2E1-02

A Clustering Based Approach to Sentence Ordering for Multidocument Summarization and its Evaluation

Danushka Bollegala Naoaki Okazaki Mitsuru Ishizuka

Graduate School of Information Science and Technology, The University of Tokyo

Ordering information is a difficult but a important task for natural language generation applications. A wrong order of information does not only make it difficult to understand, but also conveys an entirely different idea to the reader. This paper proposes an algorithm that learns orderings from a set of human ordered texts. We cluster sentences into locally coherent blocks and use a hierarchical clustering algorithm which merge these blocks to create a complete summary. Our experimental results show a significant improvement over the existing methods of sentence ordering.

1. Introduction

With the popularity of the internet, information available in electronic formats have grown rapidly. It has become impossible for users to go through all these information to find what they need. Therefore, efficient searching and summarization algorithms are a must. However, extracting information is not sufficient and we need to properly organize the extracted information. For example, in multidocument summarization (MDS), a summary is generated from a set of documents. The documents may belong to different topics and written by different authors. The information they contain may be contradictory or repetitive. Sentences extracted from such a diverse set of documents need to be properly ordered to create a coherent summary. Barzilay [1] shows that proper ordering of sentences improves readability of a summary.

In the case of news summarization, ordering the sentences according to their publication date is an effective heuristic [5, 8]. News events have the tendency to occur in a chronological order. Barzilay [1] proposes an improved version of this chronological ordering by grouping the sentences according to their topics. Without a proper pretext certain sentences are incomprehensible. Such constraints among sentences are called precedence relations. Okazaki [9] proposes a sentence ordering algorithm that use precedence relations among sentences to improve the chronological ordering. In addition to these studies which make use of chronological ordering, Lapata [4] proposes a probabilistic model of text structuring and its application to the sentence ordering. Her system calculates the conditional probabilities between sentences from a corpus and uses a greedy ordering algorithm to arrange sentences according to the conditional probabilities. Even though these previous studies proposed different strategies to decide the sentence ordering, the appropriate way to combine these different methods to obtain more robust and coherent text remains unknown. In our previous work [2] we proposed succedence as another strategy for sentence ordering and combined it with all the existing heurstics for the task of sentence ordering using Cohen's [3] hedge based ordering model.

Contact: {danushka,okazaki,ishizuka}@miv.t.u-tokyo.ac.jp

All the above mentioned strategies to sentence ordering has a top-down approach and tries to find a total order among sentences. However, in some cases it is impossible to decide the order between two sentences simply by considering those two sentences only. First we need to consider the sentences which are more clear in their order and then order the rest of the sentences to fix into this framework. Marcu [7] argues that global coherence can be achieved by satisfying local coherence constraints. He combines the locally coherent blocks in a tree (discourse tree) such that the adjacent nodes satisfy some rhetorical relation. Then the problem of planning a coherent text becomes a one of searching for the discourse tree that best satisfies the rhetorical relations. He uses Cocke-Kasami-Younger (CYK) parsing algorithm to generate all the valid trees and proposes greedy searching techniques to speed up the search.

However, in a set of sentences extracted from different documents, identifying the rhetorical relations is a difficult task. Therefore, we define a fitness measure for two blocks of texts to be cohesive based on chronology, topical relevance, precedence and succedence and use this fitness measure to merge the blocks in a hierarchical clustering manner. According to Halliday and Hasan [6] sentences are bound together by various cohesion relations. We should retain such blocks of texts in our summary for it to be comprehensible. Therefore, we take a bottom-up approach and first group the sentences which are locally coherent and then merge these groups to build a complete summary.

2. Method

We define the fitness of merging two blocks of texts using four different values; chronology, topical relatedness, precedence and succedence. Then we partition the human-made orderings into blocks and calculate the individual fitness values for each pair of blocks. We then train a two class support vector machine(SVM) [10] using these training blocks. We consider the class probability of support vector machine as the integrated fitness for a given pair of blocks. Using the trained SVM, we hierarchically cluster the blocks to construct a complete summary.

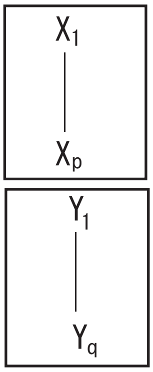

Figure 1: Comparing two blocks of texts

2.1 Chronology

Figure 1 shows two blocks of texts. Block A consists of p number of sentences $(X_1, \ldots, X_p)$ and block B has q number of sentences $(Y_1, \ldots, Y_q)$. The two blocks can be ordered B after A or A after B. We use the notation (A,B) to denote the ordering where block B comes after block A. Then, the chronological fitness of (A,B) is defined as follows.

$$\mathrm{F}_{chro}(A, B) \quad (1)$$
$$= \begin{cases} 1 & T(X_p) < T(Y_1) \\ 1 & [D(X_p) = D(Y_1)] \wedge [N(X_p) < N(Y_1)] \\ 0.5 & [T(X_p) = T(Y_1)] \wedge [D(X_p) \neq D(Y_1)] \\ 0 & \text{otherwise} \end{cases}$$

Therein: $T(u)$ is the publication date of sentence u; $D(u)$ presents the unique identifier of the document to which sentence u belongs; $N(u)$ denotes the line number of sentence u in the original document.

2.2 Topical Relevance

Grouping sentences according to their topics improves coherence of a text [6]. Motivated by this fact, we defined a topical relevance function for extract sentences [2]. We extend this concept to cover blocks of sentences by averaging the topical relevance of each sentence in the block. First we extract the nouns, verbs and connection words for each sentence and create the word vector for that sentence. Then, for each sentence $l \in B$, we define its topical relevance as follows.

$$\mathrm{topic}(l) = \max_{x \in A} \mathrm{sim}(l, x) \quad (2)$$

Here, we take the cosine similarity of the word vectors as $\mathrm{sim}(l, x)$. Using equation 2 we define the topical relevance of blocks (A,B) as follows.

$$F_{topic}(A, B) = \frac{1}{q} \sum_{y \in B} \mathrm{topic(y)} \quad (3)$$

2.3 Precedence

When placing a sentence in the summary it is important to check whether the preceding sentences convey the necessary background information for this sentence to be clearly understood. Placing a sentence without its context being stated in advanced, makes an unintelligible summary. Such constraints that must be satisfied by sentences are called precedence relations [9]. For example let us consider the situation illustrated in figure 2. Here, we are interested in the case where block A precedes block B as in figure 1. Sentence $l \in B$, is preceded by a block of text P in the source document D from which sentence l was extracted. The author of document D assumes that the block P is necessary to properly understand sentence l. Therefore, in our summary too we need to find a block A that matches best with P if we are to connect block B after block A. The similarity between block A and P is defined as precedence of sentence l and it is written as $\mathrm{pre}(l)$.

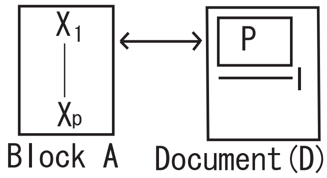

Figure 2: Precedence

$$\mathrm{pre}(l) = \max_{x \in A} \mathrm{sim}(l, x) \quad (4)$$

We define the precedence fitness of (A,B) as follows.

$$\mathrm{F}_{pre}(A, B) = \frac{1}{q} \sum_{y \in B} \mathrm{pre}(y) \quad (5)$$

2.4 Succedence

When summarizing a set of documents which belong to the same event, there may be more than one sentences which convey the same information. In order to avoid repetition of information, most summarization algorithms select only one sentence out of these multiple candidates to be included in the summary. However, the left out candidates provide valuable information to decide the sentence order. Succedence [2] is a measure of cross-document similarity which helps to recognize such hidden orderings.

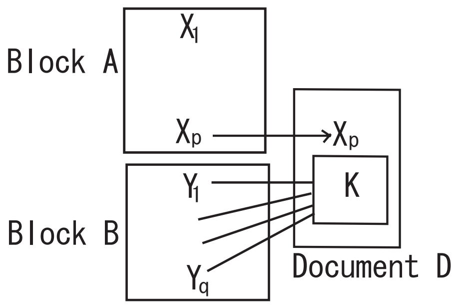

Figure 3: succedence

Let us consider the situation illustrated in figure 3 where block A is being placed ahead of block B. X_p is the last sentence of block A and it comes from document D. Succedence of sentence $y \in B$, $\mathrm{succ}(y)$, is calculated as the similarity between y and the block K of text that appears after the sentence X_p in document D. $\mathrm{succ}(y)$ defined by equation 6.

$$\mathrm{succ}(y) = \max_{l \in K} \mathrm{sim}(l, y) \tag{6}$$

We define the succedence of a block of text as the average of each sentence's succedence.

$$F_{succ}(A, B) = \frac{1}{q} \sum_{y \in B} \mathrm{succ}(y) \tag{7}$$

Therein; $\mathrm{succ}(y)$ is the succedence of sentence y and $F_{succ}(A, B)$ is the succedence of block B appearing after block A.

2.5 Training

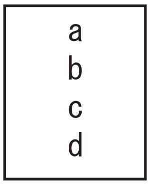

Figure 4: Generating training examples

We partition each human-ordered summary into continuous blocks. For example, in the case illustrated in figure 4, the summary consists of four sentences a,b,c and d. We then generate the following six pairs of blocks from this summary by excluding sentences from the end of the summary; $a|bcd, ab|cd, abc|d, a|bc, ab|c, a|b$. Usually, a summary is read from top to bottom and sentences appearing at the top of the summary are more important than the ones that appear to the end of the summary. Therefore, we decided to retain the top sentences in our blocks and remove from the end. For a summary of length n this partition process generates $(n-1)(n-2)/2 (= \mathrm{O}(n^2))$ number of pairs of blocks. Although there are other partitioning methods to generate blocks, one must consider the complexity issues when learning with lots of training data. Moreover, the paritioning method we propose yields continuous blocks in pairs and such blocks are more reliable than discontinous blocks. We create training data vectors with label +1 for the case where the two blocks are ordered as in the human-ordered summary and label -1 for its reverse. In our example, we have the following two cases for the pair of blocks $ab|cd$.

$$[F_{chro}(ab, cd), F_{topic}(ab, cd), F_{pre}(ab, cd), F_{succ}(ab, cd), +1]$$
$$[F_{chro}(cd, ab), F_{topic}(cd, ab), F_{pre}(cd, ab), F_{succ}(cd, ab), -1]$$

We train a two-class linear-kernel SVM using these training vectors.

2.6 Clustering

Initially, we create a block per each extract sentence. Then we repeatedly merge two blocks at a time in a hierarchical manner until we are left with a single block of text as depicted in figure 5. At each step we find the pair of blocks which has the highest integrated fitness value. The integrated fitness value is the class probability returned by the trained SVM in section 2.5. Note that (A,B) and (B,A) have different integrated fitness values in our model. We use arrows to state this fact in figure 5.

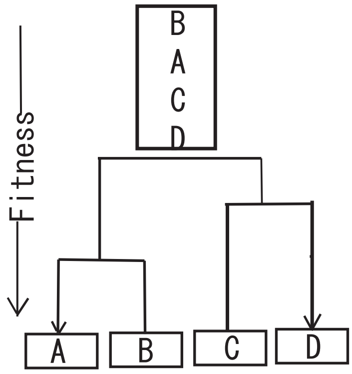

Figure 5: Clustering process

Table 1: Comparison with Human Ordering

	Sp	K	Cont	WK	AC
RO	-0.127	-0.069	0.127	0.025	0.011
PO	0.076	0.068	0.126	0.065	0.037
ChO	0.583	0.587	0.576	0.634	0.356
HeO	0.585	0.589	0.639	0.639	0.402
ClO	0.603	0.612	0.694	0.669	0.459
HO	1.000	1.000	1.000	1.000	1.000

3. Results

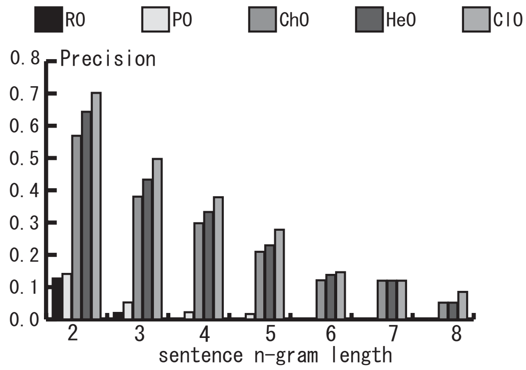

Figure 6: Precision vs sentence n-gram length

We used the 3rd Text Summarization Challenge (TSC) corpus for our experiments. TSC[*1] corpus contains news articles taken from two leading Japanese newspapers; Mainichi and Yomiuri. TSC-3 corpus contains extracted sentences for 30 summaries of different topics. However, in the TSC corpus the extracted sentences are not ordered to make a readable summary. Therefore, we first prepared 30 summaries by ordering the extraction data of TSC-3 corpus by hand. We then compared the orderings by the proposed algorithm against these human ordered summaries using

*1 http://lr-www.pi.titech.ac.jp/tsc/index-en.html five evaluation metrics: Spearman rank correlation coefficient (Sp), Kendall's coefficient (K), Continuity (C) [9], Weighted Kendall's coefficient (WK) [2] and Average Continuity (AC) [2]. Weighted Kendall Coefficient is an exponentially weighted version of the discordants in Kendall's coefficient. Average Continuity is defined by equation 9. It is the logarithmic average of the sentence continuity precision, P_n defined by equation 8, where N is the summary length and n is the length of sentence n-gram.

$$P_n = \frac{\text{number of matched n-grams}}{N - n + 1}. \quad (8)$$

$$\text{Average Continuity} = \exp\left(\frac{1}{3}\sum_{n=2}^{4}\log(P_n)\right) \quad (9)$$

We compared the proposed clustering order algorithm (ClO) with Random Order (RO), Probabilistic Order (PO) [4], Chronology Order (ChO), Hedge learning based Order (HeO) [2] and Human-made Order (HO). Results from our experiments are shown in table 1 and figure 6. ANOVA test shows a statistically significant difference among the methods compared in table 1. Moreover, student t-tests performed between chronological order and clustering order showed a statistically significant improvement when compared by Average Continuity and Continuity metric [9]. However, we could not find any statistical significance when compared by the other evaluation metrics. According to figure 6, clustering order has the highest precision values among all the methods for all lengths of sentence continuities.

4. Conclusion

We proposed a clustering based approach to sentence ordering for multidocument summarization. Our experimental results showed a satisfactory improvement over the existing methods for sentence ordering. However, we need to perform a human evaluation on the readability of our ordered summaries. Ordering the sentences is only the first step towards creating a readable summary. We need to perform amendments to sentences and smoothen the transition from a sentence to the next. Moreover, our algorithm does not take into account the cohesion relations suggested by Halliday and Hasan [6]. In our future work, we plan to extend the algorithm to cover these issues.

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

[2] Danushka Bollegala, Naoaki Okazaki, and Mitsuru Ishizuka. A machine learning approach to sentence ordering for multidocument summarization and its evaluation. *Proceedings of the Natural Language Processiong society of Japan*, pages 000–000, 2005.

[3] W. W. Cohen, R. E. Schapire, and Y. Singer. Learning to order things. *Journal of Artificial Intelligence Research*, 10:243–270, 1999.

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

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

[6] M.A.K.Halliday and Ruqaiya Hasan. *Cohesion in English.* Longman, 1976.

[7] Daniel Marcu. 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, 1997.

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

[9] Naoaki Okazaki, Yutaka Matsuo, and Mitsuru Ishizuka. An integrated summarization system with sentence ordering using precedence relation. *ACM-TALIP, to appear in 2005.*, 2005.

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