Opinion Summarization with Integer Linear Programming Formulation for Sentence Extraction and Ordering

Hitoshi Nishikawa, Takaaki Hasegawa, Yoshihiro Matsuo and **Genichiro Kikui**
NTT Cyber Space Laboratories, NTT Corporation
{ nishikawa.hitoshi, hasegawa.takaaki
  matsuo.yoshihiro, kikui.genichiro } @lab.ntt.co.jp

Abstract

In this paper we propose a novel algorithm for opinion summarization that takes account of content and coherence, simultaneously. We consider a summary as a sequence of sentences and directly acquire the optimum sequence from multiple review documents by extracting and ordering the sentences. We achieve this with a novel Integer Linear Programming (ILP) formulation. Our proposed formulation is a powerful mixture of the Maximum Coverage Problem and the Traveling Salesman Problem, and is widely applicable to text generation and summarization tasks. We score each candidate sequence according to its content and coherence. Since our research goal is to summarize reviews, the content score is defined by opinions and the coherence score is developed in training against the review document corpus. We evaluate our method using the reviews of commodities and restaurants. Our method outperforms existing opinion summarizers as indicated by its ROUGE score. We also report the results of human readability experiments.

1 Introduction

The Web now holds a massive number of reviews describing the opinions of customers about products and services. These reviews can help the customer to reach purchasing decisions and guide the business activities of companies such as product improvement. It is, however, almost impossible to read all reviews given their sheer number.

Automatic text summarization, particularly opinion summarization, is expected to allow all possible reviews to be efficiently utilized. Given multiple review documents, our summarizer outputs text consisting of ordered sentences. A typical summary is shown in Table 1. This task is considered as multidocument summarization.

This restaurant offers customers a delicious menu and a relaxing atmosphere. The staff are very friendly but the price is a little high.

Table 1: A typical summary.

Existing summarizers focus on organizing sentences so as to include important information in the given document into a summary under some size limitation. A serious problem is that most of these summarizers completely ignore coherence of the summary, which improves reader's comprehension as reported by Barzilay et al. (2002).

To make summaries coherent, the extracted sentences must be appropriately ordered. However, most summarization systems delink sentence extraction from sentence ordering, so a sentence can be extracted that can never be ordered naturally with the other extracted sentences. Moreover, due to recent advances in decoding techniques for text summarization, the summarizers tend to select shorter sentences to optimize summary content. It aggravates this problem.

Although a preceding work tackles this problem by performing sentence extraction and ordering simultaneously (Nishikawa et al., 2010), they adopt beam search and dynamic programming to search for the optimal solution, so their proposed method may fail to locate it.

To overcome this weakness, this paper proposes a novel Integer Linear Programming (ILP) formulation for searching for the optimal solution efficiently. We formulate the multidocument summarization task as an ILP problem that tries to optimize the content and coherence of the summary by extracting and ordering sentences simultaneously. We apply our method to opinion summarization and show that it outperforms state-of-the-art opinion summarizers in terms of ROUGE evaluations. Although in this paper we challenge

Coling 2010: Poster Volume, pages 910–918,
Beijing, August 2010

our method with opinion summarization, it can be widely applied to other text generation and summarization tasks.

This paper is organized as follows: Section 2 describes related work. Section 3 describes our proposal. Section 4 reports our evaluation experiments. We conclude this paper in Section 5.

2 Related Work

2.1 Sentence Extraction

Although a lot of summarization algorithms have been proposed, most of them solely extract sentences from a set of sentences in the source document set. These methods perform *extractive summarization* and can be formalized as follows:

$$\hat{S} = \underset{S \subseteq T}{\operatorname{argmax}} \mathcal{L}(S) \quad (1)$$
$$\text{s.t. } \operatorname{length}(S) \leq K$$

T stands for all sentences in the source document set and S is an arbitrary subset of T. $\mathcal{L}(\mathcal{S})$ is a function indicating the score of S as determined by one or more criteria. $\operatorname{length}(S)$ indicates the length of S, K is the maximum size of the summary. That is, most summarization algorithms search for, or decode, the set of sentences $\hat{S}$ that maximizes function $\mathcal{L}$ under the given maximum size of the summary K. Thus most studies focus on the design of function $\mathcal{L}$ and efficient search algorithms (i.e. argmax operation in Eq.1).

Objective Function

Many useful $\mathcal{L}$ functions have been proposed including the cosine similarity of given sentences (Carbonell and Goldstein, 1998) and centroid (Radev et al., 2004); some approaches directly learn function $\mathcal{L}$ from references (Kupiec et al., 1995; Hirao et al., 2002).

There are two approaches to defining the score of the summary. One defines the weight on each sentence forming the summary. The other defines a weight for a sub-sentence, *concept*, that the summary contains.

McDonald (2007) and Martins and Smith (2009) directly weight sentences and use MMR to avoid redundancy (Carbonell and Goldstein, 1998). In contrast to their approaches, we set weights on concepts, not sentences. Gillick and Favre (2009) reported that the concept-based model achieves better performance and scalability than the sentence-based model when it is formulated as ILP.

There is a wide range of choice with regard to the unit of the concept. Concepts include words and the relationship between named entities (Filatova and Hatzivassiloglou, 2004), bigrams (Gillick and Favre, 2009), and word stems (Takamura and Okumura, 2009).

Some summarization systems that target reviews, opinion summarizers, extract particular information, *opinion*, from the input sentences and leverage them to select important sentences (Carenini et al., 2006; Lerman et al., 2009). In this paper, since we aim to summarize reviews, the objective function is defined through opinion as the concept that the reviews contain. We explain our detailed objective function in Section 3. We describe features of above existing summarizers in Section 4 and compare our method to them as baselines.

Decoding Method

The algorithms proposed for argmax operation include the greedy method (Filatova and Hatzivassiloglou, 2004), stack decoding (Yih et al., 2007; Takamura and Okumura, 2009) and Integer Linear Programming (Clarke and Lapata, 2007; McDonald, 2007; Gillick and Favre, 2009; Martins and Smith, 2009). Gillick and Favre (2009) and Takamura and Okumura (2009) formulate summarization as a Maximum Coverage Problem. We also use this formulation. While these methods focus on extracting a set of sentences from the source document set, our method performs extraction and ordering simultaneously.

Some studies attempt to generate a single sentence (i.e. headline) from the source document (Banko et al., 2000; Deshpande et al., 2007). While they extract and order *words* from the source document as a unit, our model uses the unit of *sentences*. This problem can be formulated as the Traveling Salesman Problem and its variants. Banko et al. (2000) uses beam search to identify approximate solutions. Deshpande et al. (2007) uses ILP and a randomized algorithm to find the optimal solution.

2.2 Sentence Ordering

It is known that the readability of a collection of sentences, a summary, can be greatly improved by appropriately ordering them (Barzilay et al., 2002). Features proposed to create the appropriate order include publication date of document (Barzilay et al., 2002), content words (Lapata, 2003; Althaus et al., 2004), and syntactic role of

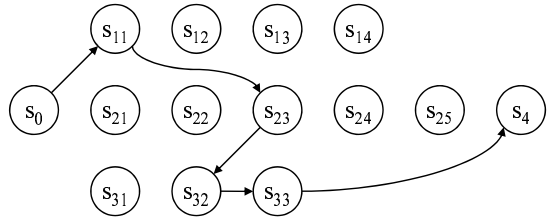

Figure 1: Graph representation of summarization.

words (Barzilay and Lapata, 2005). Some approaches use machine learning to integrate these features (Soricut and Marcu, 2006; Elsner et al., 2007). Generally speaking, these methods score the discourse coherence of a fixed set of sentences. These methods are separated from the extraction step so they may fail if the set includes sentences that are impossible to order naturally.

As mentioned above, there is a preceding work that attempted to perform sentence extraction and ordering simultaneously (Nishikawa et al., 2010). Differences between this paper and that work are as follows:

- This work adopts ILP solver as a decoder. ILP solver allows the summarizer to search for the optimal solution much more rapidly than beam search (Deshpande et al., 2007), which was adopted by the prior work. To permit ILP solver incorporation, we propose in this paper a totally new ILP formulation. The formulation can be widely used for text summarization and generation.
- Moreover, to learn better discourse coherence, we adopt the Passive-Aggressive algorithm (Crammer et al., 2006) and use Kendall's tau (Lapata, 2006) as the loss function. In contrast, the above work adopts Averaged Perceptron (Collins, 2002) and has no explicit loss function.

These advances make this work very different from that work.

3 Our Method

3.1 The Model

We consider a summary as a sequence of sentences. As an example, document set $D = \{d_1, d_2, d_3\}$ is given to a summarizer. We define d as a single document. Document d_1, which consists of four sentences, is describe by $d_1 = \{s_{11}, s_{12}, s_{13}, s_{14}\}$. Documents d_2 and d_3 consist of five sentences and three sentences (i.e. $d_2 = \{s_{21}, s_{22}, s_{23}, s_{24}, s_{25}\}$, $d_3 = \{s_{31}, s_{32}, s_{33}\}$). If the summary consists of four sentences $s_{11}, s_{23}, s_{32}, s_{33}$ and they are ordered as $s_{11} \rightarrow s_{23} \rightarrow s_{32} \rightarrow s_{33}$, we add symbols indicating the beginning of the summary s_0 and the end of the summary s_4, and describe the summary as $S = \langle s_0, s_{11}, s_{23}, s_{32}, s_{33}, s_4 \rangle$. Summary S can be represented as a directed path that starts at s_0 and ends at s_4 as shown in Fig. 1.

	e_1	e_2	e_3	$\ldots$	e_6	e_7	e_8
s_{11}	1	0	0		1	0	0
s_{12}	0	1	0		0	0	0
s_{13}	0	0	0		0	0	1
$\vdots$				$\ddots$			
s_{31}	0	0	0		0	0	0
s_{32}	0	0	1		0	1	0
s_{33}	0	0	0		0	0	1

Table 2: Sentence-Concept Matrix.

We describe a directed arc between s_i and s_j as $a_{i,j} \in A$. The directed path shown in Fig. 1 is decomposed into nodes, $s_0, s_{11}, s_{23}, s_{32}, s_{33}, s_4$, and arcs, $a_{0,11}, a_{11,23}, a_{23,32}, a_{32,33}, a_{33,4}$.

To represent the discourse coherence of two adjacent sentences, we define weight $c_{i,j} \in C$ as the coherence score on the directed arc $a_{i,j}$. We assume that better summaries have higher coherence scores, i.e. if the sum of the scores of the arcs $\sum_{a_{i,j} \in S} c_{i,j} a_{i,j}$ is high, the summary is coherent.

We also assume that the source document set D includes set of concepts $e \in E$. Each concept e is covered by one or more of the sentences in the document set. We show this schema in Table 2. According to Table 2, document set D has eight concepts $e_1, e_2, \ldots, e_7, e_8$ and sentence s_{11} includes concepts e_1 and e_6 while sentence s_{12} includes e_2.

We consider each concept e_i has a weight w_i. We assume that concept e_i will have high weight w_i if it is important. This paper improves summary quality by maximizing the sum of these weights.

We define, based on the above assumption, the following objective function:

$$\mathcal{L}(S) = \sum_{e_i \in S} w_i e_i + \sum_{a_{i,j} \in S} c_{i,j} a_{i,j} \quad (2)$$
$$\text{s.t. } \text{length}(S) \leq K$$

Summarization is, in this paper, realized by maximizing the sum of weights of concepts included in the summary and the coherence score of all adjacent sentences in the summary under the

limit of maximum summary size. Note that while S and T represents the *set* of sentences in Eq.1, they represent the *sequence* of sentences in Eq.2.

Maximizing Eq.2 is NP-hard. If each sentence in the source document set has one concept (i.e. Table 2 is a diagonal matrix), Eq.2 becomes the Prize Collecting Traveling Salesman Problem (Balas, 1989). Therefore, a highly efficient decoding method is essential.

3.2 Parameter Estimation

Our method requires two parameters: weights $w \in W$ of concepts and coherence $c \in C$ of two adjacent sentences. We describe them here.

Content Score

In this paper, as mentioned above, since we attempt to summarize reviews, we adopt *opinion* as a concept. We define opinion $e = \langle t, a, p \rangle$ as the tuple of *target* t, *aspect* a and its *polarity* $p \in \{-1, 0, 1\}$. We define target t as the target of an opinion. For example, the target t of the sentence "This digital camera has good image quality." is *digital camera*. We define aspect a as a word that represents a standpoint appropriate for evaluating products and services. With regard to digital cameras, aspects include *image quality*, *design* and *battery life*. In the above example sentence, the aspect is *image quality*. Polarity p represents whether the opinion is positive or negative. In this paper, we define $p = -1$ as negative, $p = 0$ as neutral and $p = 1$ as positive. Thus the example sentence contains opinion $e = \langle digital\ camera, image\ quality, 1 \rangle$.

Opinions are extracted using a sentiment expression dictionary and pattern matching from dependency trees of sentences. This opinion extractor is the same as that used in Nishikawa et al. (2010).

As the weight w_i of concept e_i, we use only the frequency of each opinion in the input document set, i.e. we assume that an opinion that appears frequently in the input is important. While this weighting is relatively naive compared to Lerman et al. (2009)'s method, our ROUGE evaluation shows that this approach is effective.

Coherence Score

In this section, we define coherence score c. Since it is not easy to model the global coherence of a set of sentences, we approximate the global coherence by the sum of local coherence i.e. the sum of coherence scores of sentence pairs. We define local coherence score $c_{i,j}$ of two sentences $x = \{s_i, s_j\}$ and their order $y = \langle s_i, s_j \rangle$ representing $s_i \rightarrow s_j$ as follows:

$$c_{i,j} = \mathbf{w} \cdot \phi(x, y) \tag{3}$$

$\mathbf{w} \cdot \phi(x, y)$ is the inner product of $\mathbf{w}$ and $\phi(x, y)$, $\mathbf{w}$ is a parameter vector and $\phi(x, y)$ is a feature vector of the two sentences s_i and s_j.

Since coherence consists of many different elements and it is difficult to model all of them, we approximate the features of coherence as the Cartesian product of the following features: content words, POS tags of content words, named entity tags (e.g. `LOC`, `ORG`) and conjunctions. Lapata (2003) proposed most of these features.

We also define feature vector $\Phi(\mathbf{x}, \mathbf{y})$ of the bag of sentences $\mathbf{x} = \{s_0, s_1, \ldots, s_n, s_{n+1}\}$ and its entire order $\mathbf{y} = \langle s_0, s_1, \ldots, s_n, s_{n+1} \rangle$ as follows:

$$\Phi(\mathbf{x}, \mathbf{y}) = \sum_{x,y} \phi(x, y) \tag{4}$$

Therefore, the score of order $\mathbf{y}$ is $\mathbf{w} \cdot \Phi(\mathbf{x}, \mathbf{y})$. Given a training set, if trained parameter vector $\mathbf{w}$ assigns score $\mathbf{w} \cdot \Phi(\mathbf{x}, \mathbf{y}_t)$ to correct order $\mathbf{y}_t$ that is higher than score $\mathbf{w} \cdot \Phi(\mathbf{x}, \hat{\mathbf{y}})$ assigned to incorrect order $\hat{\mathbf{y}}$, it is expected that the trained parameter vector will give a higher score to coherently ordered sentences than to incoherently ordered sentences.

We use the Passive-Aggressive algorithm (Crammer et al., 2006) to find $\mathbf{w}$. The Passive-Aggressive algorithm is an online learning algorithm that updates the parameter vector by taking up one example from the training examples and outputting the solution that has the highest score under the current parameter vector. If the output differs from the training example, the parameter vector is updated as follows;

$$\begin{gathered} \min ||\mathbf{w}^{i+1} - \mathbf{w}^{i}|| \\ \text{s.t. } \mathrm{s}(\mathbf{x}, \mathbf{y}_t; \mathbf{w}^{i+1}) - \mathrm{s}(\mathbf{x}, \hat{\mathbf{y}}; \mathbf{w}^{i+1}) \geq \ell(\hat{\mathbf{y}}; \mathbf{y}_t) \\ \mathrm{s}(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \mathbf{w} \cdot \Phi(\mathbf{x}, \mathbf{y}) \end{gathered} \tag{5}$$

$\mathbf{w}^i$ is the current parameter vector and $\mathbf{w}^{i+1}$ is the updated parameter vector. That is, Eq.5 means that the score of the correct order must exceed the score of an incorrect order by more than loss function $\ell(\hat{\mathbf{y}}; \mathbf{y}_t)$ while minimizing the change in parameters.

When updating the parameter vector, this algorithm requires the solution that has the highest score under the current parameter vector, so we have to run an argmax operation. Since we are

attempting to order a set of sentences, the operation is regarded as solving the Traveling Salesman Problem (Althaus et al., 2004); that is, we locate the path that offers the maximum score through all n sentences where s_0 and s_{n+1} are starting and ending points, respectively. This operation is NP-hard and it is difficult to find the global optimal solution. To overcome this, we find an approximate solution by beam search.[1]

We define loss function $\ell(\hat{\mathbf{y}}; \mathbf{y}_t)$ as follows:

$$\ell(\hat{\mathbf{y}}; \mathbf{y}_t) = 1 - \tau \quad (6)$$

$$\tau = 1 - 4\frac{\mathrm{S}(\hat{\mathbf{y}}, \mathbf{y}_t)}{N(N-1)} \quad (7)$$

τ indicates Kendall's tau. $\mathrm{S}(\hat{\mathbf{y}}, \mathbf{y}_t)$ is the minimum number of operations that swap adjacent elements (i.e. sentences) needed to bring $\hat{\mathbf{y}}$ to $\mathbf{y}_t$ (Lapata, 2006). N indicates the number of elements. Since Lapata (2006) reported that Kendall's tau reliably reproduces human ratings with regard to sentence ordering, using it to minimize the loss function is expected to yield more reliable parameters.

We omit detailed derivations due to space limitations. Parameters are updated as per the following equation.

$$\mathbf{w}^{i+1} = \mathbf{w}^i + \eta^i(\Phi(\mathbf{x}, \mathbf{y}_t) - \Phi(\mathbf{x}, \hat{\mathbf{y}})) \quad (8)$$

$$\eta^i = \frac{\ell(\hat{\mathbf{y}}; \mathbf{y}_t) - \mathrm{s}(\mathbf{x}, \mathbf{y}_t; \mathbf{w}^i) + \mathrm{s}(\mathbf{x}, \hat{\mathbf{y}}; \mathbf{w}^i)}{||\Phi(\mathbf{x}, \mathbf{y}_t) - \Phi(\mathbf{x}, \hat{\mathbf{y}})||^2 + \frac{1}{2C}} \quad (9)$$

C in Eq.9 is the *aggressiveness parameter* that controls the degree of parameter change.

Note that our method learns $\mathbf{w}$ from documents automatically annotated by a POS tagger and a named entity tagger. That is, manual annotation isn't required.

3.3 Decoding with Integer Linear Programming Formulation

This section describes an ILP formulation of the above model. We use the same notation convention as introduced in Section 3.1. We use $s \in S, a \in A, e \in E$ as the decision variable. Variable $s_i \in S$ indicates the inclusion of the i th sentence. If the i th sentence is part of the summary, then s_i is 1. If it is not part of the summary, then s_i is 0. Variable $a_{i,j} \in A$ indicates the adjacency of the i th and j th sentences. If these two sentences are ordered as $s_i \rightarrow s_j$, then $a_{i,j}$ is 1. Variable $e_i \in E$ indicates the inclusion of the i th concept e_i. Taking Fig.1 as an example, variables $s_0, s_{11}, s_{23}, s_{32}, s_{33}, s_4$ and $a_{0,11}, a_{11,23}, a_{23,32}, a_{32,33}, a_{33,4}$ are 1. e_i, which correspond to the concepts in the above extracted sentences, are also 1.

We represent the above objective function (Eq.2) as follows:

$$\max\left\{\lambda \sum_{e_i \in E} w_i e_i + (1-\lambda) \sum_{a_{i,j} \in A} c_{i,j} a_{i,j}\right\} \quad (10)$$

Eq.10 attempts to cover as much of the concepts included in input document set as possible according to their weights $w \in W$ and orders sentences according to discourse coherence $c \in C$. λ is a scaling factor to balance w and c.

We then impose some constraints on Eq.10 to acquire the optimum solution.

First, we range the above three variables $s \in S, a \in A, e \in E$.

$$s_i, a_{i,j}, e_i \in \{0, 1\} \ \ \forall i, j$$

In our model, a summary can't include the same sentence, arc, or concept twice. Taking Table 2 for example, if s_{13} and s_{33} are included in a summary, the summary has two e_8, but e_8 is 1. This constraint avoids summary redundancy.

The summary must meet the condition of maximum summary size. The following inequality represents the size constraint:

$$\sum_{s_i \in S} l_i s_i \leq K$$

$l_i \in L$ indicates the length of sentence s_i. K is the maximum size of the summary.

The following inequality represents the relationship between sentences and concepts in the sentences.

$$\sum_i m_{ij} s_i \geq e_j \ \ \forall j$$

The above constraint represents Table 2. $m_{i,j}$ is an element of Table 2. If s_i is not included in the summary, the concepts in s_i are not included.

Symbols indicating the beginning and end of the summary must be part of the summary.

[1] Obviously, ILP can be used to search for the path that maximizes the score. While beam search tends to fail to find out the optimal solution, it is tractable and the learning algorithm can estimate the parameter from approximate solutions. For these reasons we use beam search.

$$
\begin{aligned}
s_0 &= 1 \\
s_{n+1} &= 1
\end{aligned}
$$

n is the number of sentences in the input document set.

Next, we describe the constraints placed on arcs.

The beginning symbol must be followed by a sentence or a symbol and must not have any preceding sentences/symbols. The end symbol must be preceded by a sentence or a symbol and must not have any following sentences/symbols. The following equations represent these constraints:

$$
\begin{aligned}
\sum_i a_{0,i} &= 1 \\
\sum_i a_{i,0} &= 0 \\
\sum_i a_{n+1,i} &= 0 \\
\sum_i a_{i,n+1} &= 1
\end{aligned}
$$

Each sentence in the summary must be preceded and followed by a sentence/symbol.

$$
\begin{aligned}
\sum_i a_{i,j} + \sum_i a_{j,i} &= 2s_j \ \ \forall j \\
\sum_i a_{i,j} &= \sum_i a_{j,i} \ \ \forall j
\end{aligned}
$$

The above constraints fail to prevent cycles. To rectify this, we set the following constraints.

$$
\begin{aligned}
\sum_i f_{0,i} &= n \\
\sum_i f_{i,0} &\geq 1 \\
\sum_i f_{i,j} - \sum_i f_{j,i} &= s_j \ \ \forall j \\
f_{i,j} &\leq n a_{i,j} \ \ \forall i,j
\end{aligned}
$$

The above constraints indicate that *flows* f are sent from s_0 as a source to s_{n+1} as a sink. n unit flows are sent from the source and each node expends one unit of flows. More than one flow has to arrive at the sink. By setting these constraints, the nodes consisting of a cycle have no flow. Thus solutions that contain a cycle are prevented. These constraints have also been used to avoid cycles in headline generation (Deshpande et al., 2007).

4 Experiments

This section evaluates our method in terms of ROUGE score and readability. We tested our method and two baselines in two domains: reviews of commodities and restaurants. We collected 4,475 reviews of 100 commodities and 2,940 reviews of 100 restaurants from websites. The commodities included items such as digital cameras, printers, video games, and wines. The average document size was 10,173 bytes in the commodity domain and 5,343 bytes in the restaurant domain. We attempted to generate 300 byte summaries, so the summarization rates were about 3% and 6%, respectively.

We prepared 4 references for each review, thus there were 400 references in each domain. The authors were not those who made up the references. These references were used for ROUGE and readability evaluation.

Since our method requires the parameter vector $\mathbf{w}$ for determining the coherence scores. We trained the parameter vector for each domain. Each parameter vector was trained using 10-fold cross validation. We used 8 samples to train, 1 to develop, and 1 to test. In the restaurant domain, we added 4,390 reviews to each training set to alleviate data sparseness. In the commodity domain, we add 47,570 reviews.[2]

As the solver, we used glpk.[3] According to the development set, λ in Eq.10 was set as 0.1.

4.1 Baselines

We compare our method to the references (which also provide the upper bound) and the opinion summarizers proposed by Carenini et al. (2006) and Lerman et al. (2009) as the baselines.

In the ROUGE evaluations, Human indicates ROUGE scores between references. To compare our summarizer to human summarization, we calculated ROUGE scores between each reference and the other three references, and averaged them.

In the readability evaluations, we randomly selected one reference for each commodity and each restaurant and compared them to the results of the three summarizers.

Carenini et al. (2006)

Carenini et al. (2006) proposed two opinion

[2]The commodities domain suffers from stronger review variation than the restaurant domain so more training data was needed.

[3]`http://www.gnu.org/software/glpk/`

summarizers. One uses a natural language generation module, and other is based on MEAD (Radev et al., 2004). Since it is difficult to mimic the natural language generation module, we implemented the latter one. The objective function Carenini et al. (2006) proposed is as follows:

$$\mathcal{L}_1(S) = \sum_{a \in S} \sum_{s \in D} |\mathrm{polarity}_s(a)| \quad (11)$$

$\mathrm{polarity}_s(a)$ indicates the polarity of aspect a in sentence s present in source document set D. That is, this function gives a high score to a summary that covers aspects frequently mentioned in the input, and whose polarities tend to be either positive or negative.

The solution is identified using the greedy method. If there is more than one sentence that has the same score, the sentence that has the higher centroid score (Radev et al., 2004) is extracted.

Lerman et al. (2009)

Lerman et al. (2009) proposed three objective functions for opinion summarization, and we implemented one of them. The function is as follows:

$$\mathcal{L}_2(S) = -(\mathrm{KL}(p_S(a), p_D(a)) \quad (12)$$
$$+ \sum_{a \in A} \mathrm{KL}(\mathcal{N}(x|\mu_{a_S}, \sigma^2_{a_S}), \mathcal{N}(x|\mu_{a_D}, \sigma^2_{a_D})))$$

$\mathrm{KL}(p, q)$ means the Kullback-Leibler divergence between probability distribution p and q. $p_S(a)$ and $p_D(a)$ are probability distributions indicating how often aspect $a \in A$ occurs in summary S and source document set D respectively. $\mathcal{N}(x|\mu, \sigma^2)$ is a Gaussian distribution indicating distribution of polarity of an aspect whose mean is μ and variance is σ^2. μ_{a_S}, μ_{a_D} and $\sigma^2_{a_S}, \sigma^2_{a_D}$ are the means and the variances of aspect a in summary S and source document set D, respectively. These parameters are determined using maximum-likelihood estimation.

That is, the above objective function gives high score to a summary whose distributions of aspects and polarities mirror those of the source document set.

To identify the optimal solution, Lerman et al. (2009) use a randomized algorithm. First, the summarizer randomly extracts sentences from the source document set, then iteratively performs insert/delete/swap operations on the summary to increase Eq.12 until summary improvement saturates. While this method is prone to lock onto local solutions, the summarizer can reach the optimal solution by changing the starting sentences and repeating the process. In this experiment, we used 100 randomly selected starting points.

Commodity	R-2	R-SU4	R-SU9
(Carenini et al., 2006)	0.158	0.202	0.186
(Lerman et al., 2009)	0.205	0.247	0.227
Our Method	**0.231**	**0.251**	**0.230**
Human	0.384	0.392	0.358

Restaurant	R-2	R-SU4	R-SU9
(Carenini et al., 2006)	0.251	0.281	0.258
(Lerman et al., 2009)	0.260	0.296	**0.273**
Our Method	**0.285**	**0.303**	**0.273**
Human	0.358	0.370	0.335

Table 3: Automatic ROUGE evaluation.

	# of Sentences
(Carenini et al., 2006)	3.79
(Lerman et al., 2009)	6.28
Our Method	7.88
Human	5.83

Table 4: Average number of sentences in the summary.

4.2 ROUGE

We used ROUGE (Lin, 2004) for evaluating the content of summaries. We chose ROUGE-2, ROUGE-SU4 and ROUGE-SU9. We prepared four reference summaries for each document set.

The results of these experiments are shown in Table 3. ROUGE scores increase in the order of (Carenini et al., 2006), (Lerman et al., 2009) and our method, but no method could match the performance of Human. Our method significantly outperformed Lerman et al. (2009)'s method over ROUGE-2 according to the Wilcoxon signed-rank test, while it shows no advantage over ROUGE-SU4 and ROUGE-SU9.

Although our weighting of the set of sentences is relatively naive compared to the weighting proposed by Lerman et al. (2009), our method outperforms their method. There are two reasons for this; one is that we adopt ILP for decoding, so we can acquire preferable solutions efficiently. While the score of Lerman et al. (2009)'s method may be improved by adopting ILP, it is difficult to do so because their objective function is extremely complex. The other reason is the coherence score. Since our coherence score is based on

Commodity	(Carenini et al., 2006)	(Lerman et al., 2009)	Our Method	Human
(Carenini et al., 2006)	-	27/45	18/29	8/46
(Lerman et al., 2009)	18/45	-	29/48	11/47
Our Method	11/29	19/48	-	5/46
Human	38/46	36/47	41/46	-

Restaurant	(Carenini et al., 2006)	(Lerman et al., 2009)	Our Method	Human
(Carenini et al., 2006)	-	31/45	17/31	8/48
(Lerman et al., 2009)	14/45	-	25/47	7/46
Our Method	14/31	22/47	-	8/50
Human	40/48	39/46	42/50	-

Table 5: Readability evaluation.

content words, it may impact the content of the summary.

4.3 Readability

Readability was evaluated by human judges. Since it is difficult to perform absolute evaluation to judge the readability of summaries, we performed a paired comparison test. The judges were shown two summaries of the same input and decided which was more readable. The judges weren't informed which method generated which summary. We randomly chose 50 sets of reviews from each domain, so there were 600 paired summaries.[4] However, as shown in Table 4, the average numbers of sentences in the summary differed widely from the methods and this might affect the readability evaluation. It was not fair to include the pairs that were too different in terms of the number of sentences. Therefore, we removed the pairs that differed by more than five sentences. In the experiment, 523 pairs were used, and 21 judges evaluated about 25 summaries each. We drew on DUC 2007 quality questions[5] for readability assessment.

Table 5 shows the results of the experiment. Each element in the table indicates the number of times the corresponding method won against other method. For example, in the commodity domain, the summaries that Lerman et al. (2009)'s method generated were compared with the summaries that Carenini et al. (2006)'s method generated 45 times, and Lerman et al. (2009)'s method won 18 times. The judges significantly preferred the references in both domains. There were no significant differences between our method and the other two methods. In the restaurant domain, there was a significant difference between (Carenini et al., 2006) and (Lerman et al., 2009).

Since we adopt ILP, our method tends to pack shorter sentences into the summary. However, our coherence score prevents this from degrading summary readability.

[4] ${}_4\mathrm{C}_2 \times 100 = 600$

[5] http://www-nlpir.nist.gov/projects/duc/duc2007/quality-questions.txt

5 Conclusion

This paper proposed a novel algorithm for opinion summarization that takes account of content and coherence, simultaneously. Our method directly searches for the optimum sentence sequence by extracting and ordering sentences present in the input document set. We proposed a novel ILP formulation against selection-and-ordering problems; it is a powerful mixture of the Maximum Coverage Problem and the Traveling Salesman Problem. Experiments revealed that the algorithm creates summaries that have higher ROUGE scores than existing opinion summarizers. We also performed readability experiments. While our summarizer tends to extract shorter sentences to optimize summary content, our proposed coherence score prevented this from degrading the readability of the summary.

One future work includes enriching the features used to determine the coherence score. We expect that features such as entity grid (Barzilay and Lapata, 2005) will improve overall algorithm performance. We also plan to apply our model to tasks other than opinion summarization.

Acknowledgments

We would like to sincerely thank Tsutomu Hirao for his comments and discussions. We would also like to thank the anonymous reviewers for their comments.

References

Althaus, Ernst, Nikiforos Karamanis and Alexander Koller. 2004. Computing Locally Coherent Discourses. In *Proc. of the 42nd Annual Meeting of the Association for Computational Linguistics*.

Balas, Egon. 1989. The prize collecting traveling salesman problem. *Networks*, 19(6):621–636.

Banko, Michele, Vibhu O. Mittal and Michael J. Witbrock. 2000. Headline Generation Based on Statistical Translation. In *Proc. of the 38th Annual Meeting of the Association for Computational Linguistics*.

Barzilay, Regina, Noemie Elhadad and Kathleen McKeown. 2002. Inferring Strategies for Sentence Ordering in Multidocument Summarization. *Journal of Artificial Intelligence Research*, 17:35–55.

Barzilay, Regina and Mirella Lapata. 2005. Modeling Local Coherence: An Entity-based Approach. In *Proc. of the 43rd Annual Meeting of the Association for Computational Linguistics*.

Carbonell, Jaime and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proc. of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Carenini, Giuseppe, Raymond Ng and Adam Pauls. 2006. Multi-Document Summarization of Evaluative Text. In *Proc. of the 11th Conference of the European Chapter of the Association for Computational Linguistics*.

Clarke, James and Mirella Lapata. 2007. Modelling Compression with Discourse Constraints. In *Proc. of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*.

Collins, Michael. 2002. Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In *Proc. of the 2002 Conference on Empirical Methods in Natural Language Processing*.

Crammer, Koby, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz and Yoram Singer. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585.

Deshpande, Pawan, Regina Barzilay and David R. Karger. 2007. Randomized Decoding for Selection-and-Ordering Problems. In *Proc. of Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*.

Elsner, Micha, Joseph Austerweil and Eugene Charniak. 2007. A unified local and global model for discourse coherence. In *Proc. of Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*.

Filatova, Elena and Vasileios Hatzivassiloglou. 2004. A Formal Model for Information Selection in Multi-Sentence Text Extraction. In *Proc. of the 20th International Conference on Computational Linguistics*.

Gillick, Dan and Benoit Favre. 2009. A Scalable Global Model for Summarization. In *Proc. of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics Workshop on Integer Linear Programming for NLP*.

Hirao, Tsutomu, Hideki Isozaki, Eisaku Maeda and Yuji Matsumoto. 2002. Extracting important sentences with support vector machines. In *Proc. of the 19th International Conference on Computational Linguistics*.

Kupiec, Julian, Jan Pedersen and Francine Chen. 1995. A Trainable Document Summarizer. In *Proc. of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*.

Lapata, Mirella. 2003. Probabilistic Text Structuring: Experiments with Sentence Ordering. In *Proc. of the 41st Annual Meeting of the Association for Computational Linguistics*.

Lapata, Mirella. 2006. Automatic Evaluation of Information Ordering: Kendall's Tau. *Computational Linguistics*, 32(4):471–484.

Lerman, Kevin, Sasha Blair-Goldensohn and Ryan McDonald. 2009. Sentiment Summarization: Evaluating and Learning User Preferences. In *Proc. of the 12th Conference of the European Chapter of the Association for Computational Linguistics*.

Lin, Chin-Yew. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Proc. of Text Summarization Branches Out*.

Martins, Andre F. T., and Noah A. Smith. 2009. Summarization with a Joint Model for Sentence Extraction and Compression. In *Proc. of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics Workshop on Integer Linear Programming for NLP*.

McDonald, Ryan. 2007. A Study of Global Inference Algorithms in Multi-document Summarization. In *Proc. of the 29th European Conference on Information Retrieval*.

Nishikawa, Hitoshi, Takaaki Hasegawa, Yoshihiro Matsuo and Genichiro Kikui. 2010. Optimizing Informativeness and Readability for Sentiment Summarization. In *Proc. of the 48th Annual Meeting of the Association for Computational Linguistics*.

Radev, Dragomir R., Hongyan Jing, Magorzata Sty and Daniel Tam. 2004. Centroid-based summarization of multiple documents. *Information Processing and Management*, 40(6):919–938.

Soricut, Radu and Daniel Marcu. 2006. Discourse Generation Using Utility-Trained Coherence Models. In *Proc. of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*.

Takamura, Hiroya and Manabu Okumura. 2009. Text Summarization Model based on Maximum Coverage Problem and its Variant. In *Proc. of the 12th Conference of the European Chapter of the Association for Computational Linguistics*.

Yih, Wen-tau, Joshua Goodman, Lucy Vanderwende and Hisami Suzuki. 2007. Multi-Document Summarization by Maximizing Informative Content-Words. In *Proc. of the 20th International Joint Conference on Artificial Intelligence*.