TISS: An Integrated Summarization System for TSC-3

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

In consideration of the previous workshop, we participate in TSC-3 to make improvements on important sentence extraction used in dry run of TSC-2. We formulate important sentence extraction as a combinational optimization problem that determines a set of sentences containing as many important information fragments as possible. In addition to the extraction method, we reinforce peripheral components such as sentence ordering, anaphora analysis and sentence compression to improve summary readability. We propose a remedy of chronological ordering by complementing presupposed information of each sentence. This paper reports mainly on important sentence extraction and sentence ordering.
Keywords: *multi-document summarization, sentence extraction, sentence ordering, TSC*

1 Introduction

In the previous workshop (TSC-2) we proposed two different summarization methods in dry run and formal run [7]. The method in the dry run utilized sentence extraction for multi-document summarization which aimed at minimum inclusion of duplicate information as well as maximum coverage of original content. The evaluation result showed that we must review the representation of sentences (i.e., representation by a set of term-cooccurrence relations). The method in the formal run employed spreading activation through sentence similarity to rank sentences with an assumption that sentences which are relevant to many ones of significance are also significant. The method showed impressive results for short summaries, but not so good results for long summaries.

In the consideration of the previous workshop, we participate in TSC-3 to make improvements on the former method on the ground that we should consider information redundancy in extraction stage. In addition to the extraction method, we reinforce peripheral components such as *sentence ordering, anaphora analysis* and *sentence compression* to refine summary readability. Figure 1 shows architecture of our summarization system in TSC-3. In the first step all documents are passed to CaboCha [4] to acquire dependency structure of sentences with named entities, which is supposed to be sent to the rest of summarization components. We perform two kinds of tasks on the summarization source: *important sentence extraction* and *analyses for generating a summary of good readability*. We finally compile a summary based on the outputs from the various components in the last phase.

This paper is organized as follows. The following section describes sentence extraction as information fragment covering and sentence representation by a set of information fragments. This section shows evaluation results as well. Then we address the issue of improving chronological sentence ordering in section 3. We refine chronological sentence ordering by resolving antecedence sentences. We also show our experiment and evaluation results of the argumented algorithm. Section 4 outlines other components in our summarization system including anaphora analysis and sentence compression. After we address the TSC-3 evaluation results in terms of readability, we conclude this paper.

2 Sentence Extraction

Humans can interpret the meaning of a text and find important places in the text based on their interpretation. Understanding what each sentence is saying, a

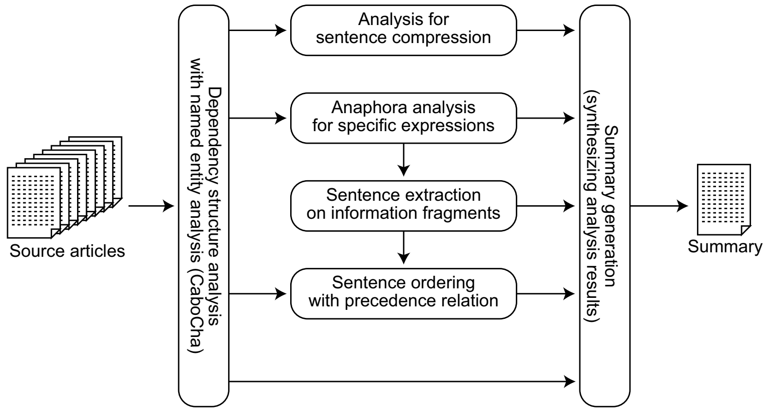

Figure 1. Architecture of the summarization system.

human discerns which information is important for inclusion in a summary. Although it remains unknown how to represent the meaning of a sentence, we assume that a human reader can break a sentence into several informational phrases to which the sentence is referring and mark several sentences that mention the important information. In other words, we assume that a sentence can be represented by a set of *information fragments* that convey fragmentary information in a sentence. This assumption allows us to see important sentence extraction as a problem of considering what kinds of information fragments are important and which sentences convey important information fragments. We first discuss sentence representation as a set of information fragments. Then we address sentence extraction based on the information fragments.

2.1 Information fragment representation

Several kinds of internal representations deal with sentence content. They can be applied to summarization. For example, the vector space model represents a sentence with a set of terms that are contained in the sentence. A number of extraction methods calculate the sentence score as the sum of term weights. Some MDS methods (e.g., [2, 8]) calculate the sentence similarity of term vectors to identify topical segments and thereby prevent the inclusion of similar or redundant information. Nagao et al. [6] proposed the use of Global Document Annotation (GDA).

Against the background of these studies, we propose the use of the dependency structure of terms in a sentence. Figure 2 demonstrates a procedure for converting a sentence into an internal representation composed of *information fragments*. We first obtain the dependency structure of a sentence using CaboCha. Note that the English version of a dependency tree (right side of the figure) does not reflect the English source sentence because it is a word-by-word translation of a Japanese dependency tree. Deleting function words and stop words, we extract pairs of terms that have a modification relation. We obtain six pairs of terms in the Fig. 2 example. These information fragments can be transcribed into comprehensible sentences respectively: “neutrino is an elementary particle”; “neutrino was verified”; “mass was verified”; “ICRR is a part of Japan-US Cooperative Research Group”; “(neutrino was) verified last week”. Because the information fragment representation of a sentence partially refers to what the original sentence is saying (with a certain degree of human interpretation), this representation is useful to keep track of information conveyed by the extracted sentences.

Moreover, adding a weight (importance) to each information fragment gives an indicator of which sentence contains important information and eventually which sentence we should choose for a summary. For an information fragment that consists of terms x and y, we define a weighting function:

$$\mathrm{w}(x, y) = \frac{\mathrm{hl}(x)\mathrm{tf}(x) + \mathrm{hl}(y)\mathrm{tf}(y)}{2}. \qquad (1)$$

Therein, $\mathrm{hl}(x) = 2$ if term x is a headline term, otherwise $\mathrm{hl}(x) = 1$; and $\mathrm{tf}(x)$ denotes occurrence frequency of term x in source documents.

2.2 Sentence extraction as information fragment covering

Important sentence extraction can be formulated as a combinational optimization problem that determines a set of sentences containing as many important information fragments as possible. Let D be source documents with n sentences $\{s_1, ..., s_n\}$. We define a function $\mathrm{s\_length}(i)$ to represent the number of characters in sentence s_i. Let us suppose we found m information fragments $\{c_1, ..., c_m\}$ in total after we analyze all sentences in documents D. We introduce a matrix

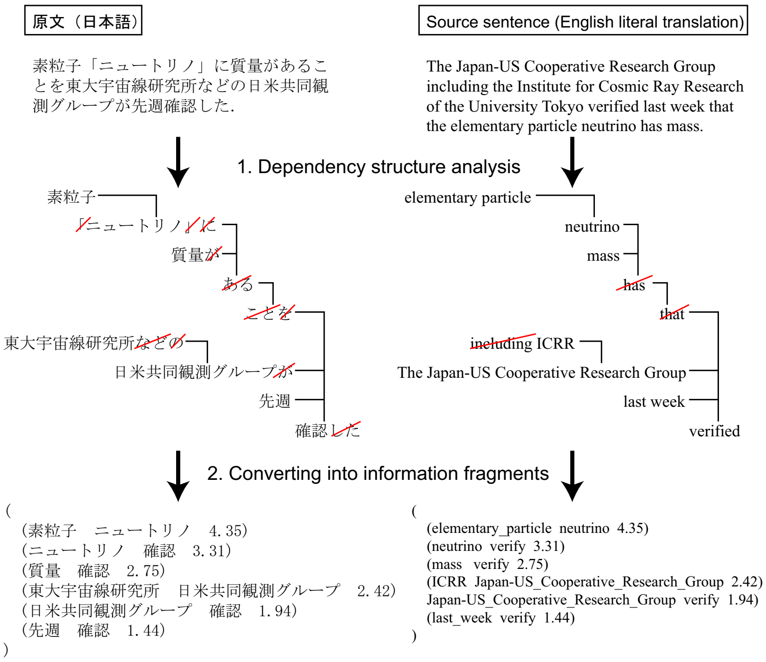

Figure 2. Generation of information fragments from a sentence.

$W(n \times m)$ whose element w_{ij} represents:

$$w_{ij} = \begin{cases} \text{weight of } c_j \text{ in } s_i & (c_j \in s_i) \\ 0 & (c_j \notin s_i) \end{cases} . \tag{2}$$

We call this matrix W *the sentence information-fragment matrix* (Figure 3).

Next we consider a method to extract important sentences that are no longer than L characters from sentence information-fragment matrix W. Introducing a function $\mathrm{s\_weight}(i)$ to represent the importance of sentence s_i, we formulate the extraction problem as an optimal path-search problem that maximizes F in the following formula to obtain a permutation [1] of index numbers of important sentences E [2] :

$$F = \operatorname*{argmax}_{E \in D^{(l)}, \forall l: 0 < l \leq n} \sum_{k=1}^{l} \mathrm{s\_weight}(E_k), \tag{3}$$

$$\text{where} \sum_{k=1}^{l} \mathrm{s\_length}(E_k) \leq L. \tag{4}$$

Therein, l is a variable that denotes the number of extracted sentences; $D^{(l)}$ is a set of all possible permutations composed of l sentences; and E_k represents the index number of sentence at k-th order in permutation E. This optimization problem finds a permutation of sentences E with maximal summation of importance within a specified summarization ratio.

[1]Note that F is dependent on the calculation order.

[2]If we choose sentences s_1, s_3, s_6, s_7, E will be $(1, 3, 6, 7)$.

	elementary particle neutrino	mystery neutrino	mass existence	mystery unravel	...	conclusion publish	
Sentence 1	0.871	0.387	0.187	0.088		0.000	
Sentence 2	0.277	0.000	0.000	0.054		0.322	
Sentence 3	1.215	0.000	0.473	0.000		0.000	
......							n rows
Sentence n	0.000	0.000	0.000	0.000		0.000	
			m columns				

Figure 3. A sentence information-fragment matrix.

We define a sentence-weighting function with a feature to lower weights that have already been mentioned in summary sentences E:

$$\mathrm{s\_weight}(i) = \sum_{j=1}^{m} \alpha^{\mathrm{num\_inc}(c_j, E)} \cdot w_{ij}. \tag{5}$$

Therein, $\mathrm{num\_inc}(c_j, E)$ denotes the number of times in which summary sentence E covers information fragment c_j before sentence s_i; and α is a $[0, 1]$ parameter to control the latitude of redundant information. We call this parameter α *the duplicate information rate*. Setting $0 \leq \alpha < 1$ and applying Formulas 3 and 4, the extraction method favors a sentence having many novel (i.e., not included to the summary sentences) information fragments because the importance of covered information fragments is estimated as lower by Formula 5. Consequently, the extraction method preferentially selects sentences with novel in-

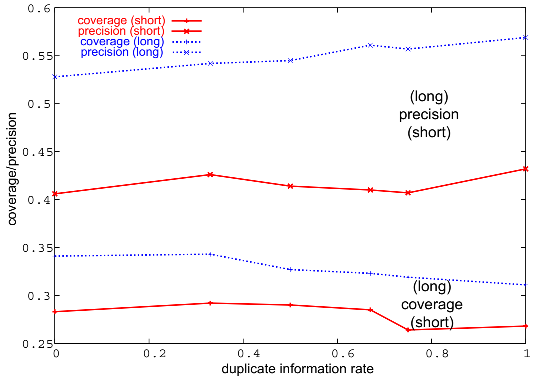

Figure 4. Coverage and precision.

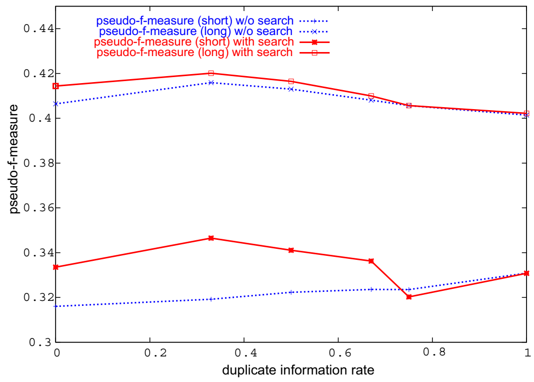

Figure 5. Effect of exploratory extraction.

formation instead of redundant ones.

Incidentally, it is difficult to find a summary E that maximizes F in Formula 3. Therefore, we introduce a search tree in which: a node represents a sentence; expanding a node corresponds to a trial to select a subsequent sentence; and summation of sentence weights from a root node to a leaf node is equivalent to the score of a summary to maximize. We find a quasi-optimal solution using beam search where a beam width is determined by summary length L [3] and acquires a set of important sentences.

2.3 Evaluation

We do not describe the evaluation methods/results at TSC-3 in this paper due to the limitation of space. Refer the TSC-3 task overview [3] for complete description of the evaluation methods/results. Table 4 in the paper [3] shows an evaluation result of content coverage by human subjects, which is to demonstrate quality of important sentence extraction. Our system (MOGS) [4] performed well (3rd place; above average) for both short and long summaries although we did not use question answering data in TSC-3 corpus.

We conducted experiments to test impacts of duplicate information rate and exploratory extraction. Figure 4 presents trends of coverage and precision when we extract sentences with several values of duplicate information rate. The coverage roughly decreases as the duplicate information rate approaches to 1. On the other hand, the precision roughly decreases as the rate approaches to 0 because our method comes to reject including redundant sentences even if the sentence is considered as important. Figure 4 also shows that $\alpha = 0.33$ was optimal through this test. Figure 5 shows trends of pseudo-f-measure [5] to test impact of exploratory extraction. We find an good effect of exploratory extraction for short summaries.

3 Improving Chronological Sentence Ordering

It is necessary to work out a good arrangement of sentences extracted from multiple documents when we generate a well-organized summary. Barzilay et. al. [1] address the problem of sentence ordering in the context of multi-document summarization and propose an algorithm that utilizes topical segmentation and chronological ordering. Lapata [5] 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.

The remainder of this section 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 to an issue of chronological ordering and explain our approach to improve chronological ordering by complementing on the presupposed information of each sentence. Then we address evaluation metrics to validate the effectiveness of our algorithm in MDS and show experimental results.

3.1 Improving chronological ordering

Against the background of these studies, we propose the use of antecedent sentences to arrange sentences coherently. Let us consider the example shown in Fig. 6. There are three sentences, `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.

[3] We determine a beam width automatically on a basis of summary length L because a longer summary requires a large search domain. The width ranges from 3 to 10.

[4] We set *duplicate information rate* α to be 0.

[5] We define pseudo-f-measure to be $\frac{2cp}{c+p}$, where c represents to coverage and p to precision.

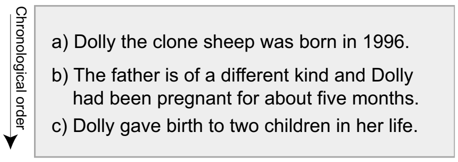

Figure 6. A problem case of chronological sentence ordering.

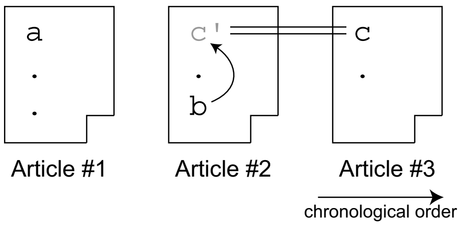

Figure 7. Background idea of ordering refinement by precedence relation.

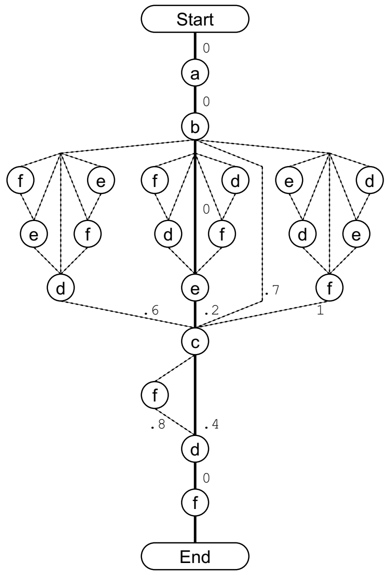

Figure 8. Ordering refinement by precedence relation as a shortest path problem. Figures around edges represent distance to put a next sentence.

Because sentence `b` is written on the presupposition that the reader may know that Dolly had a child. In other words, it is more fitting to assume sentence `b` to be an elaboration of sentence `c`. There are some precedent sentences prior to sentence `b` in the original document. Lack of presupposition obscures what a sentence is saying and confuses readers. Hence, we should refine the chronological order and revise the order to `[a-c-b]`, putting sentence `c` before sentence `b`.

Figure 7 shows the background idea of ordering refinement using a precedence relation. Just as in the example in Fig. 6, we have three sentences `a`, `b`, and `c` in chronological order. 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 as acceptable to put sentence `a` here. Then we get sentence `b` out of the 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 their subject content 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 for a substitution for what the precedent sentences are saying from the remaining sentences (i.e., sentence `c` in this example). In the Fig. 7 example, we find that sentence `a` is not referring to what sentence `c′` is saying, but sentence `c` is approximately referring to that content. Putting sentence `c` before `b`, we finally archive the refined ordering `[a-c-b]`.

3.2 Ordering algorithm

We order sentences by the chronological order in advance, assigning a time stamp for each sentence by its publication date (i.e., the date when the article was written). If there are sentences having the same time stamp, we elaborate the order on the basis of sentence position and sentence connectivity. We restore 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 not from the same article, we put a sentence which is more similar to previously ordered sentences.

Then we improve the ordering on the basis of antecedence sentences. Figure 8 illustrates how the algorithm refines a given chronological ordering `[a-b-c-d-e-f]`. We define *distance to put a sentence* as dissimilarity between precedent sentences of an arranging sentence and the previously arranged sentences. When a sentence has antecedent sentences and their content is not mentioned by previously arranged sentences, the *distance* will be high. When a sentence has no precedent sentences, we define the *distance* to be 0. In Figure 8 example we leave positions of sentences `a` and `b` because they do not have precedent sentences (i.e., they are lead sentences). On the other hand, sentence `c` has some precedent sentences in its original document. Preparing a term vector of the precedent sentences, we calculate how much the precedent content is covered by other sentences using the *distance* defined above. We search a shortest path from sentence c to sentences a and b by best-first search. Given that sentence e in Figure 8 describes similar content as the precedent sentences of sentence c and is a lead sentence, we trace the shortest path from sentence c to sentences a and b via sentence e. We extend the resultant ordering to [a-b-e-c], inserting sentence e before sentence 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 shortest path from sentence d to sentences a, b, c, and e. We leave sentence d this time because the precedent content seems to be described in sentences a, b, c, and e. In this way we get the final ordering, [a-b-e-c-d-f].

3.3 Experiment

We conducted an experiment of sentence ordering through multi-document summarization to test the effectiveness of the proposed method. We ordered the extracted sentences for long summaries by six methods: *human-made ordering (HO)* as the highest anchor; *random ordering (RO)* as the lowest anchor; *chronological ordering (CO)*; *chronological ordering with topical segmentation (COT)* (i.e., the argumented methods in [1, 7]); *proposed method without topical segmentation (PO)*; and *proposed method with topical segmentation (POT)*. We asked human judges to evaluate sentence ordering of these summaries.

The first evaluation task is a subjective grading where a human judge marks an ordering of summary sentences on a scale of 4: 4 (*perfect*: we cannot improve any further), 3 (*acceptable*: it makes sense even though there is some room for improvement), 2 (*poor*: it requires minor amendment to bring it up to the acceptable level), and 1 (*unacceptable*: it requires overall restructuring rather than partial revision). In addition to the rating, 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 so as 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 9-(1)).

Supposing a sentence ordering to be a rank, we can calculate rank correlation coefficient of permutations of an ordering π and of the reference ordering σ (see Figure 9-(2)). Spearman's rank correlation $\tau_s(\pi,\sigma)$ and Kendall's rank correlation $\tau_k(\pi,\sigma)$ are known as famous rank correlation metrics. These metrics range from -1 (an inverse rank) to 1 (an identical rank) via 0 (a non-correlated rank). We obtain $\tau_s(\pi,\sigma) = 0.85$ and $\tau_k(\pi,\sigma) = 0.72$ in the example shown in Figure 9-(2). We propose another metric to assess the degree

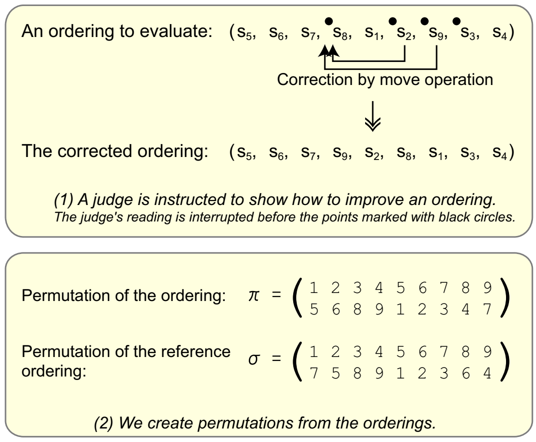

Figure 9. Correction of an ordering.

of sentence continuity in reading $\tau_c(\pi,\sigma)$:

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

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 9-(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 $\tau_c(\pi,\sigma) = (9-4)/9 = 0.56$.

3.4 Evaluation results

Table 1 shows distribution of rating score of each method in percent figures. 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 for the TSC-3 corpus [6] . On the other hand, the proposed method (PO) improved chronological ordering much better than topical segmentation. Note that sum of perfect and acceptable ratio jumped up from 36% (CO) to 55% (PO). This shows ordering refinement by precedence relation improves CO by pushing poor ordering to an acceptable level.

[6] We suppose 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 so as not to include unrelated articles to a topic.

	Perfect	Acceptable	Poor	Unacceptable
RO	0.0	0.0	6.0	94.0
CO	13.1	22.6	63.1	1.2
COT	10.7	22.6	61.9	4.8
PO	16.7	38.1	45.2	0.0
POT	15.5	36.9	44.0	3.6
HO	52.4	21.4	26.2	0.0

Table 1. Distribution of rating score of orderings in percent figures.

	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 2. Comparison with corrected ordering.

Table 2 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 show 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.

4 Other Components

The reminder of this paper reports an outline of other components in our summarization system including simple anaphora analysis and sentence compression shown in Fig. 1.

A newspaper article often substitutes a named entity with an anaphoric expression when the named entity occurs more than twice in the article. Figure 10 shows a typical example of the anaphoric reference by a Japanese term *dou* [7]. Because a term *dou* is a common expression of anaphoric reference used in

[7]The meaning of *dou* is close to *the* in English, although the usages of *dou* and *the* are quite different.

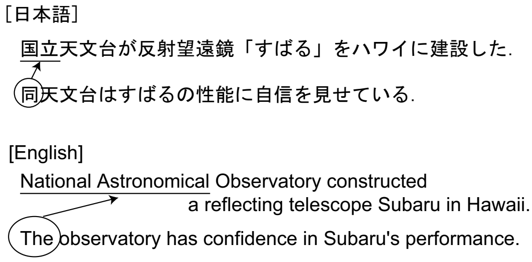

Figure 10. A typical example of anaphoric reference by a Japanese term 'dou'.

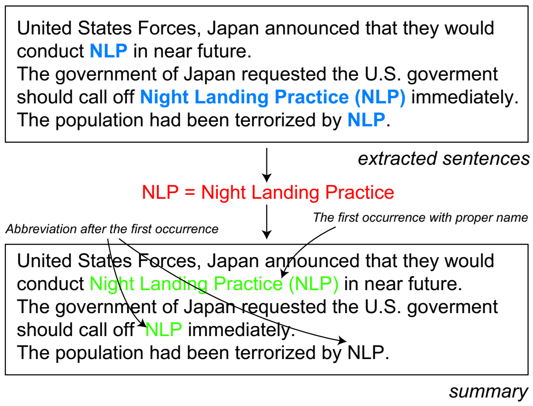

Figure 11. Entreatment of abbreviations.

Japanese newspaper articles, we replace it with the named entity to which the *dou* refers. We take advantage of two kinds of constraints to find a referred named entity from prior sentences: the identity of the succeeding term (i.e., finding a noun phrase just before the term 'observatory' in the example); and the type of named entity (i.e., finding a named entity tagged as a country name when we resolve *the country*). We replaced 90% of the anaphoric terms *dou* successfully in our summary.

Regarding sentence compression, we employ two components: *entreatment of abbreviation and proper name* (Fig. 11) and *redundant clause elimination* (Fig. 12). Figure 11 demonstrates how we standardize notations of abbreviated names. Every time we encounter an alphanumeric phrase in parentheses, we find it to be an abbreviation and the adjacent noun phrase to be the proper name. When we generate a summary, we replace the first occurrence of the abbreviation or proper name with a standardized form — "*a proper name* (*the abbreviation*)". After the first occurrence, we put only the abbreviation to save letters for other information. We recognized 0.45 kinds of abbreviations and replaced 1.2 proper names with abbreviated terms in each summary.

Figure 12 illustrates redundant clause elimination.

［日本語］

ソニーは６月１日よりペットロボットＡＩＢＯの予約をインターネット上で受け付ける.

~~ソニーが１日午前９時から予約を始めた~~ペット型ロボットＡＩＢＯが，受け付け開始から２０分後に完売した.

[English]

Sony will accept reservations for AIBO the Entertainment Robot on the Internet on June 1st.

AIBO the Entertainment Robot for ~~which Sony started to accept reservations at 9 a.m. on the 1st~~ was sold out within 20 minutes.

Figure 12. Redundant clauses.

Extracting long (longer than 25 letter) clauses modifying a noun phrase, we perform DP matching for all extracted clauses. We regard a pair of clauses that are closer than a given distance as similar clauses. In the summary generation phase, we delete clauses which are similar to previously-included clauses on the basis of redundancy analysis: our system removed 3.4% letters from extracted sentences.

5 Readability Evaluation in TSC-3

In this section we describe results of readability evaluation in TSC-3 since we have already mentioned the result of content coverage in Section 2. Table 5 in the task overview [3] presents evaluation results in terms of readability by human subjects. qq0 measures the number of redundant or unnecessary sentences in submitted summaries. Our system (MOGS) hardly includes redundant sentences (0.067 redundant sentences for a short summary and 0.167 sentences for a long summary on average). This result shows an excellent effects of the argumented sentence extraction and redundant clause elimination.

The rest of quality evaluations (q01...q15) targets at sentences which were not marked as redundant in the qq0 evaluation. Since the number of redundant sentences in our summaries are extremely small, there are a large number of target sentences left for quality evaluations. That is to say we cannot compare the figures directly between the systems. Our system makes an attempt to improve readability concerning q02 and q08 in a positive way. q02 reports the number of pronouns that lose antecedents. Our system yields 0.433 isolated pronouns for a short summary and 0.833 for a long summary. These figures are smaller than system average (0.767 for short and 1.388 for long). q08 inquires the degree of wrong chronological ordering in a summary. The evaluation result shows that our system was above average although we sacrificed accurate chronological order in favor of readability.

6 Conclusion

In this paper we described our integrated summarization system for TSC-3, focusing on important sentence extraction and sentence ordering. The argumented method of important sentence extraction performed well for both short and long summaries according to the evaluation result of content coverage in TSC-3. The proposed method of sentence ordering which utilizes precedence relation also archived good results, raising poor chronological orderings to an acceptable level by 20%. In future work we will make an evaluation of other components such as anaphora analysis and explore for a better summary.

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. Goldstein, V. Mittal, J. Carbonell, and M. Kantrowitz. Multi-document summarization by sentence extraction. In *The ANLP/NAACL2000 Workshop on Automatic Summarization*, pages 40–48, 2000.

[3] T. Hirao, M. Okumura, T. Fukushima, and H. Nanba. Text summarization challenge 3 — text summarization evaluation at ntcir workshop 3 —. In *Working Notes of the 4th NTCIR Workshop Meeting, Part ?: Text Summarization Challenge 3 (TSC3)*, to appear in 2004.

[4] T. Kudo and Y. Matsumoto. Japanese dependency analysis using cascaded chunking. In *CoNLL 2002: Proceedings of the 6th Conference on Natural Language Learning 2002 (COLING 2002 Post-Conference Workshops)*, pages 63–69, 2002.

[5] 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.

[6] K. Nagao and K. Hasida. Automatic text summarization based on the global document annotation. In *Proceedings of the 17th International Conference on Computational Linguistics / 36th Annual Meeting of the Association for Computatinal Linguistics (COLING-ACL '98)*, pages 917–921, Montreal, Quebec, Canada, Aug. 1998.

[7] N. Okazaki, Y. Matsuo, N. Matsumura, H. Tomobe, and M. Ishizuka. Two different methods at NTCIR3-TSC2: Coverage oriented and focus oriented. In *Working Notes of the Third NTCIR Workshop Meeting, Part V: Text Summarization Challenge2 (TSC2)*, pages 39–46, 2002.

[8] D. R. Radev, H. Jing, and M. Budzikowska. Centroid-based summarization of multiple documents: Sentence extraction, utility-based evaluation, and user studies. In *The ANLP/NAACL2000 Workshop on Automatic Summarization*, pages 21–30, 2000.