-reference pairs are written in Japanese, we segmented the sentences into words using the Japanese morphological analyzer developed by Fuchi and Takagi (1998). When calculating the ROUGE score, we used only content words (i.e. nouns, verbs and adjectives) and so excluded function words as stop words.
Linguistic Quality: To evaluate the linguistic quality of the summaries generated by our method, we performed a manual evaluation according to quality questions proposed by the National Institute of Standards and Technology (NIST) (2007)[5]. We randomly sampled 100 summaries from the outputs of each method described below and asked 7 subjects to evaluate the summaries according to the questions. All subjects were Japanese native and none were among the authors. Since the quality questions by NIST (2007) were designed for multi-document summarization, we used 3 of the 5 NIST questions for single-document summarization: grammaticality, referential clarity, and structure/coherence. We also asked the subjects to evaluate overall summary quality.

6.3 Compared Methods

We compared the following 8 methods.

Random: Random method selects sentences in the input document randomly.
Lead: Lead method is a classic baseline in single-document summarization. It only extracts the words from the beginning of the document until the extracted words reach the given length. We simply extracted 150 characters from the beginning of each document.
Knapsack: The knapsack problem can be used as a single-document summarization model (McDonald, 2007). In this baseline, the weight of each sentence was calculated based on the average probabilities of the words in the sentence (Nenkova and Vanderwende, 2005). Then, a summary was generated by solving the knapsack problem.
Knapsack with Supervision: Instead of the average word probabilities used in the above baseline, we used only sentence features $\mathbf{f}_w$ to weigh a sentence.
Conditional Random Fields: Conditional random fields can be used to weigh sentences (Shen et al., 2007). Since CRFs required binary labels in learning, we aligned sentences in an input document with the sentences in its reference as explained in Section 4. We used the probabilities of sentences from CRFs as the weights of the knapsack problem.
Hidden Semi-Markov Model: This is our proposed method without variants of the original sentences. It selected sentences only from the set of original sentences.
Hidden Semi-Markov Model with Compression: This is our proposed method with variants of the original sentences. It selected from among the variants and the original ones.
Human: In the linguistic quality evaluation, we added references to the summaries generated by the above methods to show the upper bound.

When learning, we did 10-fold cross validation. In the experiments, statistical significance was checked by Wilcoxon signed-rank test (Wilcoxon, 1945). To counteract the problem of multiple comparisons, we used the Holm-Bonferroni method (Holm, 1979) to adjust the significance level, α.

7 Results and Discussion

We show the results of our experiment in Table 2 and Table 3. In this section, first we discuss the results of the ROUGE evaluation, and then we discuss the results of the linguistic quality evaluation.

In the ROUGE evaluation, all the compared methods except for RANDOM showed good performance. This is because, as shown in Section 4, many references consisted of sentences identical to the original

[5] Some recent studies have tried to predict the readability of the text automatically (Pitler et al., 2010).

Method	R-1	R-2	Idt.
RANDOM	0.417	0.291	1.2%
LEAD	$0.779^{\mathrm{C,S,U,R}}$	$0.727^{\mathrm{C,S,U,R}}$	4.4%
KP	0.704^{R}	0.611^{R}	9.3%
KP(S)	$0.729^{\mathrm{U,R}}$	$0.647^{\mathrm{U,R}}$	10.4%
CRFs	$0.741^{\mathrm{U,R}}$	$0.675^{\mathrm{S,U,R}}$	11.3%
HSMM	$0.769^{\mathrm{C,S,U,R}}$	$0.703^{\mathrm{C,S,U,R}}$	15.2%
HSMM(C)	$0.785^{\mathrm{C,S,U,R}}$	$0.722^{\mathrm{C,S,U,R}}$	20.4%

Table 2: Results of the ROUGE evaluation. "R-1" and "R-2" correspond to ROUGE-1 and ROUGE-2, respectively. The values in the column of "Idt." are the percentage of summaries completely-identical to the corresponding references. In the table, $^{\mathrm{C,S,U,L,R}}$ indicate statistical significance against CRFs, KP(S), KP, LEAD, RANDOM, respectively.

Method	Gram.	Ref.	S./C.	Overall
LEAD	1.9	3.9	2.5	2.1
KP	4.1^{L}	3.7	3.4	3.5
KP(S)	4.2^{L}	3.6	3.5	3.6^{L}
CRFs	4.1^{L}	3.9	3.7^{L}	3.6^{L}
HSMM	4.3^{L}	4.0	4.1^{L}	4.0^{L}
HSMM(C)	4.0^{L}	3.9	4.0^{L}	3.9^{L}
HUMAN	4.7^{L}	4.5	4.7^{L}	4.8^{L}

Table 3: Results of the linguistic quality evaluation. The values ranged from 1 (very poor) to 5 (very good) (National Institute of Standards and Technology, 2007). We show statistical significance with the same notations as Table 2.

ones, and hence the references can be reproduced if important sentences are identified. Since the compression rate in our corpus was relatively light, it made important information easy to identify. Among the compared methods, both LEAD and our proposed method, HSMM(C), achieved the best result. There was no significant difference between LEAD and HSMM(C). This surprising performance of LEAD was due to the ROUGE evaluation. The words in the document leads were likely to be important, and LEAD drew on this property. However, as we mentioned later, it sacrificed the linguistic quality to achieve the high ROUGE score. Furthermore, it failed to yield summaries identical to the reference. In contrast to LEAD, almost 20% of the summaries generated by HSMM(C) were identical to the references. This shows that our method successfully mimicked human assessments. HSMM followed the best models. There was a statistically significant difference between HSMM(C) and HSMM. Since some sentences, especially the first sentence in the document, were long and the first sentence was particularly important to summarize the document, sentence compression yielded a significant improvement. As shown in Table 2, employing compression greatly improved the percentage of identical summaries. HSMM significantly outperformed all of the baseline extractive methods except LEAD. While CRFs can take advantage of all features used in HSMM, CRFs cannot take the evaluation measure such as ROUGE and the knapsack constraint into account in learning. HSMM also significantly outperformed KP(S). This difference is particularly important, and shows the usefulness of features related to coherence. While KP(S) used only features about sentences, HSMM successfully mimicked the references as it drew on the features related to coherence.

We show the learning curve of HSMM in Figure 3. We fixed 2,748 pairs for testing, and learned parameters from 100, 250, 500, 1,000, 2,500, 5,000, 7,500 and 10,000 pairs. The curve in the figure clearly shows the effectiveness of our large-scale corpus in learning. It seems that the curve does not saturate and hence HSMM performance can be improved by more training samples. As in the results recently shown by Filippova (2013), this result implies that large-scale data is important in the field of document summarization as in other fields of computational linguistics. Past studies in document summarization relied on relatively small datasets consisting of a few dozen or at most a few hundred pairs of a document and its reference in learning. In contrast to the past studies, there are over 10,000 pairs in our dataset and the results show its effectiveness.

Second, we discuss the result of the linguistic quality evaluation. Unlike the ROUGE evaluation, HSMM achieved the best result. As previous studies have pointed out (Nenkova and McKeown, 2011), sentence compression commonly tends to degrade the linguistic quality of a summary while improving its content. As shown in Table 3, the grammaticality of HSMM(C) is lower than that of HSMM, but the difference is not significant. Although we could not observe any significant difference between HSMM and other extractive baselines, our proposals, HSMM and HSMM(C), yielded the best result in terms of structure/coherence. By making use of the features related to coherence, we successfully improved summary quality. In contrast to the surprising performance of LEAD in the ROUGE evaluation, in the linguistic quality evaluation, LEAD yielded the worst performance. Since LEAD had to cut the sentences when it reached the given length, it create ungrammatical fragments.

Finally, we touch on the balance between the quality of content and linguistic quality. Comparing Table 2 to 3, we can see the correlation between the quality of content and linguistic quality. This result is reasonable because we can extract much more information from grammatical and well-organized sentences. Although we optimized the parameter to maximize the ROUGE score, it also yielded improvements in linguistic quality. This is because the manually-generated reference summaries are basically grammatical and well-organized and the parameter is learnt to mimic them. However, there is an inherent trade-off between the quality of content and linguistic quality. For example, under stricter length limitations, instead of cohesive devices such as conjunctions, which can improve the coherence of sentences, content words would be preferred for summary inclusion to augment information. Balancing them to maximize reader satisfaction is an interesting problem.

8 Conclusions

In this paper we presented a novel single-document summarization method based on the hidden semi-Markov model, which is a natural extension of the knapsack problem. Our model naturally takes account of sentence context when identifying important sentences. This property is particularly important to ensure the coherence of output summaries and to produce informative and linguistically high-quality summaries. We also proposed an algorithm based on dynamic programming so the best solution can be located quickly. Experiments on a very large-scale single-document summarization corpus showed that our proposed method significantly outperforms competitive baselines.

As future work, we plan to tackle on the summarization task where higher compression is demanded. To generate shorter summaries, we plan to employ more sophisticated approaches, such as paraphrasing.

Acknowledgement

The corpus used in this paper is owned by The Mainichi Newspapers Co., Ltd. and is leased to Nippon Telegraph and Telephone Corporation. We sincerely appreciate their consideration. We also appreciate the insightful comments from reviewers. Their comments greatly improved the quality of this paper.

References

Ernst Althaus, Nikiforos Karamanis, and Alexander Koller. 2004. Computing locally coherent discourses. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL)*, pages 399–406.

Regina Barzilay and Michael Elhadad. 1997. Using lexical chains for text summarization. In *Proceedings of the Intelligent Scalable Text Summarization Workshop (ISTS)*, pages 10–17.

Regina Barzilay and Mirella Lapata. 2005. Modeling local coherence: an entity-based approach. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL)*, pages 141–148.

Regina Barzilay and Lillian Lee. 2004. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *HLT-NAACL 2004: Main Proceedings*, pages 113–120.

Regina Barzilay, Noemie Elhadad, and Kathleen R. McKeown. 2002. Inferring strategies for sentence ordering in multidocument news summarization. *Journal of Artificial Intelligence Research*, 17:35–55.

Janara Christensen, Mausam, Stephen Soderland, and Oren Etzioni. 2013. Towards coherent multi-document summarization. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1163–1173.

James Clarke and Mirella Lapata. 2007. Modelling compression with discourse constraints. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1–11.

Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1–8.

Koby Crammer. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7(Mar):551–585.

Hal Daume, III and Daniel Marcu. 2002. A noisy-channel model for document compression. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 449–456.

Katja Filippova. 2013. Overcoming the lack of parallel data in sentence compression. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1481–1491.

Takeshi Fuchi and Shinichiro Takagi. 1998. Japanese morphological analyzer using word co-occurrence: Jtag. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics (ACL-COLING)*, pages 409–413.

Barbara J. Grosz, Scott Weinstein, and Aravind K. Joshi. 1995. Centering: a framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2):203–225.

Tsutomu Hirao, Yasuhisa Yoshida, Masaaki Nishino, Norihito Yasuda, and Masaaki Nagata. 2013. Single-document summarization as a tree knapsack problem. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1515–1520.

Sture Holm. 1979. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70.

Kenji Imamura, Genichiro Kikui, and Norihito Yasuda. 2007. Japanese dependency parsing using sequential labeling for semi-spoken language. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 225–228.

Nikiforos Karamanis, Massimo Poesio, Chris Mellish, and Jon Oberlander. 2004. Evaluating centering-based metrics of coherence. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL)*, pages 391–398.

Sungwoong Kim, Sungrack Yun, and Chang D. Yoo. 2011. Large margin discriminative semi-markov model for phonetic recognition. *IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING*, 7(19):1999–2012.

Bernhard Korte and Jens Vygen. 2008. *Combinatorial Optimization*. Springer-Verlag, third edition.

John Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning (ICML)*, pages 282–289.

Mirella Lapata. 2003. Probabilistic text structuring: Experiments with sentence ordering. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 545–552.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Proceedings of ACL Workshop Text Summarization Branches Out*, pages 74–81.

Annie Louis and Ani Nenkova. 2012. A coherence model based on syntactic patterns. In *Proceedings of the 2012 Conference on Empirical Methods on Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

Hans P. Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 22(2):159–165.

William C. Mann and Sandra A Thompson. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281.

Daniel Marcu. 1997. From discourse structure to text summaries. In *Proceedings of ACL/EACL 1997 Summarization Workshop*, pages 82–88.

Ryan McDonald. 2007. A study of global inference algorithms in multi-document summarization. In *Proceedings of the 29th European Conference on Information Retrieval (ECIR)*, pages 557–564.

National Institute of Standards and Technology. 2007. The linguistic quality questions. http://www-nlpir.nist.gov/projects/duc/duc2007/quality-questions.txt.

Ani Nenkova and Kathleen McKeown. 2011. *Automatic Summarization*. Now Publishers.

Ani Nenkova and Lucy Vanderwende. 2005. The impact of frequency on summarization. Technical report, MSR-TR-2005-101.

Ani Nenkova, Jieun Chae, Annie Louis, and Emily Pitler. 2010. Structural features for predicting the linguistic quality of text: Applications to machine translation, automatic summarization and human-authored text. In Emiel Krahmer and Theunem Mariet, editors, *Empirical Methods in Natural Language Generation: Data-oriented Methods and Empirical Evaluation*, pages 222–241. Springer.

Hitoshi Nishikawa, Takaaki Hasegawa, Yoshihiro Matsuo, and Genichiro Kikui. 2010. Opinion summarization with integer linear programming formulation for sentence extraction and ordering. In *Coling 2010: Posters*, pages 910–918.

Tadashi Nomoto. 2008. A generic sentence trimmer with crfs. In *Proceedings of the 46th Annual Conference of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, pages 299–307.

Naoaki Okazaki, Yutaka Matsuo, and Mitsuru Ishizuka. 2004. Improving chronological sentence ordering by precedence relation. In *Proceedings of the 20th International Conference on Computational Linguistics (Coling)*, pages 750–756.

Emily Pitler, Annie Louis, and Ani Nenkova. 2010. Automatic evaluation of linguistic quality in multi-document summarization. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 544–554.

Sunita Sarawagi and William W. Cohen. 2004. Semi-markov conditional random fields for information extraction. In *Advances in Neural Information Processing Systems 17*, pages 1185–1192.

Dou Shen, Jian-Tao Sun, Hua Li, Qiang Yang, and Zheng Chen. 2007. Document summarization using conditional random fields. In *Proceedings of the 20th international joint conference on Artifical intelligence (IJCAI)*, pages 2862–2867.

Jerod J. Weinman, Erik Learned-Miller, and Allen Hanson. 2008. A discriminative semi-markov model for robust scene text recognition. In *Proceedings of the 19th International Conference on Pattern Recognition (ICPR)*, pages 1–5.

Frank Wilcoxon. 1945. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83.

Shun-Zheng Yu. 2010. Hidden semi-markov models. *Artificial Intelligence*, 174(2):215–243.

David M. Zajic, Bonnie J. Dorr, Jimmy Lin, and Schwartz Richard. 2007. Multi-candidate reduction: Sentence compression as a tool for document summarization tasks. *Information Processing and Management*, 43:1549–1570.