Coherent Arrangement of Sentences Extracted from Multiple Newspaper Articles

Naoaki OKAZAKI † Yutaka MATSUO ‡ Mitsuru ISHIZUKA †
†Graduate School of Information Science and Technology
The University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
{okazaki, ishizuka}@miv.t.u-tokyo.ac.jp

‡Cyber Assist Research Center
AIST Tokyo Waterfront
2-41-6 Aomi, Koto-ku, Tokyo 135-0064, Japan
y.matsuo@carc.aist.go.jp

Abstract

Multi-document summarization is a challenge to information overload problem to provide a condensed text for a number of documents. Most multi-document summarization systems make use of extraction techniques (e.g., important sentence extraction) and compile a summary from the selected information. However, sentences gathered from multiple sources are not organized as a comprehensible text. Therefore, it is important to consider sentence ordering of extracted sentences in order to reconstruct discourse structure in a summary. We propose a novel method to plan a coherent arrangement of sentences extracted from multiple newspaper articles. Results of our experiment show that sentence reordering has a discernible effect on summary readability. The results also shows significant improvement on sentence arrangement compared to former methods.

1 Introduction

There is a great deal of computerized documents accessible on-line. With the help of search engines, we can obtain a set of relevant documents that fits to our interest. Even though we narrow the range of documents to be read through the search phase, we often get disgusted with the quantity of retrieved documents. Automatic text summarization is a challenge to the information overload problem to provide a condensed text for a given document. Multi-document summarization (MDS), which is an extension of summarization to related documents (e.g., a collection of documents or web pages retrieved from a search engine, collected papers on a certain research field, etc.), has attracted much attention in recent years. We obtain precise information more quickly and easily with the help of a summary or use the summary in place of the original text.

Figure 1 illustrates an example of typical MDS system. Given a number of documents, a MDS system yields a summary by gathering information from original documents. Important sentence or paragraph extraction, which finds significant textual segments to be included into a summary, plays a major role in most summarization system. There has been a great deal of research to improve sentence/paragraph extraction because the quality of extraction has much effect on overall performance in a MDS system.

However, post-processing of extraction is also important to secure summary readability. We should eliminate unnecessary parts within extracted sentences to gain a higher compression ratio or insert necessary expressions to complement missing information. We should also break a long sentence into several sentences or combine several sentences into one sentence. Although there are numerous directions to improve summary readability as a post-processing phase of extraction, we consider a method to arrange extracted sentences coherently and inquire the necessity of a sequential ordering of summary sentences.

In this paper we propose an approach for coherent arrangement of sentences extracted from multiple newspaper articles. The rest of this paper is organized as follows. We present an outline of sentence ordering problem and related research including chronological sentence ordering, which is widely used in conventional MDS systems. We point an issue of chronological ordering and explain an approach to improve chronological ordering by complementing on pre-

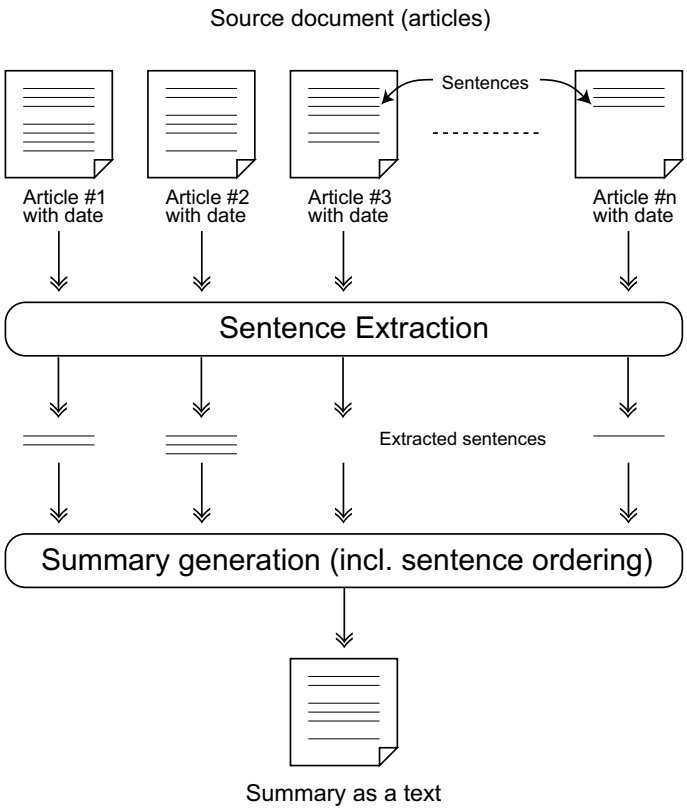

Figure 1. A simplified summarization system with sentence extraction.

supposed information of each sentence. The subsequent section (Section 3) addresses evaluation metrics to validate the effectiveness of the algorithm in MDS and show experimental results. In Section 4 we discuss future work and conclusion of this paper.

2 Sentence Ordering

Our goal is to determine a most probable permutation of sentences or, in other words, reconstruct discourse structure of sentences gathered from multiple sources. When a human is asked to make an arrangement of sentences, he or she may perform this task without difficulty just as we write out thoughts in a text. However, we must consider what accomplishes this task since computers are unaware of order of things by nature. Discourse coherence, typified by rhetorical relation [10] and coherence relation [5], is of help to this question. Hume [6] claimed qualities from which association arises and by which the mind is conveyed from one idea to another are three: *resemblance*; *contiguity in time or place*; and *cause and effect*. That is to say we should organize a text from fragmented information on the basis of topical relevancy, chronological sequence, and cause-effect relation. It is especially true in sentence ordering of newspaper articles because we must arrange a large number of time-series events concerning several topics.

Barzilay et. al. [1] address the problem of sentence ordering in the context of multi-document summarization and the impact of sentence ordering on readability of a summary. They proposed two naive sentence-ordering techniques such as majority ordering (examines most frequent orders in the original documents) and chronological ordering (orders sentence by the publication date). Showing that using naive ordering algorithms does not produce satisfactory orderings, Barzilay et. al. also investigate through experiments with humans in order to identify patterns of orderings that can improve the algorithm. Based on the experiments, they propose another algorithm that utilizes topical segment and chronological ordering. Lapata [7] proposed another approach to information ordering based on a probabilistic model that assumes the probability of any given sentence is determined by its adjacent sentence and learns constraints on sentence order from a corpus of domain specific texts. Lapata estimates transitional probability between sentence by some attributes such as verbs (precedence relationships of verbs in the corpus), nouns (entity-based coherence

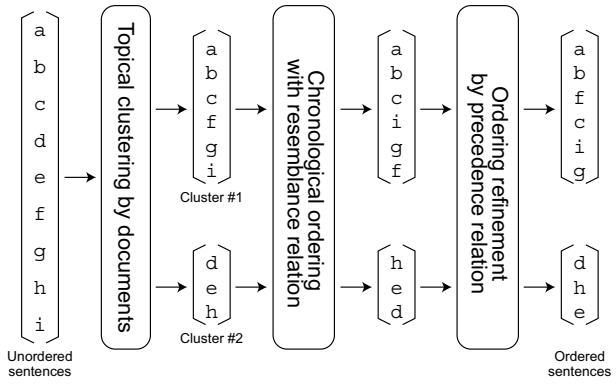

Figure 3. The outline of ordering algorithm.

by keeping track of the nouns) and dependencies (structure of sentences). The paper describes the approach was successful by comparing the proposed orderings with human-made orderings.

Against the background of these studies, we propose the use of antecedent sentences to arrange sentences coherently. Let us consider an example shown in Figure 2. There are three sentence `a`, `b`, and `c` from which we get an order `[a-b-c]` by chronological ordering. When we read these sentences in this order, we find sentence `b` to be incorrectly positioned. This is because sentence `b` is written on the presupposition that the reader may know Dolly had a child. In other words, it is more fitting to assume sentence `b` to be an elaboration of sentence `c`. As you may easily be able to imagine, there are some precedent sentences prior to sentence `b` in the original document. Lack of presupposition obscures what a sentence is saying and confuses the readers. Hence, we should refine the chronological order and revise the order to `[a-c-b]`, putting sentence `c` before sentence `b`. This example is nothing extreme because an extraction method for multi-document summarization (e.g., [2]) chooses sentences from all over the position in source documents to secure information coverage and refuse redundant information.

We show a block diagram of our ordering algorithm shown in Figure 3. Given nine sentences denoted by `[a b ... i]`, for example, the algorithm eventually produces an ordering, `[a-b-f-c-i-g-d-h-e]`. We consider topical segmentation and chronological ordering to be fundamental to sentence ordering as well as conventional ordering techniques [1] and make an attempt to refine the ordering. We firstly recognize topics in source documents to separate sentences referring to a topic from ones referring to another. In Figure 3 example we obtain two topical segments (clusters) as an output from the topical clustering. In the second phase we order sentences of each segment by the chronological order. If two sentences have the same chronological order, we elaborate the order on the basis of sentence position and resemblance relation. Finally, we refine each ordering by resolving antecedent sentences and output the final ordering. In the rest of this section we give a detailed description of each phase.

2.1 Topical segmentation

The first task is to categorize sentences by their topics. We assume a newspaper article to be written about one topic. Hence, to classify topics in sentences, we have only to classify articles by their topics. Given l articles and we found m kinds of terms in the articles. Let D be a document-term matrix ($l \times m$), whose element D_{ij} represents frequency of a term #j in document #i, We use D_i to denote a term vector (i-component row vector) of document #i. After measuring distance or dissimilarity between two articles #x and #y:

$$\mathrm{distance}(D_x, D_y) = 1 - \frac{D_x \cdot D_y}{|D_x||D_y|}, \qquad (1)$$

we apply the nearest neighbor method [3] to merge a pair of clusters when their minimum distance is lower than a given parameter $\alpha = 0.3$ (determined empirically). At last we classify sentences according to topical clusters, assuming that a sentence in a document belonging to a cluster also belongs to the same cluster.

2.2 Chronological ordering

It is difficult for computers to find a resemblance or cause-effect relation between two phenomena: there is a great deal of possible relations classified in detail; and we do not have conclusive evidence whether a pair of sentences that we arbitrarily gather from multiple documents has some relation. A newspaper usually deals with novel events that have occurred since the last publication. Hence, publication date (time) of each article turns out to be a good estimator of resemblance relation (i.e., we observe a trend or series of relevant events in a time period), contiguity in time, and cause-effect relation (i.e., an event occurs as a result of previous events). Although resolving temporal expressions in sentences (e.g., *yesterday, the next year, etc.*) [8, 9] may give a more precise estimation of these relations, it is not an easy task. For this reason we first order sentences by the chronological order, assigning a time stamp for each sentence by its publication date (i.e., the date when the article was written).

When there are sentences having the same time stamp, we elaborate the order on the basis of sentence position and sentence connectivity. We restore an original ordering if two sentences have the same time stamp and belong to the same article. If sentences have the same time stamp and are

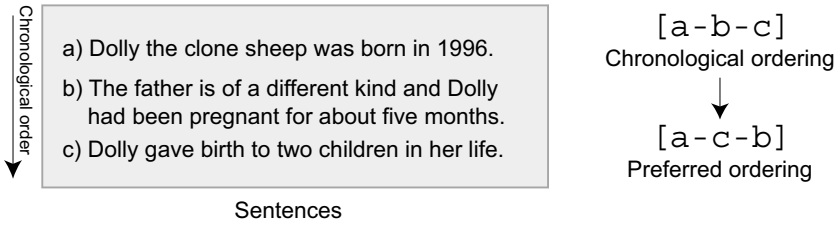

Figure 2. A problem case of chronological sentence ordering.

not from the same article, we put a sentence which is more similar to previously ordered sentences to assure sentence connectivity.

2.3 Improving chronological ordering

After we obtain a chronological order of sentences, we make an effort to improve the ordering with the help of antecedent sentences. Figure 4 shows the background idea of ordering refinement by precedence relation. Just as the example in Figure 2, we have three sentences a, b, and c in chronological order. At first we get sentence a out of the sentences and check its antecedent sentences. Seeing that there are no sentences prior to sentence a in article #1, we take it acceptable to put sentence a here. Then we get sentence b out of remaining sentences and check its antecedent sentences. We find several sentences before sentence b in article #2 this time. Grasping what the antecedent sentences are saying, we confirm first of all whether if their saying is mentioned by previously arranged sentences (i.e., sentence a). If it is mentioned, we put sentence b here and extend the ordering to [a-b]. Otherwise, we search a substitution for what the precedence sentences are saying from the remaining sentences (i.e., sentence c in this example). In Figure 4 example, we find out sentence a is not referring to what sentence c′ is saying but sentence c is approximately referring to that. Putting sentence c before b, we finally get the refined ordering [a-c-b].

Figure 5 illustrates how our algorithm refines a given chronological ordering [a-b-c-d-e-f]. In Figure 5 example we leave position of sentences a and b because they do not have precedent sentences in their original article (i.e., they are lead sentences[1]). On the other hand, sentence c has some preceding sentences in its original document. This presents two choices to us: we should check if it is safe to put sentence c just after sentences a and b; or we should arrange some sentences before sentence c as a substitute of the precedent sentences. Preparing a term vector of the precedent sentences, we search a sentence or a set of sentences which is the most similar to the precedent content in

[1] *lead sentences* are sentences which appear at the beginning in an article

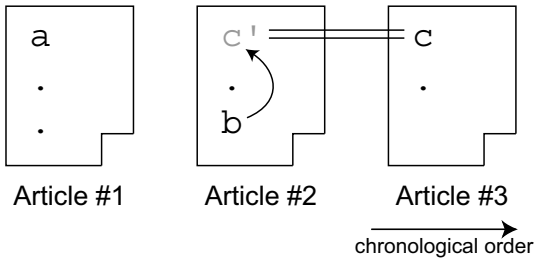

Figure 4. Background idea of ordering refinement by precedence relation.

sentences {a,b}, d, e, and f. In other words, we assume sentence ordering to be [a-b-X-c] and find appropriate sentence(s) X if any. Supposing that sentence e in Figure 5 describes similar content as the precedent sentences for sentence c, we substitute X with Y-e. We check whether we should put some sentences before sentence e or not. Given that sentence e is a lead sentence, we leave Y as empty and fix the resultant ordering to [a-b-e-c].

Then we consider sentence d, which is not a lead sentence again. Preparing a term vector of the precedent sentences of sentence d, we search a sentence or a set of sentences which is the most similar to the precedent content in sentences {a,b,e,c}, f. Supposing that either sentence a, b, e or c refers to the precedent content closer than sentence f, we make a decision to put sentence d here. In this way we get the final ordering, [a-b-e-c-d-f].

2.4 Compatibility with multi-document summarization

We describe briefly how our ordering algorithm goes together with MDS. Let us think the example shown in Figure 4 again. In this example, sentence extraction does not choose sentence c′ while sentence c is very similar to sentence c′. You may think this is rare case for explanation, but it could happen as we optimize a sentence-extraction method for MDS. A method for MDS (e.g., [2]) makes ef-

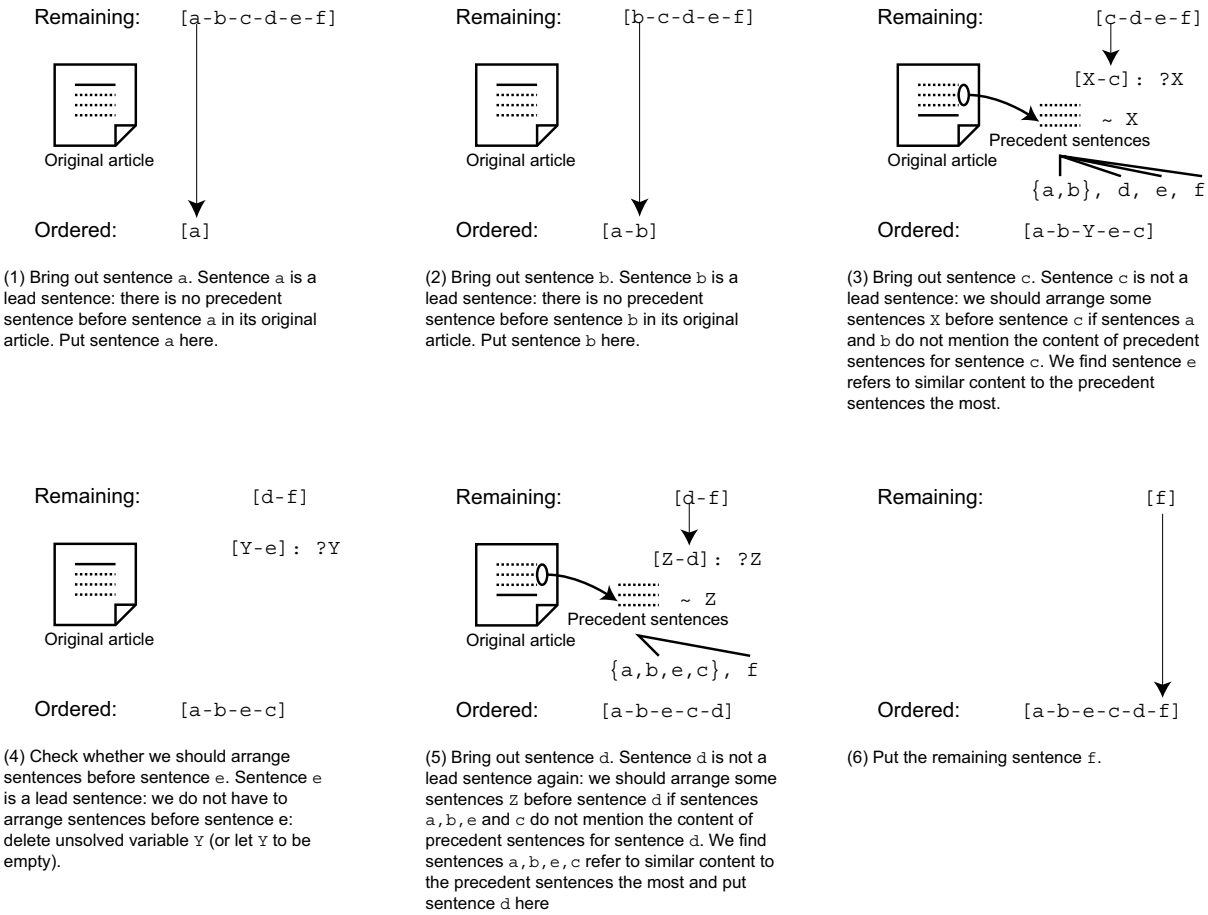

Figure 5. Improving chronological ordering with the help of antecedent sentences.

fort to acquire information coverage under a condition that there is a number of sentences as summary candidates. This is to say that an extraction method should be able to refuse redundant information.

When we collect articles which describe a series of an event, we may find that lead sentences convey similar information over the articles since the major task of lead sentences is to give a subject. Therefore, it is quite natural that: lead sentences c and c′ refer to similar content; an extraction method for MDS does not choose both sentence c′ and c in terms of redundancy; and the method also prefers either sentence c or c′ in terms of information coverage.

3 Evaluation

3.1 Experiment and evaluation metrics

We conducted an experiment of sentence ordering through multi-document summarization to test the effectiveness of the proposed method. We utilized the TSC-3 [4] test collection, which consists of 30 sets of multi-document summarization task. Performing an important sentence extraction for MDS [11] up to the specified number of sentences (approximately 10% of summarization rate), we made a material for a summary (i.e., extracted sentences) for each task. We order the sentences by six methods: *human-made ordering (HO)* as the highest anchor; *random ordering (RO)* as the lowest anchor; *chronological ordering (CO)* as a conventional method; *chronological ordering with topical segmentation (COT)* (similar to Barzilay's method [1]); *proposed method without topical segmentation (PO)*; and *proposed method with topical segmentation (POT))*. We asked three human judges to evaluate sentence ordering of 28 summaries out of TSC-3 test collection [2] .

The first evaluation task is a subjective grading where a human judge marks an ordering of summary sentences on

[2]We exclude two summaries because they are so long (approximately 30 sentences) that it is hard for judges to evaluate and revise them.

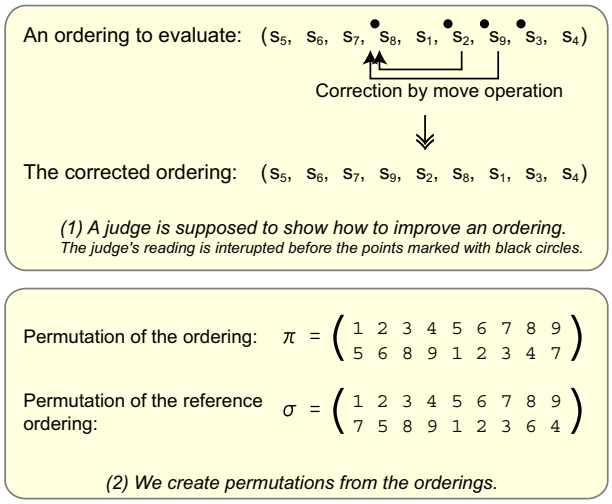

Figure 6. Correction of an ordering.

a scale of 4: 4 (*perfect*: we cannot improve any further), 3 (*acceptable*: makes sense even though there is some room for improvement), 2 (*poor*: requires minor amendment to bring it up to the acceptable level), and 1 (*unacceptable*: requires overall restructuring rather than partial revision).

In addition to the rating, it is useful that we examine how close an ordering is to an acceptable one when the ordering is regarded as *poor*. Considering several sentence-ordering patterns to be acceptable for a given summary, we think it is valuable to measure the degree of correction because this metric virtually requires a human corrector to prepare a correct answer for each ordering in his or her mind. Therefore, a human judge is supposed to illustrate how to improve an ordering of a summary when he or she marks the summary with *poor* in the rating task. We restrict applicable operations of correction to move operation to keep minimum correction of the ordering. We define a move operation here as removing a sentence and inserting the sentence into an appropriate place (see Figure 6-(1)).

Supposing a sentence ordering to be a rank, we can calculate rank correlation coefficient of a permutation of an ordering π and a permutation of the reference ordering σ. Let $\{s_1, ..., s_n\}$ be a set of summary sentences identified with index numbers from 1 to n. We define a permutation $\pi \in S_n$ to denote an ordering of sentences where $\pi(i)$ represents an order of sentence s_i. Similarly, we define a permutation $\sigma \in S_n$ to denote the corrected ordering. For example, the π and σ in Figure 6 will be:

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 5 & 6 & 8 & 9 & 1 & 2 & 3 & 4 & 7 \end{pmatrix}, \quad (2)$$

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 7 & 5 & 8 & 9 & 1 & 2 & 3 & 6 & 4 \end{pmatrix}. \quad (3)$$

Spearman's rank correlation $\tau_s(\pi, \sigma)$ and Kendall's rank correlation $\tau_k(\pi, \sigma)$ are known as famous rank correlation metrics and were used in Lapata's evaluation [7].

$$\tau_s(\pi, \sigma) = 1 - \frac{6}{n(n+1)(n-1)} \sum_{i=1}^{n} (\pi(i) - \sigma(i))^2, \quad (4)$$

$$\tau_k(\pi, \sigma) = \frac{1}{n(n-1)/2} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \mathrm{sgn}(\pi(j) - \pi(i)) \cdot \mathrm{sgn}(\sigma(j) - \sigma(i)), \quad (5)$$

where $\mathrm{sgn}(x) = 1$ for $x > 0$ and -1 otherwise. These metrics range from -1 (an inverse rank) to 1 (an identical rank) via 0 (a non-correlated rank). In the example shown in Figure 6-(2) we obtain $\tau_s(\pi, \sigma) = 0.85$ and $\tau_k(\pi, \sigma) = 0.72$.

We propose another metric to assess the degree of sentence continuity in reading, $\tau_c(\pi, \sigma)$:

$$\tau_c(\pi, \sigma) = \frac{1}{n} \sum_{i=1}^{n} \mathrm{equals}\left(\pi\sigma^{-1}(i), \pi\sigma^{-1}(i-1) + 1\right), \quad (6)$$

where: $\pi(0) = \sigma(0) = 0$; $\mathrm{equals}(x, y) = 1$ when x equals y and 0 otherwise. This metric ranges from 0 (no continuity) to 1 (identical). The summary in Figure 6-(1) may interrupt judge's reading after sentence S_7, S_1, S_2 and S_9 as he or she searches a next sentence to read. Hence, we observe four discontinuities in the ordering and calculate sentence continuity $\tau_c(\pi, \sigma) = (9 - 4)/9 = 0.56$.

3.2 Result

Figure 7 shows distribution of rating score of each method in percentage of 84 (28×3) summaries. Judges marked about 75% of human-made ordering (HO) as either perfect or acceptable while they rejected as many as 95% of random ordering (RO). Chronological ordering (CO) did not yield satisfactory result losing a thread of 63% summaries although CO performed much better than RO. Topical segmentation could not contribute to ordering improvement of CO as well: COT is slightly worse than CO After taking an in-depth look at the failure orderings, we found the topical clustering did not perform well during this test. We suppose that the topical clustering could not prove the merits with this test collection because the collection consists of relevant articles retrieved by some query and polished well by a human and thus exclude unrelated articles to a topic. On the other hand, the proposed method (PO) improved chronological ordering much better than topical segmentation: sum of perfect and acceptable ratio jumped

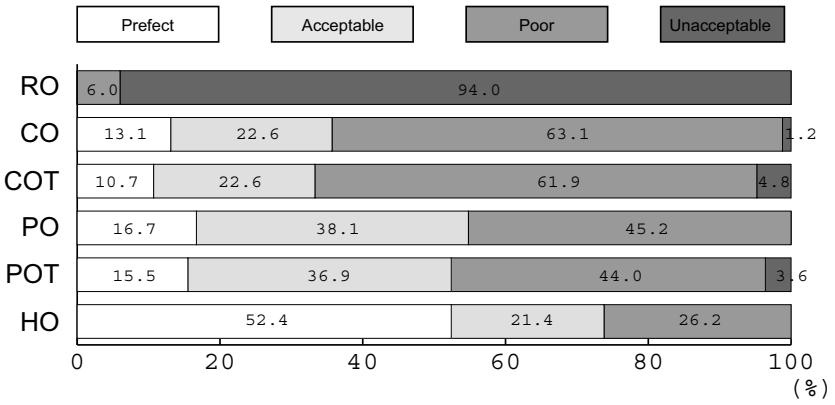

Figure 7. Distribution of rating score of orderings in percentage.

	Spearman		Kendall		Continuity	
Method	AVG	SD	AVG	SD	AVG	SD
RO	0.041	0.170	0.035	0.152	0.018	0.091
CO	0.838	0.185	0.870	0.270	0.775	0.210
COT	0.847	0.164	0.791	0.440	0.741	0.252
PO	0.843	0.180	0.921	0.144	0.856	0.180
POT	0.851	0.158	0.842	0.387	0.820	0.240
HO	0.949	0.157	0.947	0.138	0.922	0.138

Table 1. Comparison with corrected ordering.

	Spearman		Kendall		Continuity	
Method	AVG	SD	AVG	SD	AVG	SD
RO	-0.117	0.265	-0.073	0.202	0.054	0.064
CO	0.838	0.185	0.778	0.198	0.578	0.218
COT	0.847	0.164	0.782	0.186	0.571	0.229
PO	0.843	0.180	0.792	0.184	0.606	0.225
POT	0.851	0.158	0.797	0.171	0.599	0.237
HO	1.000	0.000	1.000	0.000	1.000	0.000

Table 2. Comparison with human-made ordering.

up from 36% (CO) to 55% (PO). This shows ordering refinement by precedence relation improves chronological ordering by pushing poor ordering to an acceptable level.

Table 1 reports closeness of orderings to the corrected ones with average scores (AVG) and the standard deviations (SD) of the three metrics τ_s, τ_k and τ_c. It appears that average figures shows similar tendency to the rating task with three measures: HO is the best; PO is better than CO; and RO is definitely the worst. We applied one-way analysis of variance (ANOVA) to test the effect of four different methods (RO, CO, PO and HO). ANOVA proved the effect of the different methods ($p < 0.01$) for three metrics. We also applied Tukey test to compare the difference between these methods. Tukey test revealed that RO was definitely the worst with all metrics. However, Spearman's rank correlation τ_S and Kendall's rank correlation τ_k failed to prove the significant difference between CO, PO and HO. Only sentence continuity τ_c proved PO is better than CO; and HO is better than CO ($\alpha = 0.05$). The Tukey test proved that sentence continuity has better conformity to the rating results and higher discrimination to make a comparison.

Table 2 shows closeness of orderings to ones made by human. Although we found RO is clearly the worst as well as other results, we cannot find the significant difference between CO, PO, and HO. This result revealed the difficulty of automatic evaluation by preparing a correct ordering.

4 Conclusion

In this paper we described our approach to coherent sentence arrangement for multiple newspaper articles. The results of our experiment revealed that our algorithm for sentence ordering did contribute to summary readability in MDS and improve chronological sentence ordering significantly. We plan to do further study on the sentence ordering problem in future work, explore how to apply our algorithm to documents other than newspaper and integrate ordering problem with extraction problem to benefit each other and overall quality of MDS.

Acknowledgment

We made use of Mainichi Newspaper and Yomiuri Newspaper articles and summarization test collection of TSC-3. We wish to thank reviewers for valuable comments on our paper.

References

[1] R. Barzilay, E. Elhadad, and K. McKeown. Inferring strategies for sentence ordering in multidocument summarization. *Journal of Artifical Intelligence Research (JAIR)*, 17:35–55, 2002.

[2] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 335–336, 1998.

[3] T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, IT-13:21–27, 1967.

[4] T. Hirao, M. Okumura, T. Fukusima, and H. Nanba. Text summarization challenge 3: text summarization evaluation at ntcir workshop4. In *Working note of the 4th NTCIR Workshop Meeting*, pages 407–411, 2004.

[5] J. Hobbs. *Literature and Cognition, CSLI Lecture Notes 21*. CSLI, 1990.

[6] D. Hume. *Philosophical Essays concerning Human Understanding*. 1748.

[7] M. Lapata. Probabilistic text structuring: experiments with sentence ordering. In *Proceedings of the 41st Meeting of the Association of Computational Linguistics*, pages 545–552, 2003.

[8] I. Mani, B. Schiffman, and J. Zhang. Inferring temporal ordering of events in news. *Proceedings of the Human Language Technology Conference (HLT-NAACL) '03*, 2003.

[9] I. Mani and G. Wilson. Robust temporal processing of news. In *Proceedings of the 38th Annual Meeting of ACL'2000*, pages 69–76, 2000.

[10] W. Mann and S. Thompson. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8:243–281, 1988.

[11] N. Okazaki, Y. Matsuo, and M. Ishizuka. TISS: An integrated summarization system for TSC-3. In *Working note of the 4th NTCIR Workshop Meeting*, pages 436–443, 2004.