A Bottom-up Approach to Sentence Ordering for Multi-document Summarization

Danushka Bollegala **Naoaki Okazaki** [*] **Mitsuru Ishizuka**
Graduate School of Information Science and Technology
The University of Tokyo
7-3-1, Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan
{danushka,okazaki}@mi.ci.i.u-tokyo.ac.jp
ishizuka@i.u-tokyo.ac.jp

Abstract

Ordering information is a difficult but important task for applications generating natural-language text. We present a bottom-up approach to arranging sentences extracted for multi-document summarization. To capture the association and order of two textual segments (eg, sentences), we define four criteria, *chronology*, *topical-closeness*, *precedence*, and *succession*. These criteria are integrated into a criterion by a supervised learning approach. We repeatedly concatenate two textual segments into one segment based on the criterion until we obtain the overall segment with all sentences arranged. Our experimental results show a significant improvement over existing sentence ordering strategies.

1 Introduction

Multi-document summarization (MDS) (Radev and McKeown, 1999) tackles the information overload problem by providing a condensed version of a set of documents. Among a number of sub-tasks involved in MDS, eg, sentence extraction, topic detection, sentence ordering, information extraction, sentence generation, etc., most MDS systems have been based on an extraction method, which identifies important textual segments (eg, sentences or paragraphs) in source documents. It is important for such MDS systems to determine a coherent arrangement of the textual segments extracted from multi-documents in order to reconstruct the text structure for summarization. Ordering information is also essential for other text-generation applications such as Question Answering.

A summary with improperly ordered sentences confuses the reader and degrades the quality/reliability of the summary itself. Barzilay (2002) has provided empirical evidence that proper order of extracted sentences improves their readability significantly. However, ordering a set of sentences into a coherent text is a non-trivial task. For example, identifying rhetorical relations (Mann and Thompson, 1988) in an ordered text has been a difficult task for computers, whereas our task is even more complicated: to reconstruct such relations from unordered sets of sentences. Source documents for a summary may have been written by different authors, by different writing styles, on different dates, and based on different background knowledge. We cannot expect that a set of extracted sentences from such diverse documents will be coherent on their own.

Several strategies to determine sentence ordering have been proposed as described in section 2. However, the appropriate way to combine these strategies to achieve more coherent summaries remains unsolved. In this paper, we propose four criteria to capture the association of sentences in the context of multi-document summarization for newspaper articles. These criteria are integrated into one criterion by a supervised learning approach. We also propose a bottom-up approach in arranging sentences, which repeatedly concatenates textual segments until the overall segment with all sentences arranged, is achieved.

2 Related Work

Existing methods for sentence ordering are divided into two approaches: making use of chronological information (McKeown et al., 1999; Lin

[*] Research Fellow of the Japan Society for the Promotion of Science (JSPS)

Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, pages 385–392,
Sydney, July 2006. ©2006 Association for Computational Linguistics

and Hovy, 2001; Barzilay et al., 2002; Okazaki et al., 2004); and learning the natural order of sentences from large corpora not necessarily based on chronological information (Lapata, 2003; Barzilay and Lee, 2004). A newspaper usually disseminates descriptions of novel events that have occurred since the last publication. For this reason, ordering sentences according to their publication date is an effective heuristic for multidocument summarization (Lin and Hovy, 2001; McKeown et al., 1999). Barzilay et al. (2002) have proposed an improved version of chronological ordering by first grouping sentences into sub-topics discussed in the source documents and then arranging the sentences in each group chronologically.

Okazaki et al. (2004) have proposed an algorithm to improve chronological ordering by resolving the presuppositional information of extracted sentences. They assume that each sentence in newspaper articles is written on the basis that presuppositional information should be transferred to the reader before the sentence is interpreted. The proposed algorithm first arranges sentences in a chronological order and then estimates the presuppositional information for each sentence by using the content of the sentences placed before each sentence in its original article. The evaluation results show that the proposed algorithm improves the chronological ordering significantly.

Lapata (2003) has suggested a probabilistic model of text structuring and its application to the sentence ordering. Her method calculates the transition probability from one sentence to the next from a corpus based on the Cartesian product between two sentences defined using the following features: verbs (precedent relationships of verbs in the corpus); nouns (entity-based coherence by keeping track of the nouns); and dependencies (structure of sentences). Although she has not compared her method with chronological ordering, it could be applied to generic domains, not relying on the chronological clue provided by newspaper articles.

Barzilay and Lee (2004) have proposed *content models* to deal with topic transition in domain specific text. The content models are formalized by Hidden Markov Models (HMMs) in which the hidden state corresponds to a topic in the domain of interest (eg, earthquake magnitude or previous earthquake occurrences), and the state transitions capture possible information-presentation orderings. The evaluation results showed that their method outperformed Lapata's approach by a wide margin. They did not compare their method with chronological ordering as an application of multi-document summarization.

As described above, several good strategies/heuristics to deal with the sentence ordering problem have been proposed. In order to integrate multiple strategies/heuristics, we have formalized them in a machine learning framework and have considered an algorithm to arrange sentences using the integrated strategy.

3 Method

We define notation $a \succ b$ to represent that sentence a precedes sentence b. We use the term *segment* to describe a sequence of ordered sentences. When segment A consists of sentences a_1, a_2, ..., a_m in this order, we denote as:

$$A = (a_1 \succ a_2 \succ ... \succ a_m). \qquad (1)$$

The two segments A and B can be ordered either B after A or A after B. We define the notation $A \succ B$ to show that segment A precedes segment B.

Let us consider a bottom-up approach in arranging sentences. Starting with a set of segments initialized with a sentence for each, we concatenate two segments, with the strongest association (discussed later) of all possible segment pairs, into one segment. Repeating the concatenating will eventually yield a segment with all sentences arranged. The algorithm is considered as a variation of agglomerative hierarchical clustering with the ordering information retained at each concatenating process.

The underlying idea of the algorithm, a bottom-up approach to text planning, was proposed by Marcu (1997). Assuming that the semantic units (sentences) and their rhetorical relations (eg, sentence *a* is an *elaboration* of sentence *d*) are given, he transcribed a text structuring task into the problem of finding the best discourse tree that satisfied the set of rhetorical relations. He stated that global coherence could be achieved by satisfying local coherence constraints in ordering and clustering, thereby ensuring that the resultant discourse tree was well-formed.

Unfortunately, identifying the rhetorical relation between two sentences has been a difficult

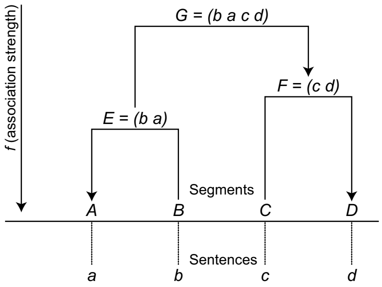

Figure 1: Arranging four sentences A, B, C, and D with a bottom-up approach.

task for computers. However, the bottom-up algorithm for arranging sentences can still be applied only if the direction and strength of the association of the two segments (sentences) are defined. Hence, we introduce a function $f(A \succ B)$ to represent the direction and strength of the association of two segments A and B,

$$f(A \succ B) = \begin{cases} p & \text{(if } A \text{ precedes } B\text{)} \\ 0 & \text{(if } B \text{ precedes } A\text{)} \end{cases}, \tag{2}$$

where p ($0 \leq p \leq 1$) denotes the association strength of the segments A and B. The association strengths of the two segments with different directions, eg, $f(A \succ B)$ and $f(B \succ A)$, are not always identical in our definition,

$$f(A \succ B) \neq f(B \succ A). \tag{3}$$

Figure 1 shows the process of arranging four sentences a, b, c, and d. Firstly, we initialize four segments with a sentence for each,

$$A = (a), B = (b), C = (c), D = (d). \tag{4}$$

Suppose that $f(B \succ A)$ has the highest value of all possible pairs, eg, $f(A \succ B)$, $f(C \succ D)$, etc, we concatenate B and A to obtain a new segment,

$$E = (b \succ a). \tag{5}$$

Then we search for the segment pair with the strongest association. Supposing that $f(C \succ D)$ has the highest value, we concatenate C and D to obtain a new segment,

$$F = (c \succ d). \tag{6}$$

Finally, comparing $f(E \succ F)$ and $f(F \succ E)$, we obtain the global sentence ordering,

$$G = (b \succ a \succ c \succ d). \tag{7}$$

In the above description, we have not defined the association of the two segments. The previous work described in Section 2 has addressed the association of textual segments (sentences) to obtain coherent orderings. We define four criteria to capture the association of two segments: *chronology*; *topical-closeness*; *precedence*; and *succession*. These criteria are integrated into a function $f(A \succ B)$ by using a machine learning approach. The rest of this section explains the four criteria and an integration method with a Support Vector Machine (SVM) (Vapnik, 1998) classifier.

3.1 Chronology criterion

Chronology criterion reflects the chronological ordering (Lin and Hovy, 2001; McKeown et al., 1999), which arranges sentences in a chronological order of the publication date. We define the association strength of arranging segments B after A measured by a chronology criterion $f_{\text{chro}}(A \succ B)$ in the following formula,

$$\begin{aligned} & f_{\text{chro}}(A \succ B) \\ = & \begin{cases} 1 & \mathrm{T}(a_m) < \mathrm{T}(b_1) \\ 1 & [\mathrm{D}(a_m) = \mathrm{D}(b_1)] \wedge [\mathrm{N}(a_m) < \mathrm{N}(b_1)] \\ 0.5 & [\mathrm{T}(a_m) = \mathrm{T}(b_1)] \wedge [\mathrm{D}(a_m) \neq \mathrm{D}(b_1)] \\ 0 & \text{otherwise} \end{cases}. \end{aligned} \tag{8}$$

Here, a_m represents the last sentence in segment A; b_1 represents the first sentence in segment B; $T(s)$ is the publication date of the sentence s; $D(s)$ is the unique identifier of the document to which sentence s belongs: and $N(s)$ denotes the line number of sentence s in the original document. The chronological order of arranging segment B after A is determined by the comparison between the last sentence in the segment A and the first sentence in the segment B.

The chronology criterion assesses the appropriateness of arranging segment B after A if: sentence a_m is published earlier than b_1; or sentence a_m appears before b_1 in the same article. If sentence a_m and b_1 are published on the same day but appear in different articles, the criterion assumes the order to be undefined. If none of the above conditions are satisfied, the criterion estimates that segment B will precede A.

3.2 Topical-closeness criterion

The topical-closeness criterion deals with the association, based on the topical similarity, of two

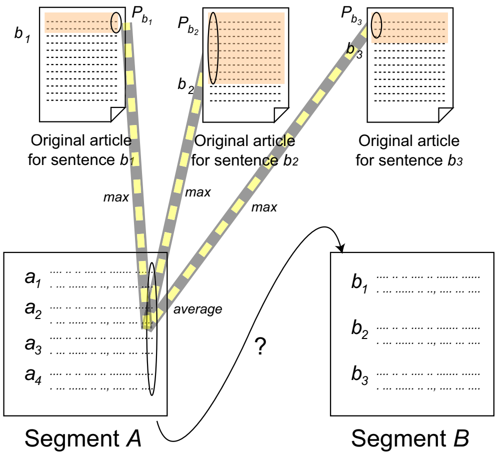

Figure 2: Precedence criterion

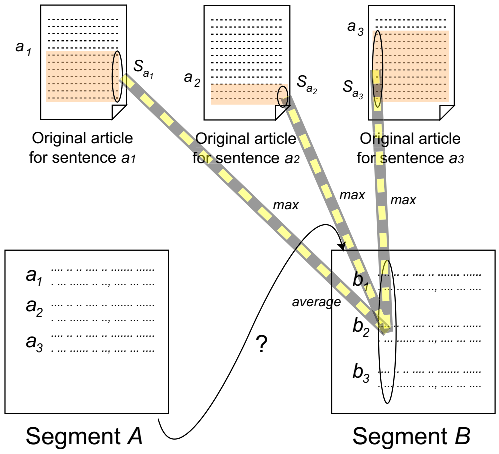

Figure 3: Succession criterion

segments. The criterion reflects the ordering strategy proposed by Barzilay et al (2002), which groups sentences referring to the same topic. To measure the topical closeness of two sentences, we represent each sentence with a vector whose elements correspond to the occurrence[1] of the nouns and verbs in the sentence. We define the topical closeness of two segments A and B as follows,

$$f_{\text{topic}}(A \succ B) = \frac{1}{|B|} \sum_{b \in B} \max_{a \in A} \text{sim}(a, b). \quad (9)$$

Here, $\text{sim}(a, b)$ denotes the similarity of sentences a and b, which is calculated by the cosine similarity of two vectors corresponding to the sentences. For sentence $b \in B$, $\max_{a \in A} \text{sim}(a, b)$ chooses the sentence $a \in A$ most similar to sentence b and yields the similarity. The topical-closeness criterion $f_{\text{topic}}(A \succ B)$ assigns a higher value when the topic referred by segment B is the same as segment A.

3.3 Precedence criterion

Let us think of the case where we arrange segment A before B. Each sentence in segment B has the presuppositional information that should be conveyed to a reader in advance. Given sentence $b \in B$, such presuppositional information may be presented by the sentences appearing before the sentence b in the original article. However, we cannot guarantee whether a sentence-extraction method for multi-document summarization chooses any sentences before b for a summary because the extraction method usually determines a set of sentences, within the constraint of summary length, that maximizes information coverage and excludes redundant information. *Precedence criterion* measures the substitutability of the presuppositional information of segment B (eg, the sentences appearing before sentence b) as segment A. This criterion is a formalization of the sentence-ordering algorithm proposed by Okazaki et al, (2004).

We define the precedence criterion in the following formula,

$$f_{\text{pre}}(A \succ B) = \frac{1}{|B|} \sum_{b \in B} \max_{a \in A, p \in P_b} \text{sim}(a, p). \quad (10)$$

Here, P_b is a set of sentences appearing before sentence b in the original article; and $\text{sim}(a, b)$ denotes the cosine similarity of sentences a and b (defined as in the topical-closeness criterion). Figure 2 shows an example of calculating the precedence criterion for arranging segment B after A. We approximate the presuppositional information for sentence b by sentences P_b, ie, sentences appearing before the sentence b in the original article. Calculating the similarity among sentences in P_b and A by the maximum similarity of the possible sentence combinations, Formula 10 is interpreted as the average similarity of the precedent sentences $\forall P_b (b \in B)$ to the segment A.

3.4 Succession criterion

The idea of *succession criterion* is the exact opposite of the precedence criterion. The succession criterion assesses the coverage of the succedent information for segment A by arranging segment B

[1] The vector values are represented by boolean values, i.e., 1 if the sentence contains a word, otherwise 0.

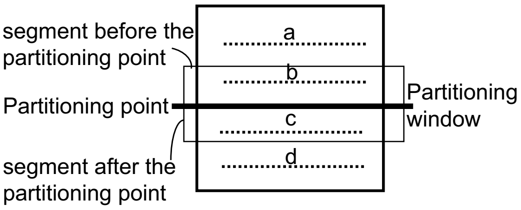
segment before the partitioning point
a
b
Partitioning point
c
d
Partitioning window
segment after the partitioning point

Figure 4: Partitioning a human-ordered extract into pairs of segments

after A:

$$f_{\mathrm{succ}}(A \succ B) = \frac{1}{|A|} \sum_{a \in A} \max_{s \in S_a, b \in B} \mathrm{sim}(s, b). \tag{11}$$

Here, S_a is a set of sentences appearing after sentence a in the original article; and $\mathrm{sim}(a, b)$ denotes the cosine similarity of sentences a and b (defined as in the topical-closeness criterion). Figure 3 shows an example of calculating the succession criterion to arrange segments B after A. The succession criterion measures the substitutability of the succedent information (eg, the sentences appearing after the sentence $a \in A$) as segment B.

3.5 SVM classifier to assess the integrated criterion

We integrate the four criteria described above to define the function $f(A \succ B)$ to represent the association direction and strength of the two segments A and B (Formula 2). More specifically, given the two segments A and B, function $f(A \succ B)$ is defined to yield the integrated association strength from four values, $f_{\mathrm{chro}}(A \succ B)$, $f_{\mathrm{topic}}(A \succ B)$, $f_{\mathrm{pre}}(A \succ B)$, and $f_{\mathrm{succ}}(A \succ B)$. We formalize the integration task as a binary classification problem and employ a Support Vector Machine (SVM) as the classifier. We conducted a supervised learning as follows.

We partition a human-ordered extract into pairs each of which consists of two non-overlapping segments. Let us explain the partitioning process taking four human-ordered sentences, $a \succ b \succ c \succ d$ shown in Figure 4. Firstly, we place the partitioning point just after the first sentence a. Focusing on sentence a arranged just before the partition point and sentence b arranged just after we identify the pair $\{(a), (b)\}$ of two segments (a) and (b). Enumerating all possible pairs of two segments facing just before/after the partitioning point, we obtain the following pairs, $\{(a), (b \succ c)\}$ and $\{(a), (b \succ c \succ d)\}$. Similarly, segment pairs, $\{(b), (c)\}$, $\{(a \succ b), (c)\}$, $\{(b), (c \succ d)\}$, $\{(a \succ b), (c \succ d)\}$, are obtained from the partitioning point between sentence b and c. Collecting the segment pairs from the partitioning point between sentences c and d (i.e., $\{(c), (d)\}$, $\{(b \succ c), (d)\}$ and $\{(a \succ b \succ c), (d)\}$), we identify ten pairs in total form the four ordered sentences. In general, this process yields $n(n^2 - 1)/6$ pairs from ordered n sentences. From each pair of segments, we generate one positive and one negative training instance as follows.

$$+1 : [f_{\mathrm{chro}}(A \succ B), f_{\mathrm{topic}}(A \succ B), f_{\mathrm{pre}}(A \succ B), f_{\mathrm{succ}}(A \succ B)]$$
$$-1 : [f_{\mathrm{chro}}(B \succ A), f_{\mathrm{topic}}(B \succ A), f_{\mathrm{pre}}(B \succ A), f_{\mathrm{succ}}(B \succ A)]$$

Figure 5: Two vectors in a training data generated from two ordered segments $A \succ B$

Given a pair of two segments A and B arranged in an order $A \succ B$, we calculate four values, $f_{\mathrm{chro}}(A \succ B)$, $f_{\mathrm{topic}}(A \succ B)$, $f_{\mathrm{pre}}(A \succ B)$, and $f_{\mathrm{succ}}(A \succ B)$ to obtain the instance with the four-dimensional vector (Figure 5). We label the instance (corresponding to $A \succ B$) as a positive class (ie, $+1$). Simultaneously, we obtain another instance with a four-dimensional vector corresponding to $B \succ A$. We label it as a negative class (ie, -1). Accumulating these instances as training data, we obtain a binary classifier by using a Support Vector Machine with a quadratic kernel. The SVM classifier yields the association direction of two segments (eg, $A \succ B$ or $B \succ A$) with the class information (ie, $+1$ or -1). We assign the association strength of two segments by using the class probability estimate that the instance belongs to a positive ($+1$) class. When an instance is classified into a negative (-1) class, we set the association strength as zero (see the definition of Formula 2).

4 Evaluation

We evaluated the proposed method by using the 3rd Text Summarization Challenge (TSC-3) corpus[2]. The TSC-3 corpus contains 30 sets of extracts, each of which consists of unordered sentences[3] extracted from Japanese newspaper articles relevant to a topic (query). We arrange the extracts by using different algorithms and evaluate

[2] http://lr-www.pi.titech.ac.jp/tsc/tsc3-en.html

[3] Each extract consists of ca. 15 sentences on average.

Table 1: Correlation between two sets of human-ordered extracts

Metric	Mean	Std. Dev	Min	Max
Spearman	0.739	0.304	-0.2	1
Kendall	0.694	0.290	0	1
Average Continuity	0.401	0.404	0.001	1

the readability of the ordered extracts by a subjective grading and several metrics.

In order to construct training data applicable to the proposed method, we asked two human subjects to arrange the extracts and obtained $30(\text{topics}) \times 2(\text{humans}) = 60$ sets of ordered extracts. Table 1 shows the agreement of the ordered extracts between the two subjects. The correlation is measured by three metrics, Spearman's rank correlation, Kendall's rank correlation, and average continuity (described later). The mean correlation values (0.74 for Spearman's rank correlation and 0.69 for Kendall's rank correlation) indicate a certain level of agreement in sentence orderings made by the two subjects. 8 out of 30 extracts were actually identical.

We applied the leave-one-out method to the proposed method to produce a set of sentence orderings. In this experiment, the leave-out-out method arranges an extract by using an SVM model trained from the rest of the 29 extracts. Repeating this process 30 times with a different topic for each iteration, we generated a set of 30 extracts for evaluation. In addition to the proposed method, we prepared six sets of sentence orderings produced by different algorithms for comparison. We describe briefly the seven algorithms (including the proposed method):

Agglomerative ordering (AGL) is an ordering arranged by the proposed method;

Random ordering (RND) is the lowest anchor, in which sentences are arranged randomly;

Human-made ordering (HUM) is the highest anchor, in which sentences are arranged by a human subject;

Chronological ordering (CHR) arranges sentences with the chronology criterion defined in Formula 8. Sentences are arranged in chronological order of their publication date;

Topical-closeness ordering (TOP) arranges sentences with the topical-closeness criterion defined in Formula 9;

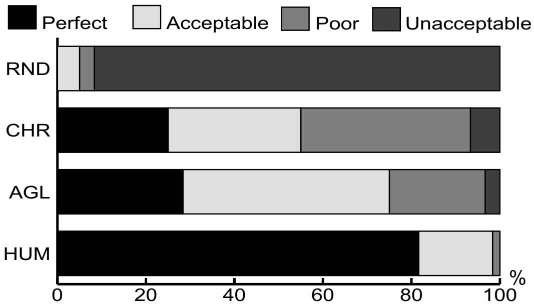

Figure 6: Subjective grading

Precedence ordering (PRE) arranges sentences with the precedence criterion defined in Formula 10;

Suceedence ordering (SUC) arranges sentences with the succession criterion defined in Formula 11.

The last four algorithms (CHR, TOP, PRE, and SUC) arrange sentences by the corresponding criterion alone, each of which uses the association strength directly to arrange sentences without the integration of other criteria. These orderings are expected to show the performance of each expert independently and their contribution to solving the sentence ordering problem.

4.1 Subjective grading

Evaluating a sentence ordering is a challenging task. Intrinsic evaluation that involves human judges to rank a set of sentence orderings is a necessary approach to this task (Barzilay et al., 2002; Okazaki et al., 2004). We asked two human judges to rate sentence orderings according to the following criteria. A *perfect* summary is a text that we cannot improve any further by re-ordering. An *acceptable* summary is one that makes sense and is unnecessary to revise even though there is some room for improvement in terms of readability. A *poor* summary is one that loses a thread of the story at some places and requires minor amendment to bring it up to an acceptable level. An *unacceptable* summary is one that leaves much to be improved and requires overall restructuring rather than partial revision. To avoid any disturbance in rating, we inform the judges that the summaries were made from a same set of extracted sentences and only the ordering of sentences is different.

Figure 6 shows the distribution of the subjective grading made by two judges to four sets of orderings, RND, CHR, AGL and HUM. Each set of or-

$$\begin{aligned} T_{eval} &= (e \succ a \succ b \succ c \succ d) \\ T_{ref} &= (a \succ b \succ c \succ d \succ e) \end{aligned}$$

Figure 7: An example of an ordering under evaluation T_{eval} and its reference T_{ref}.

derings has 30(topics) × 2(judges) $= 60$ ratings. Most RND orderings are rated as *unacceptable*. Although CHR and AGL orderings have roughly the same number of *perfect* orderings (ca. 25%), the AGL algorithm gained more *acceptable* orderings (47%) than the CHR alghrotihm (30%). This fact shows that integration of CHR experts with other experts worked well by pushing poor ordering to an acceptable level. However, a huge gap between *AGL* and *HUM* orderings was also found. The judges rated 28% AGL orderings as *perfect* while the figure rose as high as 82% for HUM orderings. Kendall's coefficient of concordance (Kendall's W), which asses the inter-judge agreement of overall ratings, reported a higher agreement between the two judges ($W = 0.939$).

4.2 Metrics for semi-automatic evaluation

We also evaluated sentence orderings by reusing two sets of gold-standard orderings made for the training data. In general, subjective grading consumes much time and effort, even though we cannot reproduce the evaluation afterwards. The previous studies (Barzilay et al., 2002; Lapata, 2003) employ rank correlation coefficients such as Spearman's rank correlation and Kendall's rank correlation, assuming a sentence ordering to be a rank. Okazaki et al. (2004) propose a metric that assess continuity of pairwise sentences compared with the gold standard. In addition to Spearman's and Kendall's rank correlation coefficients, we propose an *average continuity* metric, which extends the idea of the continuity metric to continuous k sentences.

A text with sentences arranged in proper order does not interrupt a human's reading while moving from one sentence to the next. Hence, the quality of a sentence ordering can be estimated by the number of continuous sentences that are also reproduced in the reference sentence ordering. This is equivalent to measuring a precision of continuous sentences in an ordering against the reference ordering. We define P_n to measure the precision of n continuous sentences in an ordering to be evaluated as,

$$P_n = \frac{m}{N - n + 1}. \qquad (12)$$

Here, N is the number of sentences in the reference ordering; n is the length of continuous sentences on which we are evaluating; m is the number of continuous sentences that appear in both the evaluation and reference orderings. In Figure 7, the precision of 3 continuous sentences P_3 is calculated as:

$$P_3 = \frac{2}{5 - 3 + 1} = 0.67. \qquad (13)$$

The Average Continuity (AC) is defined as the logarithmic average of P_n over 2 to k:

$$\text{AC} = \exp\left(\frac{1}{k-1}\sum_{n=2}^{k} \log(P_n + \alpha)\right). \qquad (14)$$

Here, k is a parameter to control the range of the logarithmic average; and α is a small value in case if P_n is zero. We set $k = 4$ (ie, more than five continuous sentences are not included for evaluation) and $\alpha = 0.01$. Average Continuity becomes 0 when evaluation and reference orderings share no continuous sentences and 1 when the two orderings are identical. In Figure 7, Average Continuity is calculated as 0.63. The underlying idea of Formula 14 was proposed by Papineni et al. (2002) as the BLEU metric for the semi-automatic evaluation of machine-translation systems. The original definition of the BLEU metric is to compare a machine-translated text with its reference translation by using the word n-grams.

Table 2: Comparison with human-made ordering

Method	Spearman coefficient	Kendall coefficient	Average Continuity
RND	-0.127	-0.069	0.011
TOP	0.414	0.400	0.197
PRE	0.415	0.428	0.293
SUC	0.473	0.476	0.291
CHR	0.583	0.587	0.356
AGL	0.603	0.612	0.459

4.3 Results of semi-automatic evaluation

Table 2 reports the resemblance of orderings produced by six algorithms to the human-made ones with three metrics, Spearman's rank correlation, Kendall's rank correlation, and Average Continuity. The proposed method (AGL) outperforms the

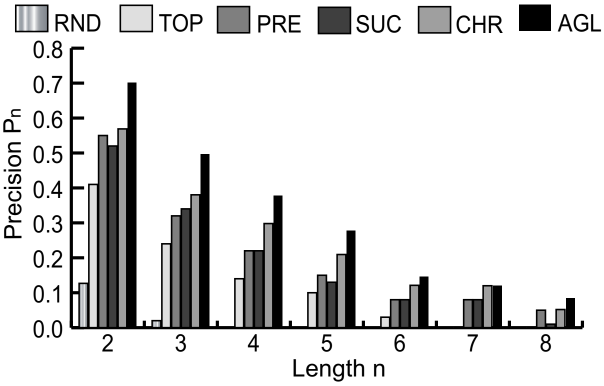

Figure 8: Precision vs unit of measuring continuity.

rest in all evaluation metrics, although the chronological ordering (CHR) appeared to play the major role. The one-way analysis of variance (ANOVA) verified the effects of different algorithms for sentence orderings with all metrics ($p < 0.01$). We performed Tukey Honest Significant Differences (HSD) test to compare differences among these algorithms. The Tukey test revealed that AGL was significantly better than the rest. Even though we could not compare our experiment with the probabilistic approach (Lapata, 2003) directly due to the difference of the text corpora, the Kendall coefficient reported higher agreement than Lapata's experiment (Kendall=0.48 with lemmatized nouns and Kendall=0.56 with verb-noun dependencies).

Figure 8 shows precision P_n with different length values of continuous sentence n for the six methods compared in Table 2. The number of continuous sentences becomes sparse for a higher value of length n. Therefore, the precision values decrease as the length n increases. Although RND ordering reported some continuous sentences for lower n values, no continuous sentences could be observed for the higher n values. Four criteria described in Section 3 (ie, CHR, TOP, PRE, SUC) produce segments of continuous sentences at all values of n.

5 Conclusion

We present a bottom-up approach to arrange sentences extracted for multi-document summarization. Our experimental results showed a significant improvement over existing sentence ordering strategies. However, the results also implied that chronological ordering played the major role in arranging sentences. A future direction of this study would be to explore the application of the proposed framework to more generic texts, such as documents without chronological information.

Acknowledgment

We used Mainichi Shinbun and Yomiuri Shinbun newspaper articles, and the TSC-3 test collection.

References

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

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

Mirella Lapata. 2003. Probabilistic text structuring: Experiments with sentence ordering. *Proceedings of the annual meeting of ACL, 2003.*, pages 545–552.

C.Y. Lin and E. Hovy. 2001. Neats:a multidocument summarizer. *Proceedings of the Document Understanding Workshop(DUC).*

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

Daniel Marcu. 1997. From local to global coherence: A bottom-up approach to text planning. In *Proceedings of the 14th National Conference on Artificial Intelligence*, pages 629–635, Providence, Rhode Island.

Kathleen McKeown, Judith Klavans, Vasileios Hatzivassiloglou, Regina Barzilay, and Eleazar Eskin. 1999. Towards multidocument summarization by reformulation: Progress and prospects. *AAAI/IAAI*, pages 453–460.

Naoaki Okazaki, Yutaka Matsuo, and Mitsuru Ishizuka. 2004. Improving chronological sentence ordering by precedence relation. In *Proceedings of 20th International Conference on Computational Linguistics (COLING 04)*, pages 750–756.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu:a method for automatic evaluation of machine translation. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318.

Dragomir R. Radev and Kathy McKeown. 1999. Generating natural language summaries from multiple on-line sources. *Computational Linguistics*, 24:469–500.

V. Vapnik. 1998. *Statistical Learning Theory*. Wiley, Chichester, GB.