IMPROVING COHERENCE IN MULTI-DOCUMENT SUMMARIZATION THROUGH PROPER ORDERING OF SENTENCES

Danushka Tarupathi Bollegala

Master of Science (Information & Communication Engineering)
Graduate School of Information Science and Technology,
The University of Tokyo,
7-3-1, Hongo, Bunkyo-ku,
Tokyo, Japan. 113-8656.

ABSTRACT

The problem of extracting salient information to include in a summary has been researched extensively in the field of automatic text summarization. However, coherent arrangement of the extracted information has received little attention. Specially, in the case of extractive multi-document text summarization, sentences that convey important information are selected from a set of documents. There is no guarantee that this set of extracted sentences will form a coherent summary by itself. The order of presentation of information is an important factor that affects the coherence of a summary. This thesis focuses on the problem of automatically generating a coherent summary from a given set of documents by ordering the extracted sentences. I propose two different approaches to this problem: a pair-wise sentence comparison approach and a bottom-up text structuring approach. The pair-wise sentence comparison approach first compares all possible pairs of sentences and decides partial orderings between the two sentences in pairs. It then creates a total ordering that optimizes a certain function. In the bottom-up text structuring approach, I define four criteria for sentence ordering: *chronology*, *topical-closeness*, *precedence* and *succedence*. I then use support vector machines to integrate these four different criteria to compute the strength of association between two sentences. For training I use a set of manually ordered summaries. A hierarchical text clustering algorithm is used to produce a total ordering of sentences. I begin by ordering the pair of sentences that has the highest strength of association. I then repeatedly order the two segments of texts with the maximum association strength until a single segment with all sentences ordered is formed. I compare the sentence orderings produced by the proposed algorithm against manually ordered summaries using various rank correlation measures. Moreover, I perform a subjective grading of the generated summaries. Both automatic evaluation and subjective grading suggest that the proposed sentence ordering algorithms significantly outperforms all existing sentence ordering methods for multi-document summarization. Moreover, I investigate the problem of automatically evaluating a sentence ordering for its coherence and propose *Average Continuity* as an automatic evaluation measure for this task. The proposed automatic evaluation measure reports a high correlation with human ratings.

Dedication

To my dearest parents, for everything they have done.

LIST OF PUBLICATIONS

International Journal Papers

1. Danushka Bollegala, Yutaka Matsuo, Mitsuru Ishizuka. Measuring Sematic Similarity between Words using Web Search Engines. *ACM Transactions on the Web*, submitted, 2007.

International Conference Papers

1. Danushka Bollegala, Yutaka Matsuo, Mitsuru Ishizuka. Measuring Semantic Similarity between Words Using Web Search Engines. To appear in proceedings of 17 th International World Wide Web Conference (WWW 2007), Banff Alberta, Canada, 2007 May.

2. Danushka Bollegala, Yutaka Matsuo, Mitsuru Ishizuka. An Integrated Approach to Measuring Semantic Similarity between Words using Information available on the Web. To appear in proceedings of the *North American Chapter of Association for Computational Linguistics (NAACL)*, Rochester, U.S.A., 2007 April.

3. Danushka Bollegala, Yutaka Matsuo, Mitsuru Ishizuka. Identifying People on the Web through Automatically Extracted Key Phrases. In Proceedings of *TextLink Workshop at International Joint Conference on Artificial Intelligence (IJCAI)*, Hyderabad, India, 2007.

4. Danushka Bollegala, Yutaka Matsuo, Mitsuru Ishizuka. Disambiguating Personal Names on the Web using Automatically Extracted Key Phrases. In Proceedings of*European Conf. on Artificial Intelligence (ECAI)*, pp.553-557, Riva, Italy, 2006.

5. Danushka Bollegala, Yutaka Matsuo, Mitsuru Ishizuka. Extracting Key Phrases to Disambiguate Personal Name Queries in Web Search. In Proceedings of the *Workshop ”How can Computational Linguistics improve Information Retreival?”, at the joint 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for*

Computational Linguistics (COLING-ACL 2006), pp.17-24 , Sydney, Australia, 2006.

6. Danushka Bollegala, Naoaki Okazaki, Mitsuru Ishizuka. A bottom-up approach to Sentence Ordering for Multi-document Summarization. In Proceedings of the *Joint 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL 2006)* ,pp. 385-392, Sydney Australia. 2006.

7. Yutaka Matsuo, Masahiro Hamasaki, Hideaki Takeda, Junichiro Mori, Danushka Bollegala, Hiroyuki Nakamura, Takuichi Nishimura, Koiti Hashida and Mitsuru Ishizuka. Spinning Multiple Social Networks for Semantic Web. In Proceedings of the *21st National Conference on Artificial Intelligence (AAAI 2006),* pp 1381-1387, Boston, MA, USA,2006.

8. Danushka Bollegala, Yutaka Matsuo and Mitsuru Ishizuka. Extracting Key Phrases to Disambiguate Personal Names on the Web. In Proceedings of the *7th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing 2006)*, pp. 223-234, Mexico City, Mexico, 2006.

9. Danushka Bollegala, Naoaki Okazaki and Mitsuru Ishizuka. A Machine Learning Approach to Sentence Ordering for Multi-document Summarization and its Evaluation. In Proceedings of the *2nd International Joint Conference on Natural Language Processing (IJCNLP 2005),* pp. 624-635, Jeju, South Korea, 2005.

Domestic Conference Papers

1. Danushka Bollegala, Naoaki Okazaki and Mitsuru Ishizuka. Agglomerative Clustering Based Approach to Sentence Ordering for Multi-document Summarization. In Proceedings of *Natural Language Understanding and Models of Communication (NLC) The Institute of Electronics, Information and Communication Engineers (IEICE)*, pp 13-18, Biwako, Japan, 2006.

2. Danushka Bollegala, Yutaka Matsuo and Mitsuru Ishizuka. Disambiguating Personal Names on Web. In Proceedings of *Annual Meeting of the Japanese Society of Artificial Intelligence (JSAI),* Tokyo, Japan, 2006.

3. Danushka Bollegala, Naoaki Okazaki and Mitsuru Ishizuka. Agglomerative Clustering Based Approach to Sentence Ordering for Multi-document Summarization. In Proceedings of *Annual Meeting of the Japanese Society of Artificial Intelligence (JSAI)*, Kokura, Japan, 2005.

4. Danushka Bollegala, Naoaki Okazaki and Mitsuru Ishizuka. A Machine Learning Approach to Sentence Ordering for Multi-document Summarization. In Proceedings of *Annual Meeting of the Natural Language Processing Society of Japan (NLP)*, pp. 636-639, Takamatsu, Japan, 2005.

5. Danushka Bollegala, Naoaki Okazaki and Mitsuru Ishizuka. A Machine Learning Approach to Sentence Ordering for Multi-document Summarization. In Proceedings of the *Annual Meeting of the Information Processing Society of Japan (IPSJ)*, Tokyo, Japan, 2005.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

We live in an era of information overload. Despite the growing popularity of audio and visual media, the majority of information is still available in textual format. May it be reading news, e-mails or technical papers, We are forced to read huge volumes of text in our day-to-day activities. Advancements of information retrieval such as Web search engines have provided us with efficient means to search and gather the information we need. However, still the information collected and presented by search engines are so vast that no human has the time or labour to go through the complete result set. Therefore, automatically creating a brief summary from a given set of documents is an important task. A short summary of a set of documents helps a user to quickly understand the important information discussed in the documents. If the user needs more information then she can read the source documents.

Automatic text summarization has received attention from both academic and commercial worlds as a possible solution to the problem of information overload. Although the research on automatic text summarization dates back to Luhn's [23] work on abstract generation, this field has received much attention lately due to the vast textual information available on the Internet. Document Understanding Conference (DUC) [1] is a dedicated conference which explores various topics related to automatic text summarization such as multidocument summarization, topic focused summarization, question answering, etc. In Japan, NTCIR (NII Test Collection for IR Systems) [2] project has played a leading role in the field of Japanese text summarization. Columbia Newsblaster [30] is a multidocument news summarization system that crawls the web for news, classifies news according to their topics and creates a summary with sentence fusion. *GoogleNews* [3] is an automatic news aggregating system that operates on the Web. It clusters news articles related to a news topic and users can search for the news they are interested. However, presently it does not create summaries form the clustered news articles.

[1] http://duc.nist.gov/

[2] http://research.nii.ac.jp/ntcir/

[3] http://news.google.com/

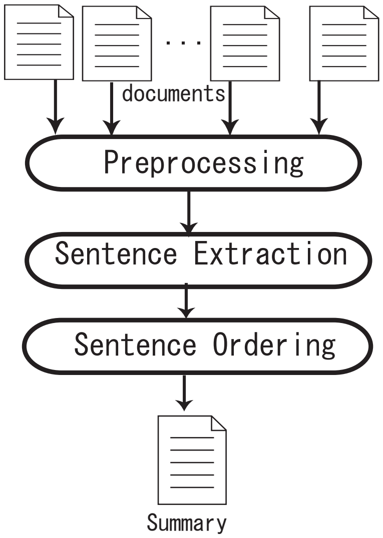


Fig. 1.1: A typical multi-document summarization system

1.1 Multi-document Summarization

Summarization can be categorized into extractive summarization and abstractive summarization according to the approach used to convey the information to the user. In extractive summarization information is selected from the source documents and presented in the summary with minimum modifications. Most extractive summarization systems select whole sentences from the source documents and include them in the summary. On the other hand, abstractive summarization attempts to understand the concepts discussed in the documents and generate natural language texts as a summary. Abstractive summarization is a difficult problem because it requires deeper analysis of source documents and concept-to-text generation. Currently most of the commercially available automatic text summarization system are extractive summarization systems.

Automatic text summarization can be further categorized into single document summarization and multi-document summarization. In single document summarization a summary is created from a single document. Whereas in multi-document summarization a single summary is created from a set of multiple documents. Multi-document summarization is a more challenging problem because one needs to recognize the relation between documents in order to create a coherent summary from a set of multiple documents. Figure 1.1 depicts a typical multi-document summarization system. First the set of source documents is preprocessed using tokenizers and part-of-speech taggers. In the case of Japanese language, words are not separated by spaces. Therefore, accurate tokenization of text is essential to further process the documents. In the case of news texts, most news providers annotate the articles by including meta information such as publication date and time, keywords that can be used to index the news article, genre information (whether it is news on politics, science, economics, sports etc). These meta information are useful when classifying, indexing and summarizing the articles. As we will see in chapter 2, particularly the time stamps (publication date and time information provided by the authors of the articles) are very useful when deciding the order among sentences in a summary. Preprocessing stage will extract such useful information from the documents. The second stage shown in Figure 1.1 represents sentence extraction in multi-document summarization. Using the information extracted in the previous stage, a set of sentences is selected from the source documents in order to include in the summary. Various algorithms have been proposed to identify salient information from the source documents [24]. Some of these methods first identify important words or phrases from the source documents using term-weighting methods such as TF-IDF [8] and then extract sentences that contain these words or phrases. However, due to the length of a summary, one cannot select all the sentences that contain such salient words. Moreover, if two or more sentences convey the same information then in order to avoid duplication of information in a summary it is desirable to select just one of them and include in the summary. Selecting a set of sentences that maximizes the information content is a central problem in extractive multi-document summarization. Even with single document summarization the same argument holds. However, when creating a summary with one document, the source document itself has less repetition of information. Whereas with a set of documents on the same event (for example a set of articles published by different authors on the same news) we can expect to find sentences from different articles that convey the same information. Once we have identified the important sentences to include in the summary, we need to produce a coherent text from the selected sentences. The final stage in Figure 1.1 represents this post-processing step. The main focus of this thesis is this final stage of multi-document summarization. In this thesis, I assume that we have already identified a set of sentences to be included in the summary and concentrate on the problem of generating a coherent summary from it.

1.2 The problem of Sentence Ordering

This thesis concentrates on the problem of improving coherence in a multidocument extractive summary with special interest in information ordering. Multi-document summarization (MDS) [38, 4, 6] tackles the information overload problem by providing a condensed version of a set of documents. Among a number of sub-tasks involved in MDS, e.g., 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 (e.g., 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 [1] has provided empirical evidence that proper order of extracted sentences improves their readability significantly. Lapata [31] shows experimentally that the time taken to read a summary strongly correlates with the arrangement of sentences in the summary.

1. Such storms have maximum sustained winds greater than 155 mph and can cause catastrophic damage.

2. Earlier Wednesday Gilbert was classified as a Category 5 storm, the strongest ad deadliest type of hurricane.

3. Tropical Storm Gilbert formed in the eastern Caribbean and strengthened into a hurricane Saturday night.

Fig. 1.2: Randomly ordered sentences in a summary

For example consider the three sentences shown in Figure 1.2. [4] First and second sentences are extracted from the same source document whereas the third sentence is extracted from a different document. Although, all three sentences are informative and talks about the storm *Gilbert* ordering shown in Figure 1.2 is confusing. The phrase *such storms* in sentence 1 refers to *category* 5 *storms* described in sentence 2. A better arrangement of sentences in this example would be 3-2-1.

In single document summarization, one possible ordering of extracted information is provided by the input document itself. However, ordering a set of sentences extracted from a set of documents into a coherent text is a non-trivial task. For example, identifying rhetorical relations [25] in an ordered text has been a difficult task for computers, whereas this 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 ex-

[4] These sentences were selected from a reference summary in 2003 Document Understanding Conference (DUC) dataset.

pect that any random ordering of a set of extracted sentences from such diverse documents to be coherent.

Several strategies to determine sentence ordering have been proposed as described in section 1.3. Several of these methods utilize chronological information such as publication date and time of documents, whereas others concentrate on grouping information by topics and computing the likelihood of a particular topic preceding or succeeding another topic. However, the appropriate way to combine these strategies to achieve more coherent summaries remains unsolved. In this paper, I explore 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. A bottom-up approach to sentence ordering is proposed. The proposed algorithm repeatedly concatenates textual segments until a segment with all sentences arranged, is obtained. Furthermore, I investigate numerous automatic evaluation measures for the task of sentence ordering for multi-document summarization.

Besides the proper order of sentences, there are other factors that contribute to the readability of a summary such as pronoun resolution [39, 15, 9] and acronym replacement [32, 33]. In order to illustrate the problem of pronoun resolution in the context of multi-document summarization consider the two sentences in Figure 1.3.

1. Iraqi Prime Minister Nuri al-Maliki is telling the United States and Iran to keep their fight out of Iraq.

2. He said he believes Iran is targeting U.S. forces in Iraq.

Fig. 1.3: Pronoun resolution in MDS

Two consecutive sentences extracted from a news article are shown in Figure 1.3. Assume that a sentence extraction algorithm only selected the second sentence and ignored the rest of the sentences from this article. Then the pronoun *he* at the beginning of this sentence becomes ambiguous. Depending on the sentence we bring before this sentence in the summary, sentence 2 can be mis-interpreted in many different ways. In order to improve the readability in the summary, the post-processing stage should replace the pronouns that appear in the set of extracted sentences.

Similarly, when there are acronyms (e.g., NLP for Natural Language Processing) in the extracted sentences, replacing the acronyms by their full-forms improves readability of the summary. Acronyms can be replaced using pre-compiled acronym dictionaries. If it a rare acronym which is not listed in standard acronym dictionaries then it is likely that the author of a news paper article gives the full form of an acronym at the first instance she uses it. Pronoun resolution and acronym extraction are beyond the scope of this thesis.

This thesis is organized as follows. I first describe the previous work in the field of sentence ordering for multi-document summarization in section 1.3. I then present a bottom-up approach to sentence ordering in chapter 2 followed by a pairwise sentence comparison approach in chapter 4. I discuss automatic evaluation measures for this task in chapter 2. The thesis concludes with a discussion on possible future work in this field in chapter 5.

1.3 Previous Work in Sentence Ordering for MDS

Existing methods for sentence ordering can be classified into two approaches: making use of chronological information [29, 22, 1, 34]; and learning the natural order of sentences from large corpora not necessarily based on chronological information [21, 3, 14].

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 multi-document summarization [22, 29]. In this approach, first sentences that are extracted from the same source document are ordered according to the order they appear in the original document. Then these sentence segments are ordered chronologically, bringing in the earliest segment (sentences that were extracted from the earliest published article) to the beginning of the summary in an ascending order of chronology. One problem frequently acknowledged with this approach is that it is unable to order sentences which were extracted from documents which were published on the same date (and time). Moreover, it ignores the events (or themes) discussed in the input documents.

Barzilay et al. [1] 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. They first use a linear text segmentation algorithm [16] based on word distribution and coreference analysis to segment each input document. They define *theme* as the set of sentences conveying similar information from different input texts. For two themes $\{A_1, \ldots, A_n\}$ and $\{B_1, \ldots, B_m\}$, they denote $\#AB$ to be the number of pairs of sentences (A_i, B_j) which appear in the same text, and $\#AB^+$ to be the number of sentence pairs which appear in the same text and are in the same segment. They then compute the ratio $\#AB^+/\#AB$ to measure the relatedness of two themes. The intuition behind this computation is that if this ratio is higher (i.e., more sentence pairs from the two themes fall in the same text segment) for two themes then they are topically related. To cluster sentences into themes they use a sentence distance measure.

Okazaki et al. [34] 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. Their experimental results show that the proposed algorithm improves the chronological ordering significantly.

Lapata [21] 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). Lapata [31] also proposed the use of Kendall's rank correlation coefficient for an automatic evaluation that quantifies the differences between orderings produced by the proposed method and a human. 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. (see section 4.1.2 for further details on this approach)

Barzilay and Lee [3] 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 (e.g., 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.

Ji and Pulman [14] proposed a sentence ordering algorithm using a semi-supervised sentence classification and historical ordering strategy. They build a network of sentences that appear in a summary. Nodes in this network represent sentences in a summary and the weights on edges are seen as transition probabilities. They then semi-supervisedly classify the sentences in the source documents into the nodes in the network. Finally, they order summary sentences according to the original positions of their partners in the same class. However, they do not compare their results against chronological ordering of sentences, which has been shown to be an effective sentence ordering strategy in multi-document news summaries.

As described above, several good strategies/heuristics to deal with the sentence ordering problem have been proposed. In order to integrate multiple strategies/heuristics, I have formalized them in a machine learning framework and have considered an algorithm to arrange sentences using the integrated strategy. In chapter 2 I describe this approach.

2. BOTTOM-UP APPROACH TO SENTENCE ORDERING IN MDS

2.1 Method

I define notation $a \succ b$ to represent that sentence a precedes sentence b. I 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, I denote as:

$$A = (a_1 \succ a_2 \succ ... \succ a_m). \tag{2.1}$$

The two segments A and B can be ordered either B after A or A after B. I 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, I 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 [27]. Assuming that the semantic units (sentences) and their rhetorical relations [25] (e.g., 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 task for computers [28]. 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, I introduce a function $f(A \succ B)$ to represent the direction and strength of the association of two segments A and B,

$$f(A \succ B) = \begin{cases} p & \text{(if } A \text{ precedes } B\text{)} \\ 0 & \text{(if } B \text{ precedes } A\text{)} \end{cases}, \tag{2.2}$$

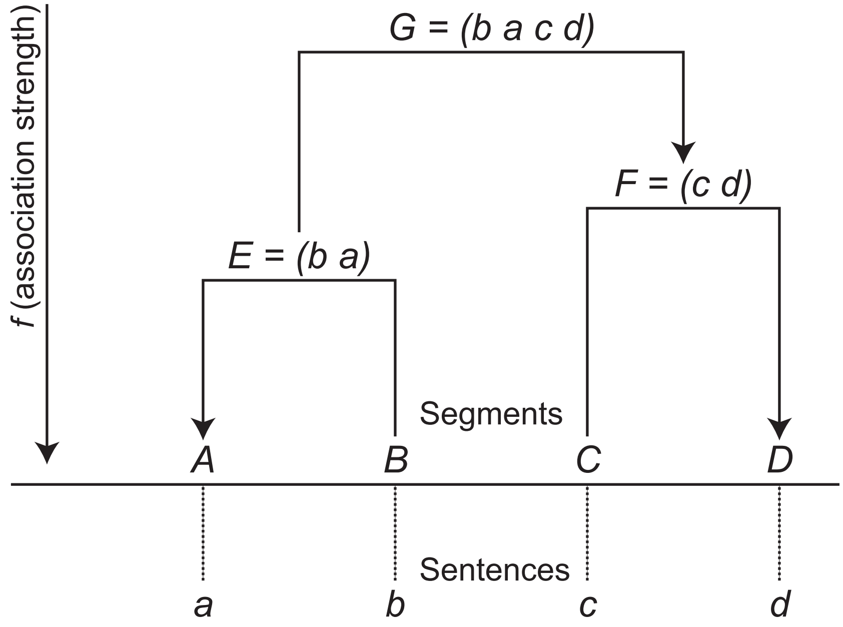


Fig. 2.1: Arranging four sentences A, B, C, and D with a bottom-up approach.

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, e.g., $f(A \succ B)$ and $f(B \succ A)$, are not always identical in this definition,

$$f(A \succ B) \neq f(B \succ A). \tag{2.3}$$

Figure 2.1 shows the process of arranging four sentences a, b, c, and d. Firstly, I initialize four segments with a sentence for each,

$$A = (a), B = (b), C = (c), D = (d). \tag{2.4}$$

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

$$E = (b \succ a). \tag{2.5}$$

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

$$F = (c \succ d). \tag{2.6}$$

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

$$G = (b \succ a \succ c \succ d). \tag{2.7}$$

In the above description, I have not defined the association of the two segments. The previous work described in section 1.3 has addressed the association of textual segments (sentences) to obtain coherent orderings. I 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) [40] classifier.

2.1.1 Chronology criterion

Chronology criterion reflects the chronological ordering [22, 29], which arranges sentences in a chronological order of the publication date. I define the association strength of arranging segments B after A measured by a chronology criterion $f_{\mathrm{chro}}(A \succ B)$ in the following formula,

$$
\begin{aligned}
& f_{\mathrm{chro}}(A \succ B) \\
= & \begin{cases} 1 & \mathrm{T}(a_m) < \mathrm{T}(b_1) \\ 1 & [\mathrm{D}(a_m) = \mathrm{D}(b_1)] \wedge [\mathrm{N}(a_m) < \mathrm{N}(b_1)] \\ 0.5 & [\mathrm{T}(a_m) = \mathrm{T}(b_1)] \wedge [\mathrm{D}(a_m) \neq \mathrm{D}(b_1)] \\ 0 & \text{otherwise} \end{cases} .
\end{aligned}
\tag{2.8}
$$

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.

Chronological ordering of sentences has shown to be particularly effective in multi-document news summarization. As already discussed in section 1.3 several previous research in this topic have proposed sentence ordering algorithms using chronological information. Usually, news publishers provide time-stamps for their news articles. This information can be used to decide the chronological order among sentences extracted from different documents. However, when there is no time stamp assigned to the document or where there exist several documents with identical time stamps, chronological ordering of sentences becomes impossible. Inferring temporal relations among events in multi-document summarization using implicit time references (such as tense system) and explicit time references (such as temporal adverbials) [7] is an interesting alternative for using time-stamps.

2.1.2 Topical-closeness criterion

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

$$f_{\mathrm{topic}}(A \succ B) = \frac{1}{|B|} \sum_{b \in B} \max_{a \in A} \mathrm{sim}(a, b). \tag{2.9}$$

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

2.1.3 Precedence criterion

Let us think of the case where I 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 (e.g., the sentences appearing before sentence b) as segment A. This criterion is a formalization of the sentence-ordering algorithm proposed by Okazaki et al, [34].

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

I define the precedence criterion in the following formula,

$$f_{\mathrm{pre}}(A \succ B) = \frac{1}{|B|} \sum_{b \in B} \max_{a \in A, p \in P_b} \mathrm{sim}(a, p). \tag{2.10}$$

Here, P_b is a set of sentences appearing before sentence b 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 2.2 shows an example of calculating the precedence criterion for arranging segment B after A. I 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 2.10 is interpreted as the average similarity of the precedent sentences $\forall P_b (b \in B)$ to the segment A.

2.1.4 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 after A:

$$f_{\mathrm{succ}}(A \succ B) = \frac{1}{|A|} \sum_{a \in A} \max_{s \in S_a, b \in B} \mathrm{sim}(s, b). \tag{2.11}$$

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 2.1.4 shows an example of calculating the succession criterion to arrange segments B after A. I approximate the information that should follow segment A by the sentences in segments S_a. I then compare each segments S_a with segment B to measure how much segment B conveys this information. The succession criterion measures the substitutability of the succedent information (e.g., the sentences appearing after the sentence $a \in A$) as segment B.

2.1.5 SVM classifier to assess the integrated criterion

I 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 2.2). 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 $f_{\mathrm{succ}}(A \succ B)$. I formalize the integration task as a binary classification problem and employ a Support Vector Machine (SVM) as the classifier. I conducted a supervised learning as follows.

I 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 2.4. Firstly, I 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, I 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 $\{(a \succ b \succ c), (d)\}$), I 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, I generate one positive and one negative training instance as follows.

Given a pair of two segments A and B arranged in an order $A \succ B$, I calculate four values, $f_{\text{chro}}(A \succ B)$, $f_{\text{topic}}(A \succ B)$, $f_{\text{pre}}(A \succ B)$, and $f_{\text{succ}}(A \succ B)$ to obtain the instance with the four-dimensional vector (Figure 2.5). I label the instance (corresponding to $A \succ B$) as a positive class (ie, $+1$). Simultaneously, I obtain another instance with a four-dimensional vector corresponding to $B \succ A$. I label it as a negative class (i.e., -1). Accumulating these instances as training data, I obtain a binary classifier by using a Support Vector Machine with a quadratic kernel. The SVM classifier yields the association direction of two segments (e.g., $A \succ B$ or $B \succ A$) with the class information (i.e., $+1$ or -1).

I assign the association strength of two segments by using the posterior probability that the instance belongs to a positive ($+1$) class. When an instance is classified into a negative (-1) class, I set the association strength as zero (see the definition of Formula 2.2).

2.2 Posterior probabilities from SVMs

Being a large-margin classifier, the output of an SVM is the distance from the decision hyper-plane. However, this is not a calibrated posterior probability. I use sigmoid functions to convert this uncalibrated distance into a calibrated posterior probability. In this section I explain the different methods proposed for converting SVM outputs to posterior probabilities and the method I chose [37].

Support vector machines classify instances based on the distance to the instance from the decision hyperplane. For an instance x, the distance for this instance from the hyperplane is give by,

$$f(x) = h(x) + b. \tag{2.12}$$

Here,

$$h(x) = \sum_i y_i \alpha_i k(x_i, x) \tag{2.13}$$

and b is the bias. $h(x)$ lies in a Reproducing Hilbert Space (RKHS) induced by a kernel k.

Training an SVM minimizes the misclassification error. Various error functions have been proposed for the training of SVMs. A popular formulation of the error function is as follows,

$$C \sum_i (1 - y_i f_i) + \frac{1}{2} ||h||, \tag{2.14}$$

where $f_i = f(x_i)$. This error function has the following properties. First, minimizing the error function will also minimize a bound on the test misclassification rate [40]. Secondly, minimizing this error function will produce a sparse machine where only a subset of possible kernels are used in the final machine. This sparsity property is a well-known desirable feature of SVMs that makes it such a powerful learning algorithm in large feature spaces such as all the words from a vocabulary and their n-grams considered in most text classification tasks. Intuitively, only the training instances that lie on the hyper plane (i.e., only the training instances which are support vectors) decide the classification result. This is usually a smaller subset of all the points in the kernel space. When converting the SVM outputs to posterior probabilities a technique which retains the sparsity property of SVMs is desirable.

One method of producing probabilistic outputs from a kernel machine (not limited to support vector machines) is by logistic link function [41],

$$\mathrm{P(class|input)} = P(y = 1|x) = p(x) = \frac{1}{1 + \exp(-f(x))}, \tag{2.15}$$

where f is defined as in Equation 2.12.

In this formulation a negative log multinomial likelihood plus a term that penalizes the norm of the Hilbert space induced by the kernel is minimized:

$$-\frac{1}{m} \sum_i (\frac{y_i + 1}{2} \log(p_i) + \frac{1 - y_i}{2} \log(1 - p_i)) + \lambda ||h||^2, \tag{2.16}$$

where $p_i = p(x_i)$. One problem with the above formulation of posterior probability using SVM outputs is that the sparseness property might no longer hold. Intuitively, the above method attempts to define a probabilistic function which maximizes likelihood of observing the training data. There have been various probability functions defined for this task including cosine fit by Vapnik himself and Gaussian fit by Hasite and Tibshirani [12]. Platt [37] propose a modification to Equation 2.15 as follows,

$$P(y = 1|f) = \frac{1}{1 + \exp(Af + B)}, \tag{2.17}$$

where A and B are two parameters fit in the model using maximum likelihood estimation from a training set (f_i, y_i). Platt's experimental results shows a very good agreement between the probability estimates by the fitted sigmoid and actual data.

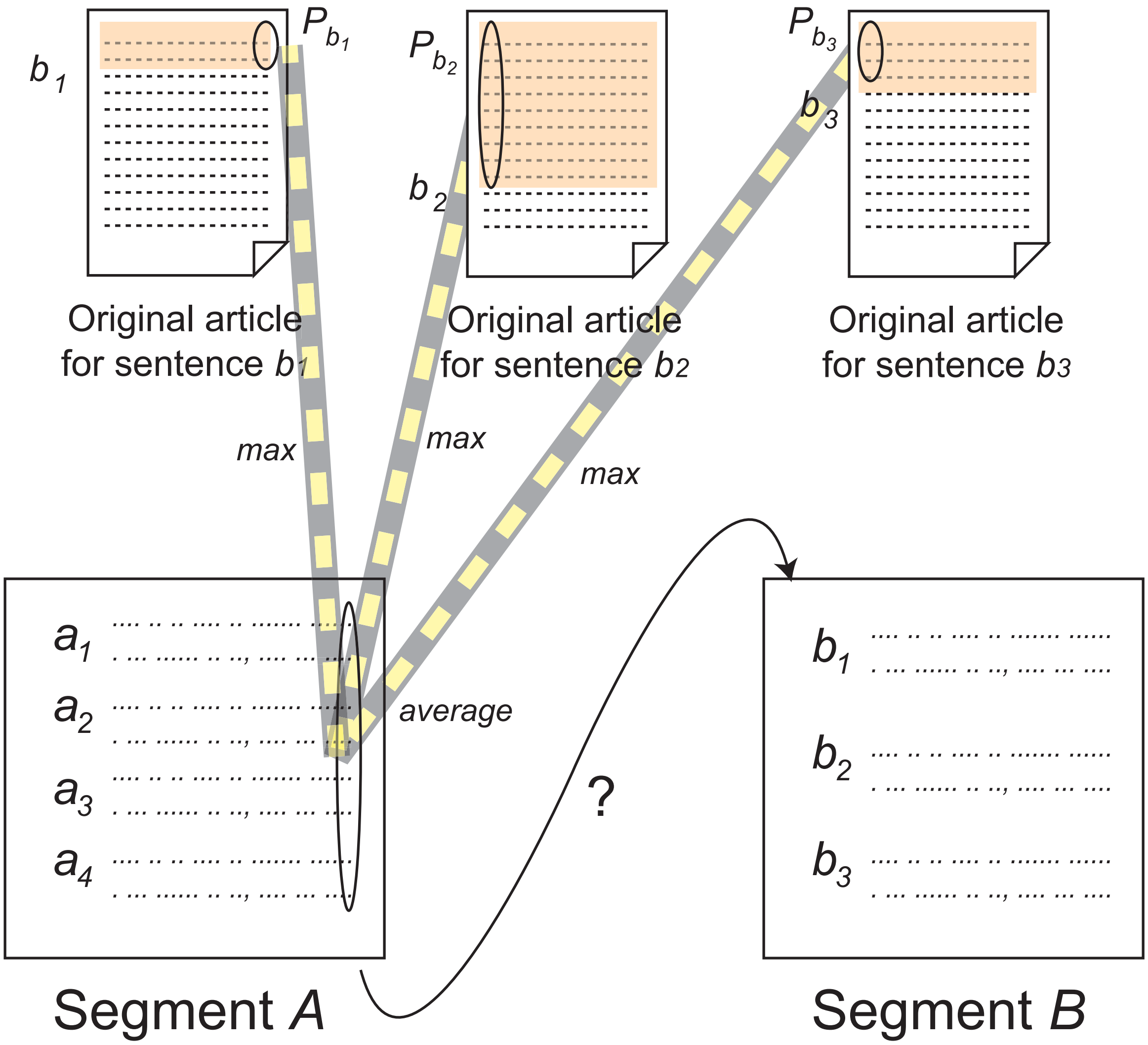


Fig. 2.2: Precedence criterion

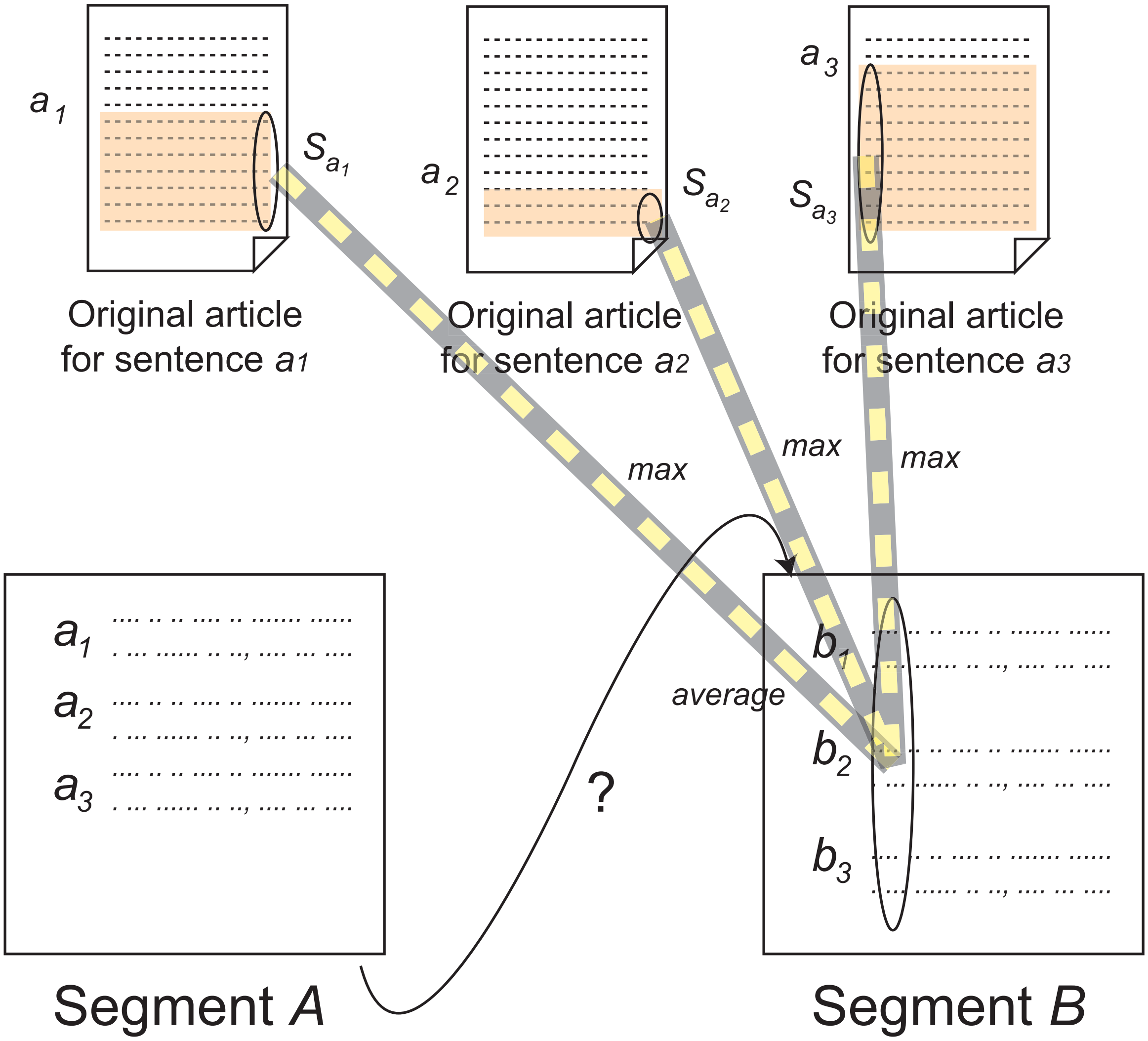

a 1
S$_{a_1}$
Original article for sentence a1
a 2
S$_{a_2}$
Original article for sentence a2
a 3
S$_{a_3}$
Original article for sentence a3
max
max
max
average
?
a1
a2
a3
b1
b2
b3
Segment A
Segment B

Fig. 2.3: Succession criterion

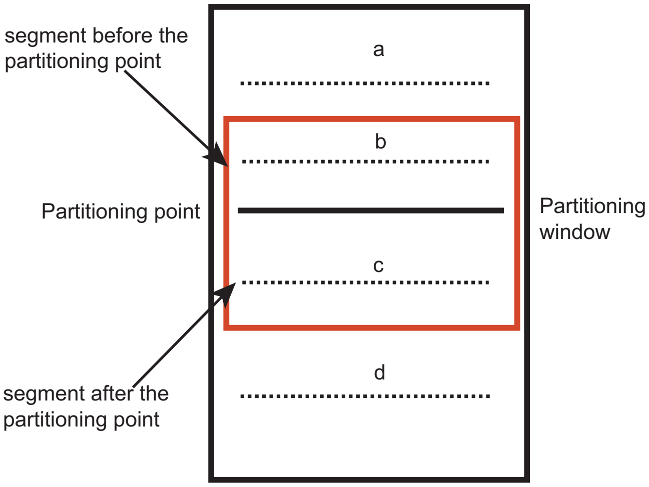

segment before the partitioning point
a
b
Partitioning point
Partitioning window
c
d
segment after the partitioning point

Fig. 2.4: Partitioning a human-ordered extract into pairs of segments

$$
\begin{aligned}
+1 &: [f_{\text{chro}}(A \succ B), f_{\text{topic}}(A \succ B), f_{\text{pre}}(A \succ B), f_{\text{succ}}(A \succ B)] \\
-1 &: [f_{\text{chro}}(B \succ A), f_{\text{topic}}(B \succ A), f_{\text{pre}}(B \succ A), f_{\text{succ}}(B \succ A)]
\end{aligned}
$$

Fig. 2.5: Two vectors in a training data generated from two ordered segments $A \succ B$

3. EVALUATION OF SENTENCE ORDERINGS

I evaluated the proposed method using the 3rd Text Summarization Challenge (TSC-3) corpus[1]. The TSC-3 corpus contains 30 sets of extracts, each of which consists of unordered sentences[2] extracted from Japanese newspaper articles relevant to a topic (query). I arrange the extracts by using different algorithms and evaluate the readability of the ordered extracts by a subjective grading and several automatic evaluation measures.

Tab. 3.1: 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

In order to construct training data applicable to the proposed method, I asked two human subjects to arrange the extracts and obtained $30(\text{topics}) \times 2(\text{humans}) = 60$ sets of ordered extracts. Table 3.1 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. This experiment suggests that there is a high correlation between humans for sentence orderings. However, the experiment needs to be carried out using a large number of subjects in order to obtain any statistically guaranteed results.

I 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, I generated a set of 30 extracts for evaluation. In addition to the proposed method, I prepared six sets

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

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

of sentence orderings produced by different algorithms for comparison. I 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 2.8. Sentences are arranged in chronological order of their publication date;

Topical-closeness ordering (TOP) arranges sentences with the topical-closeness criterion defined in Formula 2.9;

Precedence ordering (PRE) arranges sentences with the precedence criterion defined in Formula 2.10;

Suceedence ordering (SUC) arranges sentences with the succession criterion defined in Formula 2.11.

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 criterion independently and their contribution to solving the sentence ordering problem.

3.1 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 [1, 34, 31]. I asked two human judges to rate sentence orderings according to the following criteria.

Perfect A *perfect* summary is a text that we cannot improve any further by reordering.

Acceptable 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.

Poor A *poor* summary is one that loses the thread of the story at some places and requires minor amendments to bring it up to an acceptable level.

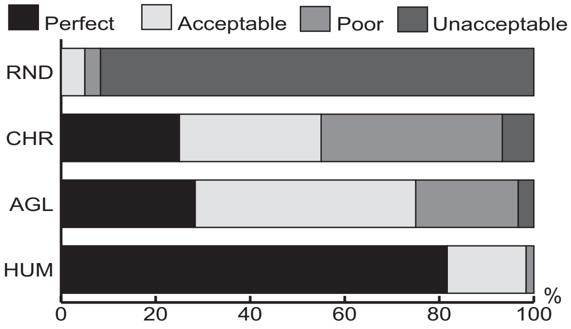


Fig. 3.1: Subjective grading

Unacceptable 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, I inform the judges that the summaries were made from a same set of extracted sentences and only the ordering of sentences is different. Furthermore, the judges were given access to the source documents for each summary.

Figure 3.1 shows the distribution of the subjective grading made by two judges to four sets of orderings, RND, CHR, AGL and HUM. Each set of orderings has 30(topics) $\times$ 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 criterion with other criteria 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 ($W = 0.939$).

3.2 Methods for semi-automatic evaluation

In general, subjective grading consumes much time and effort, even though we cannot reproduce the evaluation after wards. Automatic evaluation measures are particularly useful when evaluations must be performed quickly and frequently.

There have been several methods proposed [2, 17, 18] for the task of automatically evaluating coherence in a given text. Barzilay and Lapata [2] proposed a coherence model inspired by centering theory.

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

Fig. 3.2: An example of an ordering under evaluation T_{eval} and its reference T_{ref}.

However, in the current task summaries produced by different systems for a particular topic differ only in their orderings of sentences. I need a measure that only concentrates on the effect on coherence due to different sentence orderings. Although there are differences in applications, automatic evaluation methods usually compare system output against a set of gold standards and produce a numeric value indicating the similarity or closeness between the system output and the gold standards. Following this procedure, numerous automatic evaluation methods have been proposed in natural language processing tasks such as machine translation [36] and text summarization [13].

The previous studies [3, 21, 31] employ rank correlation coefficients such as Spearman's rank correlation and Kendall's rank correlation, assuming a sentence ordering to be a rank.

Let $S = s_1 \ldots s_N$ be a set of N items to be ranked. Let π and σ denote two distinct orderings of S. Then Kendall's rank correlation coefficient [20] (also known as Kendall's τ) is defined as follows,

$$\tau = 1 - \frac{2\mathrm{D}(\pi, \sigma)}{N(N-1)/2}. \tag{3.1}$$

Here, $\mathrm{D}(\pi, \sigma)$ denotes the discordant pairs of sentences in the two rankings. For example, in Figure 3.2 the four sentences pairs (e, a), (e, b), (e, c) and (e, d) appear in reverse-order in T_{eval}. These four discordant sentences pairs between T_{ref} and T_{eval} results in a Kendall's τ of 0.2. Kendall's τ is in the range $[-1, +1]$. It takes the value $+1$ if the two sets of orderings are identical and -1 if one is the reverse of the other.

Similarly, Spearman's rank correlation coefficient (r_s) between orderings π and σ is defined as follows,

$$r_s = 1 - \frac{6}{N(N+1)(N-1)} \sum_{i=1}^{N} (\pi(i) - \sigma(i))^2. \tag{3.2}$$

Spearman's rank correlation coefficient for the example shown in Figure 3.2 is 0. r_s lies in the range $[-1, +1]$. As with Kendall's τ, r_s value of $+1$ is obtained for two identical orderings. Whereas r_s computed between an ordering and its revers is -1.

Okazaki et al. [34] 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, I propose an *average continuity* metric, which extends the idea of the continuity metric to continuous k sentences.

3.2.1 Average Continuity

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. I define P_n to measure the precision of n continuous sentences in an ordering to be evaluated as,

$$P_n = \frac{m}{N - n + 1}. \tag{3.3}$$

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 3.2, the precision of 3 continuous sentences P_3 is calculated as:

$$P_3 = \frac{2}{5 - 3 + 1} = 0.67. \tag{3.4}$$

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

$$\text{AC} = \exp\left(\frac{1}{k-1}\sum_{n=2}^{k}\log(P_n + \alpha)\right). \tag{3.5}$$

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. I set $k = 4$ (i.e., more than five continuous sentences are not included for evaluation) and $\alpha = 0.01$. Average continuity is in range $[0, +1]$. It becomes 0 when evaluation and reference orderings share no continuous sentences and $+1$ when the two orderings are identical. In Figure 3.2, Average Continuity is calculated as 0.63. The underlying idea of Formula 3.5 was proposed by Papineni et al. [36] 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.

Tab. 3.2: Comparison with human-made ordering

Method	Spearman	Kendall	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

3.3 Results of semi-automatic evaluation

Table 3.2 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 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 ($p < 0.01$). I 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 I could not compare my experiment with the probabilistic approach [21] directly due to the difference of the text corpora, the Kendall coefficient reported higher agreement than Lapata's experiment Her experiments report a Kendall coefficient of 0.48 with lemmatized nouns and 0.56 with verb-noun dependencies.

Experimental results of learning with different kernels are shown in Table 3.3. I tested five popular kernel types: linear kernel (with-out using any kernel), polynomial kernel (quadratic), polynomial kernel (cubic), radial basis functions (RBF) kernel, sigmoid kernel. Among the different kernels tested, the best results are achieved with quadratic kernel. All three automatic evaluation measures used in Table 3.3 report the maximum value for quadratic kernel. Theoretically, higher degree kernels can capture complex non-linear dependencies between the features. However, in order to accurately learn in these complex spaces, one needs a lot of training data. We see a deterioration of performance when we move from quadratic kernel to cubic kernel. Considering the few number of examples (only 30 summaries and ca. 8000 training instances) this behavior is expected. Moreover, quadratic, radial basis functions and sigmoid kernels perform better than (or at the same level in the case of RBF and Sigmoid kernels) the linear kernel. This suggests that the exact combination of the four criteria I discussed in the thesis is non-linear.

Tab. 3.3: Performance vs SVM kernel type

Kernel Type	Spearman	Kendall	Average Continuity
Linear	0.524	0.507	0.294
Polynomial (degree=2)	0.529	0.511	0.311
Polynomial (degree=3)	0.513	0.499	0.311
Radial Basis Functions	0.524	0.507	0.294
Sigmoid Function	0.524	0.507	0.294

Tab. 3.4: Effect of removing a criterion

Criterion removed	Spearman	Kendall	Average Continuity
chronology	0.398	0.363	0.076
topical-closeness	0.532	0.517	0.295
precedence	0.520	0.502	0.311
succession	0.524	0.507	0.294

3.4 Effect of each criterion on coherence

I used four criteria in this thesis and integrated them using support vector machines to leverage the association strength between two text segments. However, it is important to know how each criterion contributes to the overall performance. In order to find out the influence that each criterion imparts on the performance I conducted the following experiment. I removed a criterion at both learning phase and ordering phase from the model and repeated this procedure for each criterion. I evaluate the sentence orderings produced by eliminating a criterion at a time by comparing them with human-ordered summaries using automatic evaluation measures. Results from our experiment are summarized in Table 3.4.

If removing a particular criterion from the model deteriorates the performance then that criterion can be considered as important. According to Table 3.4, removing the chronology criterion results in the poorest performance. This is to be expected as chronological information have shown to be very useful in deciding the order among extracted sentences in previous work of news summarization [1, 35]. Surprisingly, removing topical-closeness criterion improves both Spearman and Kendall rank correlation coefficient (compare the results with polynomial (degree=2) kernel in Table 3.3). However, removing the topical-closeness criterion reduces average continuity. Topical-closeness criterion was defined in 2.9 using only the extracted sentences without referring to the source documents. Therefore, the value of topical-closeness remains unchanged for all permutations

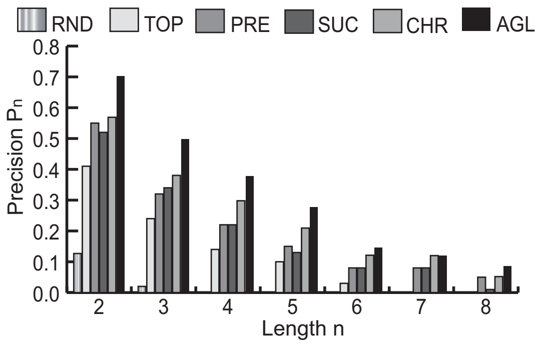


Fig. 3.3: Precision vs unit of measuring continuity.

of sentences in the two text segments. Therefore, we cannot select the best ordering among the all permutations inside a particular text segment using topical-closeness alone. However, by definition of Equation 2.9, topical-closeness criterion assigns a higher score for pairs of text segments in which the two segments contain a lot of overlapping words. This results in a better continuity as sentences which discusses on the same topic are ordered in close proximity.

Figure 3.3 shows precision P_n with different length values of continuous sentence n for the six methods compared in Table 3.2. The number of continuous sentences becomes sparse for a higher value of length n. Therefore, the precision values decrease as the length n increases. I considered geometric mean of individual precision instead of arithmetic mean in order to account for this phenomenon in Equation 3.5. Although RND ordering reported some continuous sentences for lower n values, no continuous sentences could be observed for higher n values. Four criteria described in Section 2.1 (i.e., CHR, TOP, PRE, SUC) produce segments of continuous sentences at all values of n. AGL ordering obtained the highest precision for any length n. CHR reports the second highest precision values for any length n.

3.5 Correlation between subjective gradings and semi-automatic evaluation measures

In section 3.2 I defined three evaluation measures: Kendall's τ, Spearman's rank correlation coefficient (r_s) and average continuity. Ideally, a semi-automatic evaluation measure should have a good agreement with subjective gradings. For this purpose, I measure the correlation between the elicited gradings in section 3.1 and the values reported by the semi-automatic measures described in section 3.2 as follows.

First, I order each set of extracted sentences using three different methods:

random ordering, chronological ordering and the proposed method. For the 30 topics in my dataset, this procedure yields 90 (30×3) summaries. I then compute Kendall's τ, Spearman's r_s and Average Continuity (AC) for each of the 90 summaries using two reference summaries for each topic. I take the average value between the two evaluations and consider it to be the automatically evaluated score of a summary. I then assign a grade based on the automatically evaluated score for a summary as in Table 3.5.

Tab. 3.5: Assigning grades based on semi-automatic evaluation scores

Grade	range of τ	range of r_s	range of AC
Unacceptable	$[-1, -0.5)$	$[-1, -0.5)$	$[0, 0.25)$
Poor	$[-0.5, 0)$	$[-0.5, 0)$	$[0.25, 0.5)$
Acceptable	$[0, 0.5)$	$[0, 0.5)$	$[0.5, 0.75)$
Perfect	$[0.5, 1.0]$	$[0.5, 1.0]$	$[0.75, 1.0]$

It is noteworthy that the three evaluation measures mentioned in Table 3.5 might have different distributions of scores. An equal partitioning as shown in Table 3.5 might not necessarily be the best way to separate the four gradings. A different partitioning scheme might yield a better correlation with subjective gradings. However, the equal partitioning scheme as shown in Table 3.5 is both simple and intuitive. For example, if we adopt a partitioning scheme that depends on the number of sentences in the summary, then it would be difficult to compare evaluations done on different datasets.

Tab. 3.6: Spearman coefficient vs Subjective grading

Grade	Spearman	Agree	Disagree	Ratio
Perfect	37	17	20	0.459
Acceptable	14	5	9	0.357
Poor	21	9	12	0.428
Unacceptable	18	12	6	0.667
Total	90	43	47	0.478

Tables 3.6, 3.7 and 3.8 show the agreement between human gradings and gradings assigned based on each of the semi-automatic evaluation measures. In 57 out of the 90 summaries human grades and grades assigned based on the value of Average Continuity agreed, giving rise to the highest overall agreement rate of 0.633 among the three evaluation measures. Kendall's τ had the second best overall agreement ratio of 0.567. Agreement between grades assigned based on Spearman's coefficient and human grades was the lowest (agreement ratio=0.478). Moreover, Table 3.8 reports a perfect agreement (agreement ratio=1.0) between

Tab. 3.7: Kendall coefficient vs Subjective grading

Grade	Kendall	Agree	Disagree	Ratio
Perfect	32	17	15	0.531
Acceptable	17	9	8	0.529
Poor	25	12	13	0.480
Unacceptable	16	13	3	0.812
Total	90	51	39	0.567

Tab. 3.8: Average continuity vs Subjective grading

Grade	Average Continuity	Agree	Disagree	Ratio
Perfect	14	14	0	1.000
Acceptable	7	5	2	0.714
Poor	8	2	6	0.250
Unacceptable	61	36	25	0.590
Total	90	57	33	0.633

the grades assigned by Average Continuity and human grades for perfectly ordered summaries. This means that average continuity is a very reliable measure to judge perfectly ordered summaries. However, average continuity rejects 61 out of the 90 summaries as *Unacceptable*. On the other hand, Kendall's τ reports a higher agreement with subjective grades for *Unacceptable* summaries (agreement ratio=0.812 in Table 3.7). This means when Kendall's τ measure rejects a summary as being *Unacceptable*, its judgment is very reliable. This contrasting behavior of Average Continuity and Kendall's τ is particularly important when evaluating sentence orderings for a specific application. For instance, if we want to select a few summaries with very good sentence orderings then the judgment based on Average Continuity is appropriate. Average Continuity has a high precision but a low recall in identifying perfect summaries. Average Continuity might miss out certain number of perfect summaries. On the other hand, Kendall's τ will judge several non-perfect summaries as perfect. If we want to adopt a more conservative evaluation measure which penalizes improper orderings, then average continuity would be the better choice.

4. PAIR-WISE SENTENCE COMPARISON APPROACH

4.1 Pair-wise comparison of sentences

In chapter 2 I described a bottom-up approach to sentence ordering for multi-document summarization. In this chapter I explain a different approach to the same problem. In this approach we compare only sentence-pairs. For a set of N sentences there are $N(N-1)/2$ number of different sentence-pairs. Moreover, the two sentences in each pair can be ordered in two different ways. This gives rise to an interesting combinatorial problem where you have to generate a total ordering among a set of N things (in this case we have to create a summary for a set of sentences) such that the total ordering is optimal in some sense. In our case, we are interested in finding the total ordering among sentences that forms the most coherent summary.

However, there are two difficulties in this approach. First, it is computationally prohibitive to generate all possible orderings of N items and evaluate each one of them for their degree of coherence. Theoretically there are $N!$ ways of ordering N different sentences. We take a greedy search algorithm in this thesis to reduce the search space. Secondly, the criterion we need to optimize, coherence in a text, is not a well defined function. Although various factors that contribute to coherence in a text have been identified by previous work on linguistics [11] the exact combination of these factors in multi-document summaries remains unknown. In this thesis, we attempt to define various criteria (some of these criteria were already discussed in chapter 2) and use a machine learning approach to find the best combination among them.

For sentences taken from the same document we keep the order in that document as done in single document summarization. However, we have to be careful when ordering sentences which belong to different documents. To decide the order among such sentences, we implement five ranking experts: Chronological, Probabilistic, Topical relevance, Precedent and Succedent. These experts return precedence preference between two sentences. Cohen [5] proposes an elegant learning model that works with preference functions and we adopt this learning model to our task. Each expert e generates a pair-wise preference function defined as following:

$$\mathrm{PREF}_e(u, v, Q) \in [0, 1]. \tag{4.1}$$

Where, u, v are two sentences that we want to order; Q is the set of sentences which has been already ordered. The expert returns its preference of u to v. If the expert prefers u to v then it returns a value greater than 0.5. In the extreme case where the expert is absolutely sure of preferring u to v it will return 1.0. On the other hand, if the expert prefers v to u it will return a value lesser than 0.5. In the extreme case where the expert is absolutely sure of preferring v to u it will return 0. When the expert is undecided of its preference between u and v it will return 0.5.

The linear weighted sum of these individual preference functions is taken as the total preference by the set of experts as follows:

$$\mathrm{PREF}_{total}(u, v, Q) = \sum_{e \in E} w_e \mathrm{PREF}_e(u, v, Q). \tag{4.2}$$

Therein: E is the set of experts and w_e is the weight associated to expert $e \in E$. These weights are normalized so that the sum of them is 1. We use the Hedge learning algorithm to learn the weights associated with each expert's preference function. Then we use the greedy algorithm proposed by Cohen [5] to get an ordering that approximates the total preference.

4.1.1 Chronological Expert

Chronological expert emulates conventional chronological ordering [22] which arranges sentences according to the dates on which the documents were published and preserves the appearance order for sentences in the same document. We define a preference function for the expert as follows:

$$\mathrm{PREF}_{chro}(u, v, Q) = \begin{cases} 1 & T(u) < T(v) \\ 1 & [D(u) = D(v)] \wedge [N(u) < N(v)] \\ 0.5 & [T(u) = T(v)] \wedge [D(u) \neq D(v)] \\ 0 & \text{otherwise} \end{cases} . \tag{4.3}$$

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. Chronological expert gives 1 (preference) to the newly published sentence over the old and to the prior over the posterior in the same article. Chronological expert returns 0.5 (undecided) when comparing two sentences which are not in the same article but have the same publication date.

4.1.2 Probabilistic Expert

Lapata [21] proposes a probabilistic model to predict sentence order. Her model assumes that the position of a sentence in the summary depends only upon the sentences preceding it. For example let us consider a summary T which has sentences $S_1, \ldots, S_n$ in that order. The probability $P(T)$ of getting this order is given by:

$$P(T) = \prod_{i=1}^{n} P(S_n | S_1, \ldots, S_{n-i}). \tag{4.4}$$

She further reduces this probability using bi-gram approximation as follows.

$$P(T) = \prod_{i=1}^{n} P(S_i | S_{i-1}) \tag{4.5}$$

She breaks each sentence into features and takes the vector product of features as follows:

$$P(S_i | S_{i-1}) = \prod_{(a_{<i,j>}, a_{<i-1,k>}) \in S_i \times S_{i-1}} P(a_{<i,j>}, a_{<i-1,k>}). \tag{4.6}$$

Feature conditional probabilities can be calculated using frequency counts of features as follows:

$$P(a_{<i,j>} | a_{<i-1,k>}) = \frac{f(a_{<i,j>}, a_{<i-1,k>})}{\sum_{a_{<i,j>}} f(a_{<i,j>}, a_{<i-1,k>})}. \tag{4.7}$$

Lapata [21] uses nouns,verbs and dependency structures as features. Where as in our expert we implemented only nouns and verbs as features. We performed back-off smoothing on the frequency counts in equation 4.7 as these values were sparse. Once these conditional probabilities are calculated, for two sentences u,v we can define the preference function for the probabilistic expert as follows:

$$\text{PREF}_{prob}(u, v, Q) = \begin{cases} \frac{1+P(u|r)-P(v|r)}{2} & Q \neq \emptyset \\ \frac{1+P(u)-P(v)}{2} & Q = \emptyset \end{cases}. \tag{4.8}$$

Where, Q is the set of sentences ordered so far and $r \in Q$ is the lastly ordered sentence in Q. Initially, Q is null and we prefer the sentence with higher absolute probability. When Q is not null and u is preferred to v, i.e. $P(u|r) > P(v|r)$, according to definition 4.8 a preference value greater than 0.5 is returned. If v is preferred to u, i.e. $P(u|r) < P(v|r)$, we have a preference value smaller than 0.5. When $P(u|r) = P(v|r)$, the expert is undecided and it gives the value 0.5.

We performed back-off smoothing [19] on the frequency counts in equation 4.7 as these values were sparse. In back-off smoothing, a portion of probabilities of frequently occurring terms are transferred to sparsely occurring terms. For simplicity, I shall write w_1^m to denote the n-gram of length m, $w_1, w_2, \ldots, w_m$. $C(w_1^m)$ is the count of w_1^m in the corpus. Then the smoothed conditional probability $P_s(w_i | w_{i-n+1}^{i-1})$ of seeing w_i after $w_{i-n+1}, \ldots, w_{i-1}$ is given recursively as follows, [26]

$$P_s(w_i|w_{i-n+1}^{i-1}) = \begin{cases} (1 - d_{w_{i-n+1}^{i-1}})\frac{C(w_{i-n+1}^{i})}{C(w_{i-n+1}^{i-1})} & C(w_{i-n+1}^{i}) > k \\ \alpha_{w_{i-n+1}^{i-1}} P_s(w_i|w_{i-n+2} \dots w_{i-1}) & \text{otherwise} \end{cases} . \tag{4.9}$$

In definition 4.9 the first condition applies to terms w_{i-n+1}^{i} which exceeds a certain value k of counts. When using this model to smooth probabilities in sparse data k is set to zero. Therefore, for terms appearing one or more times in the corpus the conditional probabilities are reduced by a factor of $0 < d_{w_{i-n+1}^{i-1}} < 1$. Setting this value to 0, does not reserve any probabilities to be assigned for sparse data. These reserved probabilities are then assigned to the unseen n-grams as shown in the second condition in definition 4.9. The factor $\alpha_{w_{i-n+1}^{i-1}}$ is selected as in equation 4.10 so that the total probability remains a constant.

$$\alpha_{w_{i-n+1}^{i-1}} = \frac{1 - \sum_{C(w_{i-n+1}^{i})>k}(1 - d_{w_{i-n+1}^{i-1}})\frac{C(w_{i-n+1}^{i})}{C(w_{i-n+1}^{i-1})}}{1 - \sum_{C(w_{i-n+1}^{i})<k} P_s(w_i|w_{i-n+2})} \tag{4.10}$$

In the probabilistic expert we need to consider only bi-grams of words which appear in consecutive sentences. Therefore, the recursive formula in 4.9 considers only bi-grams and uni-gram of words. The only remaining parameter in formula 4.9 is $d_{w_{i-n+1}^{i-1}}$. Katz [19] proposes a method based on Turing's estimate to decide the value of $d_{w_{i-n+1}^{i-1}}$). Before, explaining this method we shall redefine $d_{w_{i-n+1}^{i-1}}$ as D_r, where $r = C(w_{i-n+1}^{i-1})$. For higher r values we shall not discount the probabilities because higher frequencies are reliable.

$$D_r = 1 \text{ for } r > R \tag{4.11}$$

In my experiments I took frequencies over five to be reliable. Therefore, in my experiments I took $R = 5$. When, n_r is the number of words (n-grams of words) which occurred exactly r times in the corpus, Turing's estimate P_T for a probability of a word (n-grams of words) which occurred in the sample r times is,

$$P_T = \frac{r^*}{N}. \tag{4.12}$$

where,

$$r^* = (r+1)\frac{n_{r+1}}{n_r} \tag{4.13}$$

We shall select D_r such that the contribution of probabilities yielded by this method is proportional to the contributions by the Turing [10] estimate. Taking the proportional coefficient to be μ, we can write this relation as,

$$(1 - D_r) = \mu(1 - \frac{r^*}{r}). \tag{4.14}$$

The unique solution to the equation 4.14 is,

$$D_r = \frac{\frac{r^*}{r} - \frac{(k+1)n_{k+1}}{n_1}}{1 - \frac{(k+1)n_{k+1}}{n_1}} \text{ for } 1 \leq r \leq k \tag{4.15}$$

4.1.3 Topical Relevance Expert

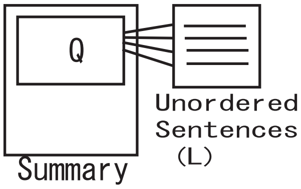


Fig. 4.1: Topical relevance expert

In MDS, the source documents could contain multiple topics. Therefore, the extracted sentences could be covering different topics. Grouping the extracted sentences which belong to the same topic, improves readability of the summary. Motivated by this fact, we designed an expert which groups the sentences which belong to the same topic. This expert prefers sentences which are more similar to the ones that have been already ordered. For each sentence l in the extract we define its topical relevance, $\text{topic}(l)$ as follows:

$$\text{topic}(l) = \max_{q \in Q} \text{sim}(l, q). \tag{4.16}$$

We use cosine similarity to calculate $\text{sim}(l, q)$. The preference function of this expert is defined as follows:

$$\text{PREF}_{topic}(u, v, Q) = \begin{cases} 0.5 & [Q = \oslash] \vee [\text{topic}(u) = \text{topic}(v)] \\ 1 & [Q \neq \oslash] \wedge [\text{topic}(u) > \text{topic}(v)] \\ 0 & \text{otherwise} \end{cases} . \tag{4.17}$$

Where, $\oslash$ represents the null set, u,v are the two sentences under consideration and Q is the block of sentences that has been already ordered so far in the summary.

4.1.4 Precedent Expert

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

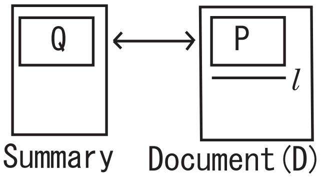


Fig. 4.2: Precedent expert

advanced, makes an unintelligible summary. As shown in figure 4.2, for each extracted sentence l, we can compare the block of text that appears before it in its source document (P) with the block of sentences which we have ordered so far in the summary (Q). If P and Q matches well, then we can safely assume that Q contains the necessary background information required by l. We can then place l after Q. Such relations among sentences are called precedence relations. Okazaki [35] proposes precedence relations as a method to improve the chronological ordering of sentences. He considers the information stated in the documents preceding the extracted sentences to judge the order. Based on this idea, we define precedence $\mathrm{pre}(l)$ of the extracted sentence l as follows:

$$\mathrm{pre}(l) = \max_{p \in P, q \in Q} \mathrm{sim}(p, q). \tag{4.18}$$

Here, P is the set of sentences preceding the extract sentence l in the original document. We calculate $\mathrm{sim(p, q)}$ using cosine similarity. The preference function for this expert can be written as follows:

$$\mathrm{PREF}_{pre}(u, v, Q) = \begin{cases} 0.5 & [Q = \oslash] \vee [\mathrm{pre}(u) = \mathrm{pre}(v)] \\ 1 & [Q \neq \oslash] \wedge [\mathrm{pre}(u) > \mathrm{pre}(v)] \\ 0 & \text{otherwise} \end{cases} . \tag{4.19}$$

4.1.5 Succedent Expert

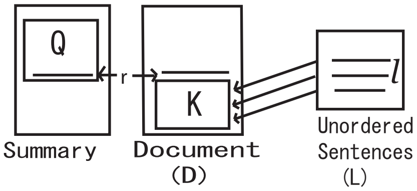


Fig. 4.3: Succedent expert

When extracting sentences from source documents, sentences which are similar to the ones that are already extracted, are usually ignored to prevent repetition of information. However, this information is valuable when ordering sentences. For example, a sentence that was ignored by the sentence extraction algorithm might turn out to be more suitable when ordering the extracted sentences. However, we assume that the sentence ordering algorithm is independent from the sentence extraction algorithm and therefore does not possess this knowledge regarding the left out candidates. This assumption improves the compatibility of our algorithm as it can be used to order sentences extracted by any sentence extraction algorithm. We design an expert which uses this information to order sentences.

Let us consider the situation depicted in Figure 4.3 where a block Q of text is ordered in the summary so far. The lastly ordered sentence r belongs to document D in which a block K of sentences follows r. The author of this document assumes that K is a natural consequence of r. However, the sentence selection algorithm might not have selected any sentences from K because it already selected some sentences with this information from some other document. Therefore, we search the extract L for a sentence that best matches with a sentence in K. We define succession as a measure of this agreement(4.20) as follows:

$$\mathrm{succ}(l) = \max_{k \in K} \mathrm{sim}(l, k). \tag{4.20}$$

Here, we calculate $\mathrm{sim(l, k)}$ using cosine similarity. Sentences with higher succession values are preferred by the expert. The preference function for this expert can be written as follows:

$$\mathrm{PREF}_{succ}(u, v, Q) = \begin{cases} 0.5 & [Q = \oslash] \vee [\mathrm{succ}(u) = \mathrm{succ}(v)] \\ 1 & [Q \neq \oslash] \wedge [\mathrm{succ}(u) > \mathrm{succ}(v)] \\ 0 & \text{otherwise} \end{cases} . \tag{4.21}$$

4.2 Ordering Algorithm

Using the five preference functions described in the previous sections, we compute the total preference function of the set of experts as defined by equation 4.2. Section 4.3 explains the method that we use to calculate the weights assigned to each expert's preference. In this section we will consider the problem of finding an order that satisfies the total preference function. Finding the optimal order for a given total preference function is NP-complete [5]. However, Cohen [5] proposes a greedy algorithm that approximates the optimal ordering. Once the unordered extract X and total preference (equation 4.2) are given, this greedy algorithm can be used to generate an approximately optimal ordering function $\hat{\rho}$.

```
let $V = X$
for each $v \in V$ do
    $\pi(v) = \sum_{u \in V} \mathrm{PREF}(v, u, Q) - \sum_{u \in V} \mathrm{PREF}(u, v, Q)$
while $V$ is non-empty do
    let $t = \arg\max_{u \in V} \pi(u)$
    let $\hat{\rho}(t) = |V|$
    $V = V - \{t\}$
    for each $v \in V$ do
        $\pi(v) = \pi(v) + \mathrm{PREF}(t, u) - \mathrm{PREF}(v, t)$
endwhile
```

However, there are some fundamental differences between the algorithm proposed by Cohen [5] and the one used by us. Our preference function has Q, the so far ordered summary, as a parameter in it. Therefore, the value of preference function changes while ordering. It remains an open question whether the guarantees for the result bounds in Cohen's original paper still holds for this modified version of the algorithm.

4.3 Learning Algorithm

Cohen [5] proposes a weight allocation algorithm that learns the weights associated with each expert in equation 4.2. We shall explain this algorithm in regard to our model of five experts.

Rate of learning $\beta \in [0, 1]$, initial weight vector $\vec{w}^1 \in [0, 1]^5$, s.t. $\sum_{e \in E} \vec{w}_e^1 = 1$.

Do for $t = 1, 2, \ldots, T$ where T is the number of training examples.

1. Get X^t; the set of sentences to be ordered.

2. Compute a total order $\hat{\rho}^t$ which approximates,

$$\mathrm{PREF}^t_{total}(u, v, Q) = \sum_{e \in E} \mathrm{PREF}^t_e(u, v, Q).$$

   We used the greedy ordering algorithm described in section 4.2 to get $\hat{\rho}^t$.

3. Order X^t using $\hat{\rho}^t$.

4. Get the human ordered set F^t of X^t. Calculate the loss for each expert.

$$\mathrm{Loss}(\mathrm{PREF}^t_e, F^t) = 1 - \frac{1}{|F|} \sum_{(u,v) \in F} \mathrm{PREF}^t_e(u, v, Q) \qquad (4.22)$$

5. Set the new weight vector,

$$w_e^{t+1} = \frac{w_e^t \beta^{\mathrm{Loss}(\mathrm{PREF}_e^t, F^t)}}{Z_t} \tag{4.23}$$

where, Z_t is a normalization constant, chosen so that, $\sum_{e \in E} w_e^{t+1} = 1$

In our experiments we set $\beta = 0.5$ and $w_i^1 = 0.2$. To explain equation 4.22 let us assume that sentence u comes before sentence v in the human ordered summary. Then the expert must return the value 1 for PREF(u,v,Q). However,if the expert returns any value less than 1, then the difference is taken as the loss. We do this for all such sentence pairs in F. For a summary of length N we have $N(N-1)/2$ such pairs. Since this loss is taken to the power of β, a value smaller than 1, the new weight of the expert gets changed according to the loss as in equation 4.23.

4.4 Evaluation

In addition to Kendall's τ coefficient, Spearman's rank correlation coefficient and Average Continuity which we already defined in chapter 2, I use sentence continuity [35] and Weighted Kendall coefficient for evaluate the sentence orderings generated by the proposed algorithm.

4.4.1 Continuity Metric

A summary is usually read from top to bottom in one dimension. The reader brings together continuous sentences in a text and interpret their meaning. Therefore, if the summary has a lot of continuous blocks of texts, it helps the reader to easily comprehend the summary. Okazaki [35] proposes a metric to grasp the continuity of a summary. A summary which can be read continuously is better than a one with lots of discontinuities. Using the permutations π, σ of the two orderings, he defines the continuity metric τ_c as,

$$\tau_c = \frac{1}{n} \sum_{i=1}^{n} \mathrm{equals}(\pi\sigma^{-1}(i), \pi\sigma^{-1}(i-1)+1). \tag{4.24}$$

Therein: $\pi(0) = \sigma(0) = 0$;

$$\mathrm{equals}(x,y) = \begin{cases} 1 & x = y \\ 0 & \text{otherwise} \end{cases}. \tag{4.25}$$

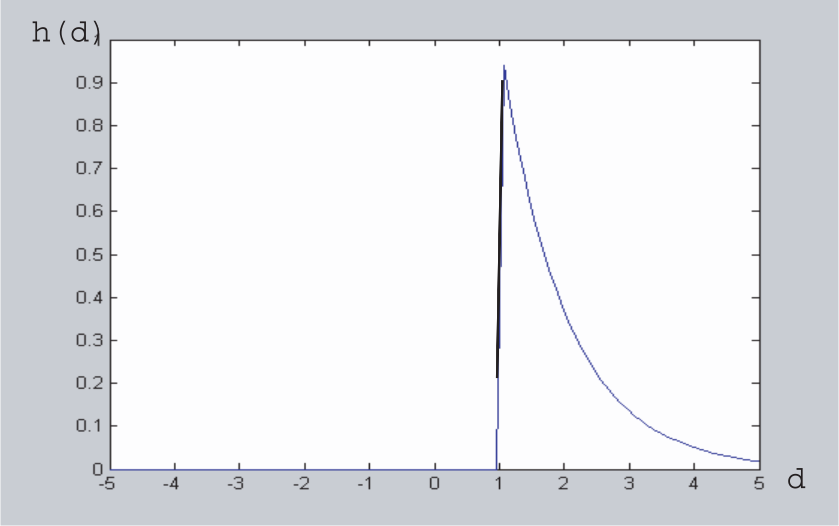


Fig. 4.4: Weighting function

4.4.2 Weighted Kendall Coefficient

Kendall correlation coefficient, τ_k was defined in 3.1 using permutations. It can also be stated more simply using the number of discordant pairs D as,

$$\tau_k = 1 - \frac{2D}{^nC_2}. \tag{4.26}$$

However, one major drawback of this metric when evaluating sentence orderings is that, it does not take into consideration the relative distance between the discordant pairs. On the other hand, Spearman correlation coefficient, τ_s, considers the relative distance between all pairs and does not distinguish between discordant and concordant pairs. None of the above mentioned metrics consider the position of the sentences in the summary. However, when reading a text a human reader is likely to be more sensitive to a closer discordant pair than a pair far apart. Therefore, a closer discordant pair is more likely to harm the readability of the summary compared to a far apart discordant pair. In order to reflect these difference in our metric, we use an exponentially decreasing weight function as follows.

$$h(d) = \begin{cases} \exp(1-d) & d \geq 1 \\ 0 & \text{else} \end{cases} \tag{4.27}$$

This weighting function can be expressed graphically as in figure 4.4.2. Going by the traditional Kendall's τ coefficient we defined our weighted Kendall coefficient as following, so that it becomes a metric in $[1, -1]$ range.

Tab. 4.1: Weights learned

Expert	Chronological	Probabilistic	Topical Relevance	Precedent	Succedent
Weights	0.327947	0.000039	0.016287	0.196562	0.444102

$$\tau_w = 1 - \frac{2\sum_d h(d)}{\sum_{i=1}^{n} h(i)} \tag{4.28}$$

4.5 Results

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 human selected extracts for 30 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. We used 10-fold cross validation to learn the weights assigned to each expert in our proposed algorithm. These weights are shown in table 4.1. According to table 4.1, succedent, chronology and precedent experts have the highest weights among the five experts and therefore almost entirely control the process of ordering. Whereas probabilistic and topical relevance experts have almost no influence on their decisions. However, we cannot directly compare Lapata's [21] approach with our probabilistic expert as we do not use dependency structure in our probability calculations. Moreover, Topical relevance, Precedent and Succedent experts require other experts to guide them at the start as they are not defined when Q is null. This inter-dependency among experts makes it difficult to interpret the results in table 4.1. However, we could approximately consider the values of the weights in table 4.1 as expressing the reliability of each expert's decisions.

We ordered each extract by five methods: Random Ordering (RO); Probabilistic Ordering (PO); Chronological Ordering (CO); Learned Ordering (LO); and HO (Human-made Ordering) and evaluated the orderings. The results are shown in table 4.2.

According to table 3.2 LO outperforms RO,PO and CO in all metrics. ANOVA test of the results shows a statistically significant difference among the five methods compared in table 3.2 under 0.05 confidence level. However, we could not

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

Tab. 4.2: Comparison with Human Ordering

	Spearman	Kendall	Continuity	Weighted Kendall	Average Continuity
RO	-0.267	-0.160	-0.118	-0.003	0.024
PO	0.062	0.040	0.187	0.013	0.029
CO	0.774	0.735	0.629	0.688	0.511
LO	0.783	0.746	0.706	0.717	0.546
HO	1.000	1.000	1.000	1.000	1.000

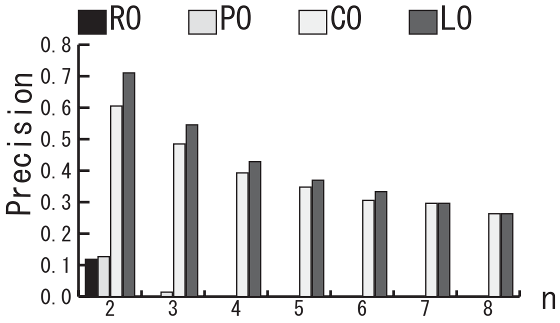


Fig. 4.5: Precision vs sentence n-gram length

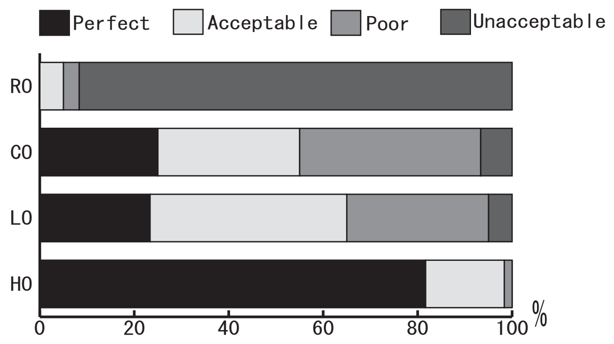


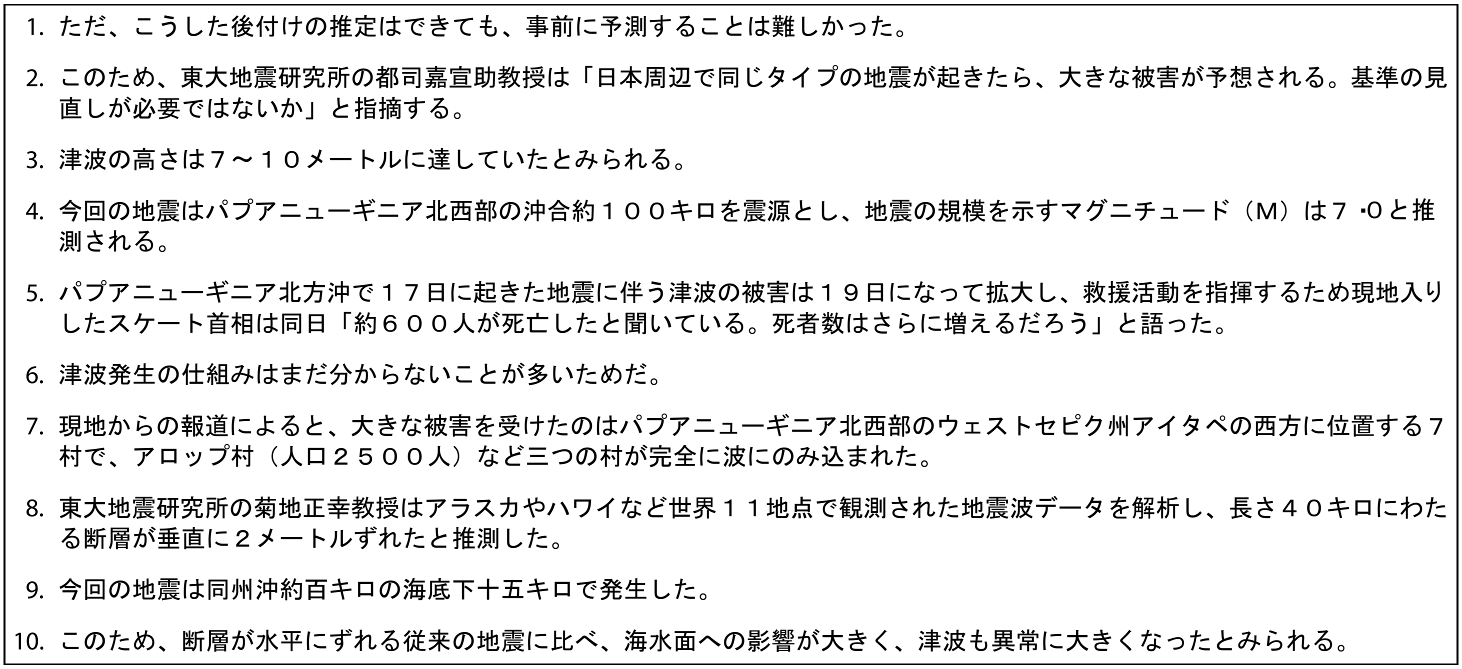
Fig. 4.6: Human Evaluation

find a statistically significant difference between CO and LO. Topical relevance, Precedent and Succedent experts cannot be used stand-alone to generate a total ordering because these experts are not defined at the start, where Q is null. These experts need Chronological and Probabilistic experts to guide them at the beginning. Therefore we have not compared these orderings in table 4.2.

Continuity precision, defined in equation 3.3, against the length of continuity n, is shown in figure 4.5. According to figure 4.5, for sentence n-grams of length up to 6, LO has the highest precision (defined by equation 3.3) among the compared orderings. PO did not possess sentence n-grams for n greater than two. Due to the sparseness of the higher order n-grams, precision drops in an exponential-like curve with the length of sentence continuity n. This justifies the logarithmic mean in the definition of average continuity in equation 3.5. A similar tendency could be observed for the BLEU metric [36].

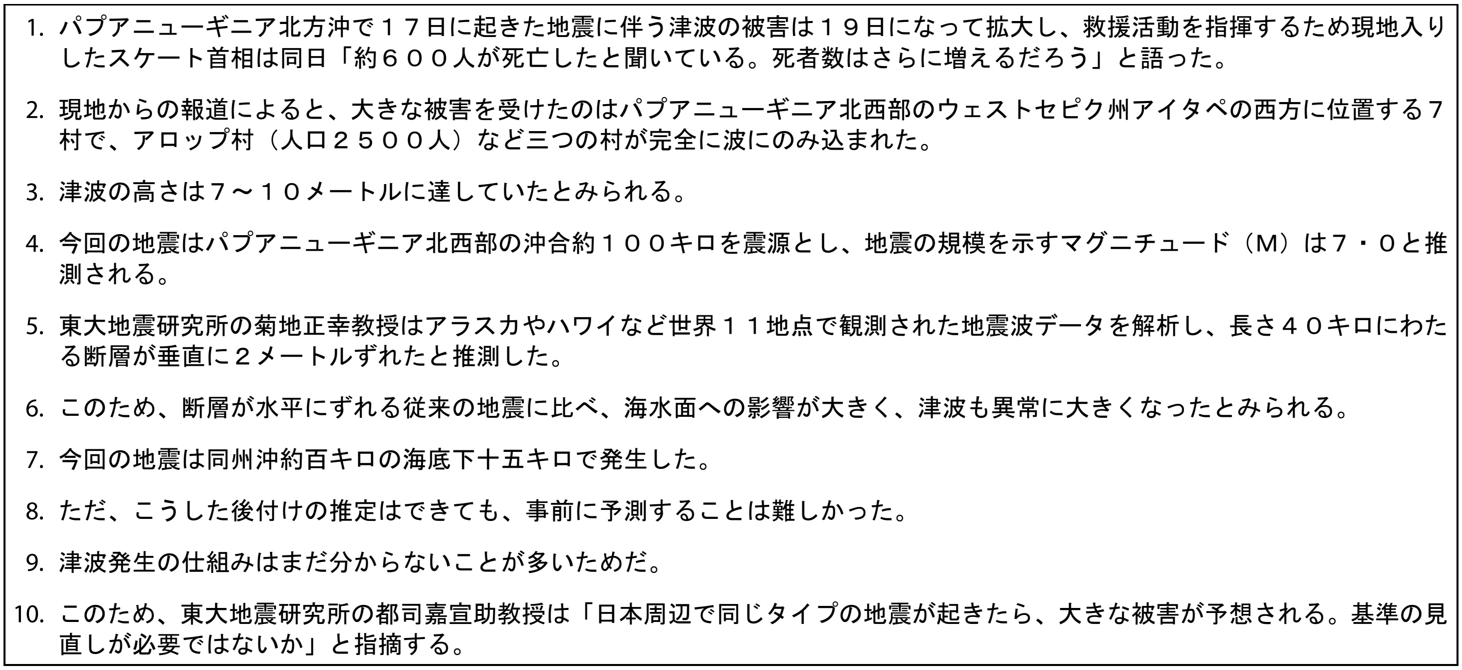
We also performed a human evaluation of our orderings. We asked two human judges to grade the summaries into four categories. The four grades are; *perfect*: no further adjustments are needed, *acceptable*: makes sense even though there is some room for improvement, *poor*: requires minor amendments to bring it up to the acceptable level, *unacceptable*: requires overall restructuring rather than partial revision. The result of the human evaluation of the 60 (2×30) summaries is shown in figure 4.6. It shows that most of the randomly ordered summaries (RO) are unacceptable. Although both CO and LO have same number of perfect summaries, the acceptable to poor ratio is better in LO. Over 60 percent of LO is either perfect or acceptable. Kendall's coefficient of concordance (W), which assesses the inter-judge agreement of overall ratings, reports a higher agreement between judges with a value of $W = 0.937$.

Although relatively simple in implementation, the chronological orderings works satisfactorily in our experiments. This is mainly due to the fact that the TSC corpus only contains news paper articles. Barzilay [1] shows chronological ordering to work well with news summaries. In news articles, events normally



1. ただ、こうした後付けの推定はできても、事前に予測することは難しかった。
2. このため、東大地震研究所の都司嘉宣助教授は「日本周辺で同じタイプの地震が起きたら、大きな被害が予想される。基準の見直しが必要ではないか」と指摘する。
3. 津波の高さは７～１０メートルに達していたとみられる。
4. 今回の地震はパプアニューギニア北西部の沖合約１００キロを震源とし、地震の規模を示すマグニチュード（Ｍ）は７ ･０と推測される。
5. パプアニューギニア北方沖で１７日に起きた地震に伴う津波の被害は１９日になって拡大し、救援活動を指揮するため現地入りしたスケート首相は同日「約６００人が死亡したと聞いている。死者数はさらに増えるだろう」と語った。
6. 津波発生の仕組みはまだ分からないことが多いためだ。
7. 現地からの報道によると、大きな被害を受けたのはパプアニューギニア北西部のウェストセピク州アイタペの西方に位置する７村で、アロップ村（人口２５００人）など三つの村が完全に波にのみ込まれた。
8. 東大地震研究所の菊地正幸教授はアラスカやハワイなど世界１１地点で観測された地震波データを解析し、長さ４０キロにわたる断層が垂直に２メートルずれたと推測した。
9. 今回の地震は同州沖約百キロの海底下十五キロで発生した。
10. このため、断層が水平にずれる従来の地震に比べ、海水面への影響が大きく、津波も異常に大きくなったとみられる。

Fig. 4.7: Randomly Ordered



1. パプアニューギニア北方沖で１７日に起きた地震に伴う津波の被害は１９日になって拡大し、救援活動を指揮するため現地入りしたスケート首相は同日「約６００人が死亡したと聞いている。死者数はさらに増えるだろう」と語った。
2. 現地からの報道によると、大きな被害を受けたのはパプアニューギニア北西部のウェストセピク州アイタペの西方に位置する７村で、アロップ村（人口２５００人）など三つの村が完全に波にのみ込まれた。
3. 津波の高さは７～１０メートルに達していたとみられる。
4. 今回の地震はパプアニューギニア北西部の沖合約１００キロを震源とし、地震の規模を示すマグニチュード（Ｍ）は７・０と推測される。
5. 東大地震研究所の菊地正幸教授はアラスカやハワイなど世界１１地点で観測された地震波データを解析し、長さ４０キロにわたる断層が垂直に２メートルずれたと推測した。
6. このため、断層が水平にずれる従来の地震に比べ、海水面への影響が大きく、津波も異常に大きくなったとみられる。
7. 今回の地震は同州沖約百キロの海底下十五キロで発生した。
8. ただ、こうした後付けの推定はできても、事前に予測することは難しかった。
9. 津波発生の仕組みはまだ分からないことが多いためだ。
10. このため、東大地震研究所の都司嘉宣助教授は「日本周辺で同じタイプの地震が起きたら、大きな被害が予想される。基準の見直しが必要ではないか」と指摘する。

Fig. 4.8: Ordered by the Learned Algorithm

occur in a chronological order. To evaluate the true power of the other experts in our algorithm, we need to experiment using other genre of summaries other than news summaries.

Finally, I show an example of sentence orderings by the learned algorithm. The summary is about the earthquake in Papua New Guinea in 1998. Figure 4.7 is the randomly ordered extract for the summary. Learned algorithm takes the set of sentences in figure 4.7 as input and orders them as in figure 4.8.

4.6 Bottom-up vs Pair-wise Approaches

In this thesis, I compared two different approaches: a bottom-up approach and a pair-wise comparison approach to order a given set of extracted sentences to create a coherent summary. The bottom-up approach reports better results in both subjective evaluations and evaluations based on automatic correlation measures. Reasons for the superiority of the bottom-up approach can be listed as follows.

1. *Cyclic Orderings*
   Pair-wise approach considers a pair of sentences at a time and decide the order between them. However, when we use the ordering algorithm described in section 4.2 we might end up with a cyclic ordering. To illustrate this fact let us consider three sentences A, B and C. Let us assume that pair-wise comparison of these three sentences gives the following partial ordering. $A \succ B$, $B \succ C$ and $C \succ A$. In this case A, B, C yields a cyclic order. The final ordering produced by the pair-wise ordering algorithm described in section 4.2 depends on the greedy selection it makes among the many different acceptable orderings.

2. *Pair Clipping*
   Pair-wise approach can yield sub-optimal orderings even when there are no cyclic partial orders. For example, let us assume we compared A and B and decided the partial order between them to be $A \succ B$. Furthermore, let us assume the comparison between C and D gave us the partial order $C \succ D$. From a sentence ordering for a summary point-of-view we would like to keep sentence B after A and D after C. However, further comparisons between B and C might force us to select the total order $A \succ C \succ B \succ D$, even without comparing A and C (this is possible due to greedy comparisons in the ordering algorithm). Such *clipping* of pairs reduces continuity in a summary. This is one major source of errors in the pair-wise approach that results in poor average continuity scores. On the other hand, in the bottom-up approach once we have merged two segments, we do not break the merged segment back into sentences. Neither do we *clip* two segments.

I believe the better continuity scores reported for the bottom-up approach are attributable to this nature of the algorithm.

The two approaches have comparable computational complexities. Pair-wise approach requires comparing all sentence pairs. For a summary with length N, this amounts to $N(N-1)$ comparisons (there are $N(N-1)/2$ number of sentence pairs and we need to make two comparisons for each sentence pair). Even with the bottom-up approach in its initial step, we need to compare all text segments which amounts to exactly the same number of comparisons. However, due to the hierarchical nature of the clustering the number of segments to compare reduces one at a time with the progress of the algorithm. In the pair-wise approach we reduce the number of comparisons just by selecting the best pair as at that point (greedy selection). Although exact computation of complexity bounds are difficult due to the complex interactions of support vector machines in the bottom-up approach, it is effectively a hierarchical clustering algorithm and have $\bigcirc(N^4)$.

I believe that bottom-up approach might be closely related to how humans read and comprehend a text. We first read sentences in paragraphs in a linear manner. Most texts are read from top to bottom. We then comprehend each paragraph. Finally, we comprehend the overall text using our understanding of the individual paragraphs. The hierarchical property and the non-clipping property of the bottom-up approach closely simulates this process of human understanding of text.

5. FUTURE WORK

In this thesis, I discussed the problem of automatically ordering a set of sentences extracted from multiple documents in order to create a coherent summary. In this chapter, I discuss the future work that I intend to carry on these lines and possible applications of current work to related problems.

Although I focused on the problem of sentence ordering in the context of multi-document summaries, the problem of text coherence is by no means limited to automatic text summarization. The problem of automatically arranging information in a text to make it coherent, is an old problem in computational linguistics [11]. Although various factors that contribute to coherence in a text such as ellipsis, repetition, co-reference have been identified in linguistic literature, computational method have not successfully made use of them. Automatic identification of factors that contribute to textual coherence is a challenging task that needs to be addressed in detail in future research in this field.

Recently, the task of textual entailment [1] has received much attention. Given two sentences A and B, textual entailment problem attempts to identify whether one can infer sentence B from sentence A (or vice-versa). If we can infer sentence B from A, then it is said that sentence A *entails* sentence B. This is a fundamental problem that one needs to solve in order to properly order sentences in a text to make it coherent. Although certain machine learning techniques have been employed to automatic identification textual entailment the results still remain sub-optimal. In my future research, I plan to explore the possibility of extending the method I proposed for sentence ordering for multi-document summarization to textual entailment problem.

[1] http://www.pascal-network.org/Challenges/RTE/

ACKNOWLEDGEMENT

This thesis would not have been possible if not for the advice and support by my adviser professor Mitsuru Ishizuka. I would like to take this opportunity to convey my gratitude for his insightful comments and directions throughout my masters research.

I would like to thank Dr. Yutaka Matsuo for showing the guidelines for most of my papers. Discussions at our weekly research meeting were very helpful in developing the methods described in this thesis.

My special thanks goes to Dr. Naoaki Okazaki for his progressive comments throughout my masters research. Without his directions my work in multi-document summarization would not be possible.

I would like to thank the members of Ishizuka Lab for their kind support and encouragement.

The work mentioned in this thesis was submitted to various conferences and workshops. I received many invaluable comments during the process which greatly helped to improve the quality of the research. I would like to thank all the anonymous reviewers for their comments.

I used Yomiuri and Mainichi newspaper data and NTCIR data collection for the experiments in sentence ordering. I would like to thank the workshop organizers for providing the datasets.

BIBLIOGRAPHY

[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] Regina Barzilay and Mirella Lapata. Modeling local coherence: An entity-based approach. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 141–148, 2004.

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

[4] 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 Retreival*, 1998.

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

[6] N. Elhadad and K. McKeown. Generating patient specific summaries of medical articles. In *Proceedings of the NAACL 2001 Workshop on Automatic Summarization*, 2001.

[7] E. Filatova and E. Hovy. Assining time-stamps to event-clauses. In *Proceedings of the 2001 ACL Workshop on Temporal and Spatial Information Processing*, 2001.

[8] Salton G. and McGill M.J. *Introduction to Modern Information Retreival*. McGraw-Hill, 1988.

[9] A. Garnham. *Mental models and the interpretations of anaphora*. Psychology Press, 2001.

[10] I.J. Good. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40:237–264, 1953.

[11] Halliday and Ruqaiya Hasan. *Cohesion in English*. Longman, 1976.

[12] T. Hastie and R. Tibshirani. Classification by pairwise coupling. *Technical report, Stanford University and University of Toronto*, 1996.

[13] Eduard Hovy and Chin-Yew Lin. Automatic evaluation of summaries using *n*-gram co-occurrence statistics. In *Proceedings of the 1st Human Language Technology Conference and Annual Meeting of the North American Chapter of the Association for Computational Linguistics*, pages 71–78, 2003.

[14] Paul D. Ji and Stephen Pulman. Sentence ordering with manifold-based classification in multi-document summarization. In *Proceedings of Empherical Methods in Natural Language Processing*, 2006.

[15] P. Johnson-Laird. *Mental Models*. Cambridge University Press, 1983.

[16] M. Kan, J. Klavans, and K. McKeown. Linear segmentation and segment significance. *Proceedings of 6th Workshop on Very Large Corpora (WVLC-98)*, pages 197–205, 1998.

[17] Nikiforos Karamanis. Evaluating centering for sentence ordering in two new domains. In *Proceedings of HLT-NAACL*, pages 65–68, 2006.

[18] Nikiforos Karamanis and Chris Mellish. Using a corpus of sentence orderings defined by many experts to evaluate metrics of coherence for text structuring. In *Proceedings of the 10th European Workshop on Natural Language Generation*, pages 174–179, 2005.

[19] Salva M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics Speech and Signal Processing*, 33(3):400–401, 1987.

[20] Maurice G. Kendall. A new measure of rank correlation. *Biometrika*, 30:81–93, 1938.

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

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

[23] P.H. Luhn. Automatic creation of literature abstracts. *IBM Journal*, pages 159–165, 1958.

[24] Inderjeet Mani and Mark T. Maybury, editors. *Advances in automatic text summarization*. The MIT Press, 2001.

[25] W. Mann and S. Thompson. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281, 1988.

[26] Christopher D. Manning and Hinrich Schutze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts London, England, 2 edition, 2002.

[27] 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.

[28] Daniel Marcu. The rhetorical parsing of unrestricted texts: A surface-based approach. *Computational Linguistics*, 26(3):395–448, 2000.

[29] 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.

[30] Kathleen R. McKeown, Regina Barzilay, David Evans, Vasileios Hatzivassiloglou, Judith L. Klavans, Ani Nenkova, Carl Sable, Barry Schiffman, and Sergey Sigelman. Tracking and summarizing news on a daily basis with columbia's newsblaster. In *Proceedings of HLT*, 2002.

[31] Lapata Mirella. Automatic evaluation of information ordering. *Computational Linguistics*, 32(4), 2006.

[32] Naoaki Okazaki and Sophia Ananiadou. Building an abbreviation dictionary using a term recognition approach. *Bioinformatics*, 22(24):3089–3095, 2006.

[33] Naoaki Okazaki and Sophia Ananiadou. A term recognition approach to acronym recognition. In *Proceedings of COLING/ACL 2006 Main Conference Poster Sessions*, pages 643–650, 2006.

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

[35] Naoaki Okazaki, Yutaka Matsuo, and Mitsuru Ishizuka. Improving chronological orderin. *ACM-TALIP, to appear in 2005.*, 2005.

[36] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 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, 2002.

[37] J. Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. *Advances in Large Margin Classifiers*, pages 61–74, 2000.

[38] Dragomir R. Radev and Kathy McKeown. Generating natural language summaries from multiple on-line sources. *Computational Linguistics*, 24(3):469–500, 1999.

[39] A.J. Sanford and S.C. Garrod. What, when, and how?: Questions of immediacy in anaphoric reference resolution. *Language and Cognitive Processes*, 4(3/4):235–262, 1989.

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

[41] G. Wahba. The bias-variance tradeoff and the randomized gacv. *Advances in Neural Information Processing Systems*, 11, 1999.