



A preference learning approach to sentence ordering for multi-document summarization

Danushka Bollegala^{*}, Naoaki Okazaki, Mitsuru Ishizuka

Graduate School of Information Science and Technology, The University of Tokyo, 7-3-1, Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

ARTICLE INFO

Article history:

Received 1 March 2011

Received in revised form 3 June 2012

Accepted 5 June 2012

Available online 19 June 2012

Keywords:

Sentence ordering

Multi-document summarization

Natural language processing

ABSTRACT

Ordering information is a difficult but an important task for applications generating natural-language texts such as multi-document summarization, question answering, and concept-to-text generation. In multi-document summarization, information is selected from a set of source documents. Therefore, the optimal ordering of those selected pieces of information to create a coherent summary is not obvious. Improper ordering of information in a summary can both confuse the reader and deteriorate the readability of the summary. Therefore, it is vital to properly order the information in multi-document summarization. We model the problem of sentence ordering in multi-document summarization as a one of learning the optimal combination of preference experts that determine the ordering between two given sentences. To capture the preference of a sentence against another sentence, we define five preference experts: *chronology*, *probabilistic*, *topical-closeness*, *precedence*, and *succession*. We use summaries ordered by human annotators as training data to learn the optimal combination of the different preference experts. Finally, the learnt combination is applied to order sentences extracted in a multi-document summarization system. The proposed sentence ordering algorithm considers pairwise comparisons between sentences to determine a total ordering, using a greedy search algorithm, thereby avoiding the combinatorial time complexity typically associated with total ordering tasks. This enables us to efficiently order sentences in longer summaries, thereby rendering the proposed approach useable in real-world text summarization systems. We evaluate the sentence orderings produced by the proposed method and numerous other baselines using both semi-automatic evaluation measures as well as performing a subjective evaluation.

© 2012 Elsevier Inc. All rights reserved.

1. Introduction

The rapid growth of the World Wide Web has resulted in large amounts of electronically available textual information. We use Web search engines to retrieve information relevant to a particular query. However, often Web search engines return more than one relevant search result. A user must read all those Web documents and obtain the necessary information. A text summarization system can reduce the time and effort required by a user to read a set of documents by automatically generating a short and informative summary of all the information that exist in the set of documents. The problem of generating a single coherent summary from a given set of documents that describes a particular event, is referred to as

^{*} Corresponding author.

E-mail addresses: danushka@iba.t.u-tokyo.ac.jp (D. Bollegala), okazaki@is.s.u-tokyo.ac.jp (N. Okazaki), ishizuka@i.u-tokyo.ac.jp (M. Ishizuka).

multi-document summarization. The related problem of generating a single coherent summary from a *single document* is named as *single document summarization*.

Multi-document summarization [36,7,12] tackles the information overload problem by providing a condensed and coherent version of a set of documents. Among a number of sub-tasks involved in multi-document summarization including sentence extraction, topic detection, sentence ordering, information extraction, and sentence generation, most multi-document summarization systems have been based on an extraction method, which identifies important textual segments (e.g. sentences or paragraphs) in source documents. To reconstruct the text structure for summarization, it is important for such multi-document summarization systems to determine a coherent arrangement for the textual segments extracted from multi-documents.

A summary with improperly ordered sentences both confuses the reader and degrades the quality/reliability of the summary. Barzilay et al. [2] show that the proper order of extracted sentences significantly improves their readability. Lapata [22] experimentally shows that the time taken to read a summary strongly correlates with the arrangement of sentences in the summary.

For example, consider the three sentences shown in Fig. 1, selected from a reference summary in Document Understanding Conference (DUC) 2003 dataset. The 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 talk about the storm, *Gilbert*, the sentence ordering shown in Fig. 1 is inadequate. For example, the phrase, *such storms*, in sentence 1, in fact 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, where a summary is created using only one document, it is natural to arrange the extracted information in the same order as in the original document. In contrast, for multi-document summarization, we need to establish a strategy to arrange sentences extracted from different documents. Therefore, the problem of sentence ordering is more critical for multi-document summarization systems compared to single document summarization systems. In this paper, we focus on the sentence ordering problem in multi-document summarization.

Ordering extracted sentences into a coherent summary is a non-trivial task. Rhetorical relations [28] such as *cause-effect* relation and *elaboration* relation exist between sentences in a coherent text. If we can somehow determine the rhetorical relations that exist among a given set of sentences, then we can use those relations to infer a coherent ordering of the set of sentences. For example, if a sentence *A* is the effect of the cause mentioned in a sentence *B*, then we might want to order the sentence *A* after sentence *B* in a summary that contains both sentences *A* and *B*. Unfortunately, the problem of automatically detecting the rhetorical structure of an arbitrary text is a difficult and an unsolved one. The performance reported by the state-of-the-art rhetorical structure analysis systems is insufficient to be used in a sentence ordering system.

The task of constructing a coherent summary from an unordered set of sentences has several unique properties that makes it challenging. Source documents for a summary may have been written by different authors, have different writing styles, or written on different dates, and based on different background knowledge. Often a multi-document summarization system is presented with a set of articles that discuss about a particular news event. Those news articles are selected from different newspapers. Although the articles themselves are related and discuss a particular event, those articles are written by different authors. Therefore, the collection of texts that the multi-document summarization system receives is not always coherent with regard to their authorship. We cannot expect a set of extracted sentences from such a diverse set of documents to be coherent on their own.

The problem of information ordering is not limited to automatic text summarization, and concerns all natural language generation applications in general. A typical natural language generation (NLG) [37] system consists of six components: content determination, discourse planning, sentence aggregation, lexicalization, referring expression generation, and orthographic realization. Among those, information ordering is particularly important in discourse planning, and sentence aggregation [18,11,10]. In concept-to-text generation [37], given a concept (e.g. a keyword, a topic, or a collection of data), the objective is to produce a natural language text about the given concept. For example, consider the case where generating game summaries, given a database containing statistics of American football. A sentence ordering algorithm can support a natural language generation system by helping to order the sentences in a coherent manner.

This paper is organized as follows. In Section 2, we introduce the previous work on sentence ordering methods for multi-document summarization. Next, in Section 3, we present the proposed preference learning approach to sentence ordering. Specifically, we describe numerous preference experts and describe a method to learn the optimal combination of those experts using a set of sentences ordered by humans. Section 4 describes the semi-automatic evaluation measures that we use to evaluate the sentence ordering algorithms. Experimental results on a set of multi-document summaries are presented in

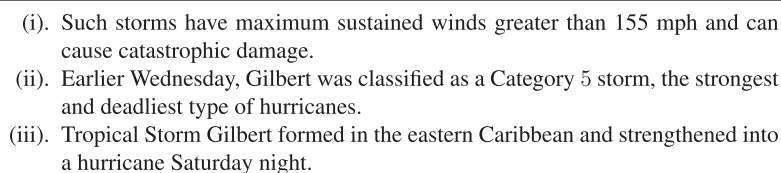
- 
- (i). Such storms have maximum sustained winds greater than 155 mph and can cause catastrophic damage.
 - (ii). Earlier Wednesday, Gilbert was classified as a Category 5 storm, the strongest and deadliest type of hurricanes.
 - (iii). Tropical Storm Gilbert formed in the eastern Caribbean and strengthened into a hurricane Saturday night.

Fig. 1. Randomly ordered sentences in a summary.

Section 5. We discuss the potential future research directions in Section 6. Finally, in Section 6, we discuss future research directions and conclude.

2. Related work

Existing methods for sentence ordering are divided into two approaches: making use of chronological information [30,24,2,33], and learning the natural order of sentences from large corpora [21,3,17]. A newspaper usually disseminates descriptions of novel events that have occurred since the last publication. For this reason, the chronological ordering of sentences is an effective heuristic for multi-document summarization [24,30]. Barzilay et al. [2] proposed an improved version of the chronological ordering by first grouping sentences into sub-topics discussed in the source documents, then arranging the sentences in each group chronologically.

Okazaki et al. [33] proposed an algorithm to improve the 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 that must 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. Experimental results show that their algorithm improves the chronological ordering significantly.

Lapata [21] presented a probabilistic model for text structuring and its application to sentence ordering. Her method computes the transition probability from one sentence to the next in two consecutive sentences, from a corpus based on the Cartesian product 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 [22] also proposed the use of the Kendall's rank correlation coefficient (Kendall's τ) for semi-automatically evaluating the differences between orderings produced by an algorithm and by a human. Although she did not compare her method against chronological ordering, it can be applied to generic domains, not relying on the chronological clues unique to newspaper articles.

Barzilay and Lee [3] proposed *content models* to deal with the topic transition in domain specific text. The content models are implemented as Hidden Markov Models (HMMs), in which the hidden states correspond to topics in the domain of interest (e.g. earthquake magnitude or previous earthquake occurrences), and state transitions capture possible information-presentation orderings. Experimental results show that their method outperformed Lapata's approach significantly. However, they did not compare their method against chronological.

Paul et al. [17] proposed a sentence ordering algorithm using a semi-supervised sentence classification and historical ordering strategy. Their algorithm includes three steps: the construction of sentence networks, sentence classification, and sentence ordering. First, they represent a summary as a network of sentences. Nodes in this network represent sentences in a summary, and edges represent transition probabilities between two nodes (sentences). Next, the sentences in the source documents are classified into the nodes in this network. The probability $p(c_k|s_i)$, of a sentence s_i in a source document belonging to a node c_k in the network, is defined as the probability of observing s_k as a sample from a Markov random walk in the sentence network. Finally, the extracted sentences are ordered according to the weights of the corresponding edges. They compare the sentence ordering produced by their method against manually ordered summaries using Kendall's τ . Unfortunately, they do not compare their results against the chronological ordering of sentences, which has been shown to be an effective sentence ordering strategy in multi-document news summaries.

The problem of "Learning to Rank (LETOR)" has been studied in the information retrieval community [4,15,35,42] and is closely related to the sentence ordering problem discussed in this paper. Given a set of documents retrieved as relevant search results for a particular query, in learning to rank the goal is to learn a ranking function that can be used to induce a total ordering among the retrieved documents according to their relevance to the query. Clickthrough data have been often used as a training signal to generate large training datasets. The sentence ordering problem in the context of multi-document summarization is closely-related to learning to rank problem studied in information retrieval in the sense that in both tasks we are given a set of items among which we must induce a total ordering. In this regard, it is possible to use most learning algorithms proposed for learning to rank in information retrieval to learn sentence ordering methods for multi-document summarization. However, there are some important differences between the two tasks that must be carefully considered. First, in multi-document summarization we have the access to the original set of documents from which the set of sentences to be ordered are extracted as auxiliary information. The original documents provide useful clues regarding the order among sentences. As described later in this paper, we use the original set of documents to construct several sentence ordering criteria. Second, the amount of training data available for sentence ordering is much less compared to that for learning to rank in information retrieval. Therefore, it remains unclear whether it is possible to sufficiently train some of the learning algorithms proposed for learning to rank under the settings for sentence ordering. Third, a human user expects a summary to be coherent and views it as a single text and not as a set of sentences. However, in the case of a search engine, the set of documents retrieved and displayed for a user query are not read as a continuous body of text but a collection of documents relevant to the query. Therefore, the requirement for textual coherence is much stronger in the case of sentence ordering in multi-document summarization.

Feng and Allan [13] proposed a method to automatically detect the incidents in a given set of news passages and to organize those incidents as a network. They refer to this task as *incident threading*. They define an *incident* as a real-world

occurrence that involves certain main characters, happening at a specific time and a place. They use chronological information such as the time stamp of a newspaper article to create an incident thread. Although both incident threading and sentence ordering in multi-document summarization focus on organizing information presented in newspaper articles, there are several important differences between the two tasks. First, unlike in incident threading, we are not required to detect incidents in sentence ordering. Second, in incident threading, different sentences that are related to the same incident are grouped and incidents are then arranged in a sequential order. On the other hand, in sentence ordering, we must induce a total ordering among all the extracted summary sentences, including those sentences that might belong to the same incident.

3. Sentence ordering in multi-document summarization

The first step in multi-document summarization is to extract a set of sentences from the given set of documents. The set of documents to be summarized can be either manually picked by a user or can be automatically retrieved from a search engine using some keywords that describe a particular event. Numerous methods have been proposed in previous work on multi-document summarization to extract a set of sentences to be included in a summary [25]. The second step of multi-document summarization is to order the extracted sentences such that to make a coherent summary. Our work specifically focuses on this second step of sentence ordering in multi-document summarization. We do not consider the first step of document retrieval or sentence extraction in this paper. By decoupling the sentence ordering problem from the sentence extraction problem in multi-document summarization, we can both study and evaluate the sentence ordering problem without taking into consideration the added complications in sentence extraction. Note that *all* previous work on sentence ordering for multi-document summarization have followed this de-coupling approach and consider sentence ordering as a separate problem.

The author of a particular document is likely to order the sentences logically to make the document coherent. Therefore, for sentences belonging to a particular document, we can safely retain this original order given by the author. In single document summarization this simple ordering strategy is often adequate to order all the extracted sentences because those sentences belong to a single document. However, we cannot apply this simple method to multi-document summarization because, the sentences belong to different documents. Such documents might have been written by various authors on various dates.

To decide the order among such sentences, we use five independent ordering strategies which we designate as *experts* in this paper. When presented with a pair of sentences, each of those experts gives its preference for one sentence over another as a value in the range $[0, 1]$. Each expert e is defined by a pairwise preference function as follows,

$$\text{PREF}_e(u, v, Q) \in [0, 1]. \quad (1)$$

where u, v are two sentences that we want to order, Q is the set of sentences which has been ordered so far by some ordering algorithm. Note that a total ordering exist among sentences in Q . The expert returns its preference of $u-v$. If the expert prefers $u-v$ then it returns a value greater than 0.5. In the extreme case where the expert is absolutely sure of preferring $u-v$ it will return the value 1. On the other hand, if the expert prefers $v-u$ it will return a value less than 0.5. In the extreme case where the expert is absolutely sure of preferring $v-u$ it will return 0. When the expert is undecided of its preference between u and v it will return 0.5. Note that initially Q will be the empty set (denoted by \emptyset in this paper) because we have not yet ordered any sentences.

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

$$\text{PREF}_{\text{total}}(u, v, Q) = \sum_{e \in E} w_e \text{PREF}_e(u, v, Q). \quad (2)$$

Therein: E is the set of experts and w_e is the weight associated with expert $e \in E$. These weights are normalized such that the sum of them equals to 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 [9] to get an ordering that approximates the total preference.

3.1. Chronological expert

Chronology expert reflects the chronological ordering [24,30], by which sentences are arranged in the chronological order of publication timestamps. A newspaper usually deals with novel events that have occurred since the last publication. Consequently, the chronological ordering of sentences has shown to be particularly effective in multi-document news summarization. As already discussed in Section 2, previous studies have proposed sentence ordering algorithms using chronological information. Publication timestamps are used to decide the chronological order among sentences extracted from different documents. However, if no timestamp is assigned to documents, or if several documents have the identical timestamp, the chronological ordering does not provide a clue for sentence ordering. Inferring temporal relations among events [26,27] using implicit time references (such as tense system) [23], and explicit time references (such as temporal adverbials) [14], might provide an alternative clue for chronological ordering. However, inferring temporal relations across a diverse set

of multiple documents is a difficult task. Consequently, by assuming the availability of temporal information in the form of timestamps, we define a preference function for the chronology expert as follows:

$$\text{PREF}_{\text{chro}}(u, v, Q) = \begin{cases} 1 & T(u) < T(v) \\ 1 & [D(u) = D(v)] \wedge [N(u) < N(v)] \\ 0.5 & [T(u) = T(v)] \wedge [D(u) \neq D(v)] \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

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.

The chronology expert assesses the appropriateness of arranging sentence v after u if sentence u is published earlier than sentence v , or if sentence u appears before v in the same article. For sentences extracted from the same source document, preferring the original order in the source document has proven to be effective for single document summarization [2]. The second condition in the chronological criterion defined in Eq. (3) imposes this constraint. If sentence u and v are published on the same day, but appear in different articles, the chronology expert assumes the order to be undefined and returns a preference value of 0.5. If none of the above conditions are satisfied, the chronological expert predicts that the sentence v will precede u . By assigning a score of zero for this condition in Eq. (3), the chronological expert guarantees that sentence orderings which contradicts with the definition of chronological ordering are not produced.

In addition to the formulation of chronology criterion defined by Eq. (3), in our preliminary experiments we tried alternatives that consider the absolute time difference between the publication dates of articles. For two sentences extracted from different articles (i.e. $D(u) \neq D(v)$), we defined the chronological distance between them as the difference of publication dates using the number of days. Moreover, the chronological distances in a summary are normalized to values in range [0, 1] by dividing from the maximum value of chronological distances. However, we did not find any significant improvement in the sentence orderings produced by this alternative approach in our experiments. Therefore, we only consider the simpler version of chronological criterion defined in Eq. (3).

3.2. Probabilistic expert

Events that are described in a newspaper article typically follows a fixed pattern. For example, a newspaper article on an earthquake usually first presents information regarding the epicenter of the earthquake, its magnitude and then describes information regarding any subsequent tsunamis. If we can learn such patterns between events, then it can be used to infer the ordering among sentences in a summary.

Based on this observation, 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 which precede it in the summary. For example let us consider a summary T which contains sentences S_1, \dots, S_n in that order. The probability $P(T)$ of observing this order is given by

$$P(T) = \prod_{i=1}^n P(S_i | S_1, \dots, S_{i-1}). \quad (4)$$

Using the Markov assumption that the probability of a sentence depends only upon the sentence that directly precedes it in the summary (i.e. given S_{i-1} , S_i is independent of S_1, \dots, S_{i-2}) she further reduces this probability to

$$P(T) = \prod_{i=1}^n P(S_i | S_{i-1}). \quad (5)$$

Because exact occurrences of two sentences in a corpus is rare, she represents each sentence using a set of features and takes the vector product of the two sets of features corresponding to two adjacent sentences as follows:

$$P(S_i | S_{i-1}) = \prod_{(a_{(i,j)}, a_{(i-1,k)}) \in S_i \times S_{i-1}} P(a_{(i,j)} | a_{(i-1,k)}). \quad (6)$$

Here, $a_{(i,j)}$ denotes the j th word in sentence S_i . Feature conditional probabilities can be calculated using frequency counts of features as follows:

$$P(a_{(i,j)} | a_{(i-1,k)}) = \frac{f(a_{(i,j)}, a_{(i-1,k)})}{\sum_{a_{(i,j)}} f(a_{(i,j)}, a_{(i-1,k)})}. \quad (7)$$

Here, $f(a_{(i,j)}, a_{(i-1,k)})$ denotes the number of times the word $a_{(i-1,k)}$ appears in a sentence S_{i-1} that is immediately followed by a sentence S_i that contains the word $a_{(i,j)}$. Lapata [21] uses Nouns, Verbs and dependency structures as features. Once these conditional probabilities are calculated, we can define the preference function for the probabilistic expert as follows,

$$\text{PREF}_{\text{prob}}(u, v) = \frac{1 + P(v|u) - P(u|v)}{2}. \quad (8)$$

where u, v are two sentences in the extract. When u is preferred to v (i.e. $P(u|v) > P(v|u)$), according to Eq. (8), a preference value greater than 0.5 is returned. If v is preferred to u (i.e. $P(v|u) < P(u|v)$), we have a preference value smaller than 0.5. When $P(v|u) = P(u|v)$, the expert returns the value 0.5.

One problem with the above-mentioned probabilistic model is the data sparseness – two words u and v might not occur in adjacent sentences in a given corpus although such occurrences might be possible in a much larger corpus. A standard approach to overcome sparseness in probabilistic models is to use some kind of a *smoothing* technique. In this paper, we perform back-off smoothing [19] on the frequency counts in Eq. (7). In back-off smoothing, a portion of probabilities of frequently occurring terms are transferred to sparsely occurring terms. For simplicity, we will write w_1^m to denote the n -gram of length m (length counted by the number of tokens), w_1, w_2, \dots, w_m . Moreover, $C(w_1^m)$ denotes 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 observing w_i after $w_{i-n+1}, \dots, w_{i-1}$ is given recursively as follows [29]

$$P_s(w_i|w_{i-n+1}^{i-1}) = \begin{cases} \left(1 - d_{w_{i-n+1}^{i-1}}\right) \frac{C(w_{i-n+1}^i)}{C(w_{i-n+1}^{i-1})} & C(w_{i-n+1}^i) > k \\ \alpha_{w_{i-n+1}^{i-1}} P_s(w_i|w_{i-n+2} \dots w_{i-1}) & \text{otherwise} \end{cases}. \quad (9)$$

In the definition given by Eq. (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 to sparse data. These reserved probabilities are then assigned to the unseen n -grams as shown in the second condition in Eq. (9). The factor $\alpha_{w_{i-n+1}^{i-1}}$ is selected as in Eq. (10) such that the total probability remains a constant

$$\alpha_{w_{i-n+1}^{i-1}} = \frac{1 - \sum_{C(w_{i-n+1}^i) > k} \left(1 - d_{w_{i-n+1}^{i-1}}\right) \frac{C(w_{i-n+1}^i)}{C(w_{i-n+1}^{i-1})}}{1 - \sum_{C(w_{i-n+1}^i) < k} P_s(w_i|w_{i-n+2} \dots w_{i-1})}. \quad (10)$$

In the probabilistic expert, we need to consider only pairs of words that appear in adjacent sentences. Therefore, the recursive formula in Eq. (9) is limited to bi-grams and unigrams of words. The only remaining parameter in Eq. (9) is $d_{w_{i-n+1}^{i-1}}$. Katz [19] proposes a method based on Turing's estimate to determine the value of $d_{w_{i-n+1}^{i-1}}$. Before, explaining this method we will 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.

$$D_r = 1 \text{ for } r > R \quad (11)$$

In our experiments, we assume frequencies over five to be reliable (i.e. $R = 5$). When, n_r is the number of words (n -grams of words) which occur exactly r times in the corpus, Turing's estimate P_T for the probability of a word (n -grams of words), which occurs in the sample r times is,

$$P_T = \frac{r^*}{N}, \quad (12)$$

where

$$r^* = (r + 1) \frac{n_{r+1}}{n_r}. \quad (13)$$

We select D_r such that the contribution of probabilities yielded by this method is proportional to the contributions by the Good-Turing [16] estimate. Taking the proportional coefficient to be μ , we can write this relation as

$$(1 - D_r) = \mu \left(1 - \frac{r^*}{r}\right). \quad (14)$$

The unique solution to Eq. (14) is

$$D_r = \frac{\frac{r^*}{r} - \frac{(k+1)n_{k+1}}{n_1}}{1 - \frac{(k+1)n_{k+1}}{n_1}}; \text{ for } 1 \leq r \leq k. \quad (15)$$

3.3. Topical-closeness expert

A set of documents discussing a particular event usually contains information related to multiple topics. For example, a set of newspaper articles related to an earthquake typically contains information about the magnitude of the earthquake, its

- (a) The earthquake crushed cars, damaged hundreds of houses and terrified people for hundreds of kilometers around.
- (b) A major earthquake measuring 7.7 on the Richter scale rocked north Chile Wednesday.
- (c) Authorities said two women, one aged 88 and the other 54, died when they were crushed under the collapsing walls.

Fig. 2. Three sentences from a summary about an earthquake.

location, casualties, and rescue efforts. Grouping sentences by topics has shown to improve the readability of a summary [2,3]. For example, consider the three sentences shown in Fig. 2, selected from a summary of an earthquake in Chile. Sentences (a) and (c) in Fig. 2 present details about the damage by the earthquake, whereas sentence (b) conveys information related to the magnitude and location of the earthquake. In this example, sentences (a) and (c) can be considered as topically related. Consequently, when the three sentences are ordered as shown in Fig. 2, we observe abrupt shifts of topics from sentence (a) to (b), and from (b) to (c). A better arrangement of the sentences that prevents such disfluencies is (b)–(a)–(c).

The topical-closeness expert deals with the association of two sentences, based on their lexical similarity. The expert reflects the ordering strategy proposed by Barzilay et al. [2], which groups sentences referring to the same topic. To measure the topical closeness of two sentences, we represent each sentence by a vector. First, we remove *stop words* (i.e. functional words such as *and*, *or*, *the*, etc.) from a sentence and lemmatize verbs and nouns. Second, we create a vector in which each element corresponds to the words (or lemmas in the case of verbs and nouns) in the sentence. Values of elements in this vector are either 1 (for words that appear in the sentence) or 0 (for words that do not appear in the sentence).¹

The topical-closeness expert prefers sentences which are more similar to the ones that have been already ordered. For each sentence l in the extracted set of sentences, we define its topical-closeness, $\text{topic}(l)$ as follows,

$$\text{topic}(l) = \max_{q \in Q} \text{sim}(l, q). \quad (16)$$

Here, q is a sentence in the set of sentences Q that has been ordered so far. We use cosine of the angle (i.e. cosine similarity) as the similarity $\text{sim}(l, q)$ between the two feature vectors corresponding to sentences l and q . Moreover, by considering the maximum similarity with any sentence that we have ordered so far (i.e. Q), we capture the sentence that is closest in topic to l . Using the above-defined topical-closeness measure, we define the preference function of the topical-closeness expert as follows,

$$\text{PREF}_{\text{topic}}(u, v, Q) = \begin{cases} 0.5 & [Q = \emptyset] \vee [\text{topic}(u) = \text{topic}(v)] \\ 1 & [Q \neq \emptyset] \wedge [\text{topic}(u) > \text{topic}(v)] \\ 0 & \text{otherwise} \end{cases} \quad (17)$$

where \emptyset represents the null set, u, v are the two sentences under consideration and Q is the set of sentences that has been already ordered in the summary. The topical-closeness expert determines the ordering between two sentences purely based on their topical-closeness scores as given by Eq. (16). The expert is undecided if the two sentences have exactly the same topical-closeness scores, or if we have not ordered any sentences (i.e. initial state), and returns the value of 0.5 under those conditions.

Note that although we use cosine similarity between vectors that represent two sentences as our sentence similarity measure in this paper, it is possible to incorporate any sentence similarity measure as $\text{sim}(l, q)$ in Eq. (16). Measuring the similarity between sentences is a more difficult problem compared to measuring similarity between two words. A sentence similarity measure must be sensitive to the semantic similarity between individual words, word order as well as syntactic structure (e.g. active vs. passive voice and present vs. past tense). Several sentence similarity measures have been studied in previous work on sentence similarity measures [1,31,43,34]. In particular, both WordNet-based lexical semantic similarity measures coupled with syntactic information have shown to be useful for accurately measuring the similarity between sentences [1]. Although our main focus in this paper is on sentence ordering, we intend to explore the possibility of using alternative sentence similarity measures for this task in our future work.

3.4. Precedence expert

In extractive multi-document summarization, only the important sentences that convey the main points discussed in source documents are selected to be included in the summary. However, a selected sentence can presuppose information from other sentences that were not selected by the sentence extraction algorithm. For example, consider the three sentences shown in Fig. 3, selected from a summary on hurricane Mitch. Sentence (a) describes the after-effects of the hurricane,

¹ Using the frequencies of words instead of the binary (0,1) values as vector elements, did not have a positive impact in our experiments. We think this is because, compared to a document, a sentence typically has a lesser number of words, and a word does not appear many times in a single sentence.

- (a) Honduran death estimates grew from 32 to 231 in the first two days, to 6,076 with 4,621 missing.
 (b) Honduras braced as category 5 Hurricane Mitch approached.
 (c) The EU approved 6.4 million in aid to Mitch's victims.

Fig. 3. Precedence relations in a summary.

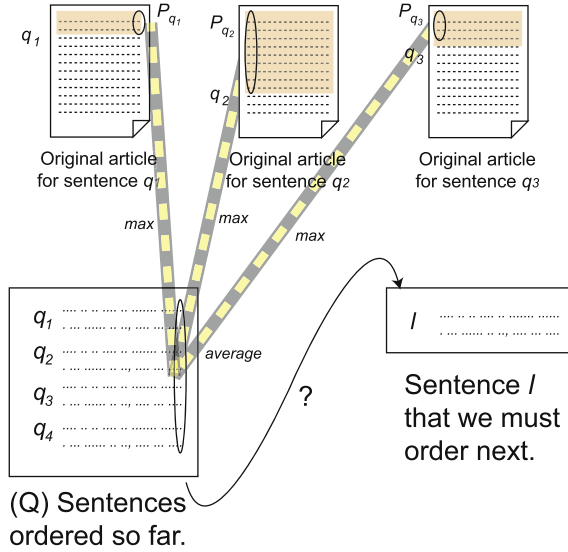


Fig. 4. Precedence expert.

whereas sentence (b) introduces the hurricane. To understand the reason for the deaths mentioned in sentence (a), one must first read sentence (b). Consequently, it is appropriate to arrange the three sentences in Fig. 3 in the order (b)-(a)-(c). In general, it is difficult to perform such an in-depth logical inference on a given set of sentences. Instead, we use source documents to estimate precedence relations. For example, assuming that in the source document from which sentence (a) was extracted, there exist a sentence that is similar to sentence (b), we can conclude that sentence (b) should precede sentence (a) in the summary.

To formally define the precedence criterion, let us consider the case illustrated in Fig. 4, where we must arrange a sentence l after ordering a segment of sentences Q up to this point. Each sentence in segment Q has the presuppositional information such as background information, or introductory facts that must be conveyed to a reader in advance. For example, the sentence $q_1 \in Q$ is preceded by a set of sentences, P_{q_1} , in the original article from which q_1 is extracted. If the information described in P_{q_1} is similar to that conveyed by the sentence l , then it is a good indicator that l should be ordered before Q in the summary. However, we cannot guarantee whether a sentence-extraction method for multi-document summarization chooses any sentences before q_1 from block P_{q_1} for a summary, because the extraction method usually determines a set of sentences within the constraint of pre-defined and fixed summary length² that maximizes information coverage and excludes redundant information.

We define the precedence, $\text{pre}(l)$, of a sentence l as follows,

$$\text{pre}(l) = \frac{1}{|Q|} \sum_{q \in Q} \max_{p \in P_q} \text{sim}(p, l). \quad (18)$$

Here, P_q is the set of sentences preceding the sentence $q \in Q$ in the original document, and $|Q|$ denotes the total number of sentences that we have ordered so far. We calculate $\text{sim}(p, l)$ using cosine similarity as described in Section 3.3. The formalism of precedence proposed in Eq. (18) captures the idea of similarity between preceding information of a sentence in an original document and an extracted sentence that must be ordered in a summary. Because it is sufficient that at least one of the preceding sentences contain the necessary background information, we consider the maximum similarity in Eq. (18). Moreover, if l contains precedence information for many sentences in Q , then it is more preferable to be ordered before

² Length of a summary can be measured either by the number words or the number of sentences. For example, in the multi-document summarization task in the Document Understanding Conferences (DUC), a typical long summary contains ca. 15 sentences.

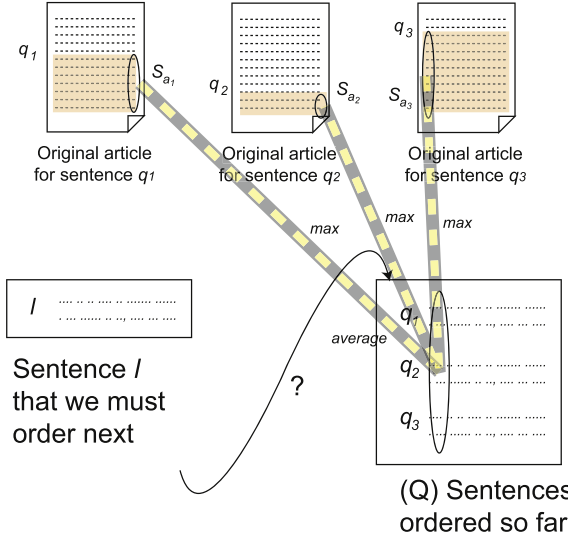


Fig. 5. Succession expert.

Q in the summary. Consequently, we consider all sentences in Q when computing the precedence score of l . To avoid the bias towards large Q segments, we normalize this score by dividing from the total number of sentences in Q (i.e. $|Q|$). Finally, the preference function for the precedence expert can then be written as follows,

$$\text{PREF}_{\text{pre}}(u, v, Q) = \begin{cases} 0.5 & [Q = \emptyset] \vee [\text{pre}(u) = \text{pre}(v)] \\ 1 & [Q \neq \emptyset] \wedge [\text{pre}(u) > \text{pre}(v)] \\ 0 & \text{otherwise} \end{cases} \quad (19)$$

The precedence expert prefers a sentence u to another sentence v purely based on their precedence scores given by Eq. (18). When the precedence scores for the two sentences are equal or if we have not yet ordered any sentences (i.e. $Q = \emptyset$), then we cannot determine the ordering between u and v based on precedence. Consequently, the precedence expert returns a preference of 0.5 under such circumstances.

3.5. Succession expert

In extractive multi-document summarization, sentences that describe a particular event are extracted from a set of source articles. Usually, there exist a logical sequence among the information conveyed in the extracted sentences. For example, in Fig. 2, sentence (a) describes the results of the earthquake described in sentence (b). It is natural to order a sentence that describes the result or an effect of a certain cause after a sentence that describes the cause. Therefore, in Fig. 2, sentence (a) should be ordered after sentence (b) to create a coherent summary. We use the information conveyed in source articles to propose *succession expert* to capture the coverage of information for sentence ordering in multi-document summarization.

Computing the value of the succession expert is illustrated in Fig. 5. Likewise the precedence score defined in Eq. (18), we define the succession score of a sentence l as follows:

$$\text{succ}(l) = \frac{1}{|Q|} \sum_{q \in Q} \max_{s \in S_q} \text{sim}(s, l). \quad (20)$$

Here, we calculate $\text{sim}(s, l)$ using cosine similarity as described in Section 3.3. S_q is the set of sentences that appear after (succeeds) the sentence q in the original document from which q was extracted. Succession score compares each sentence s that appear in S_q against the sentence l that we must order next. If some sentence in S_q contains information similar to that conveyed by l , then l obtains a higher succession score. Because it is sufficient that at least one sentence is similar to l in each succeeding block, we consider the maximum similarity in Eq. (20). Moreover, we divide the sum of similarity scores by the total number of sentences in Q to avoid any bias towards longer Q segments.

Using the succession score defined in Eq. (20), we define the succession expert as follows,

$$\text{PREF}_{\text{succ}}(u, v, Q) = \begin{cases} 0.5 & [Q = \emptyset] \vee [\text{succ}(u) = \text{succ}(v)] \\ 1 & [Q \neq \emptyset] \wedge [\text{succ}(u) > \text{succ}(v)] \\ 0 & \text{otherwise} \end{cases} \quad (21)$$

The succession expert determines the ordering between two sentences u and v purely based on the succession score defined in Eq. (20). If the succession scores of the two sentences being considered are equal or if there are no sentences ordered so far (i.e. $Q = \emptyset$), then the succession expert is undecided of the ordering between u and v , and returns the value 0.5.

3.6. 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 Eq. (2). Section 3.7 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 ordering for a given total preference function is NP-complete [9]. However, Cohen et al. [9] propose a greedy algorithm that approximates the optimal ordering.

Given an unordered set of sentences \mathcal{X} extracted from a set of documents, and total preference function, $\text{PREF}_{\text{total}}(u, v, Q)$, Algorithm 1 computes a total ordering function $\hat{\rho}$ among the extracted sentences \mathcal{X} . Specifically, for a sentence t , the function value $\hat{\rho}(t)$ denotes the ranking score of t . The higher the ranking score of a sentence, the higher that the sentence gets ordered in the summary. It has been shown theoretically that this greedy algorithm always produces an ordering that is within $1/2$ of the ranking score for the optimal ordering [9].

Algorithm 1. Sentence Ordering Algorithm

Input: A set \mathcal{X} of the extracted (unordered) sentences and a total preference function $\text{PREF}_{\text{total}}(u, v, Q)$.
Output: Ranking score $\hat{\rho}(t)$ of each sentence $t \in \mathcal{X}$.

```

1:  $\mathcal{V} = \mathcal{X}$ 
2:  $Q = \emptyset$ 
3: foreach  $v \in \mathcal{V}$  do
4:    $\pi(v) = \sum_{u \in \mathcal{V}} \text{PREF}_{\text{total}}(v, u, Q) - \sum_{u \in \mathcal{V}} \text{PREF}_{\text{total}}(u, v, Q)$ 
5: end for
6: while  $\mathcal{V} \neq \emptyset$  do
7:    $t = \arg \max_{u \in \mathcal{V}} \pi(u)$ 
8:    $\hat{\rho}(t) = |\mathcal{V}|$ 
9:    $\mathcal{V} = \mathcal{V} - \{t\}$ 
10:   $Q = Q + \{t\}$ 
11:  for each  $v \in \mathcal{V}$  do
12:     $\pi(v) = \pi(v) + \text{PREF}_{\text{total}}(t, v, Q) - \text{PREF}_{\text{total}}(v, t, Q)$ 
13:  end for
14: end while
15: return  $\hat{\rho}$ 

```

Algorithm 1 can be understood by thinking of $\text{PREF}_{\text{total}}$ as a directed weighted graph in which, initially, the set of vertices \mathcal{V} is equal to the set of instances \mathcal{X} , and each edge $u \rightarrow v$ has the weight $\text{PREF}_{\text{total}}(u, v, Q)$. We assign to each vertex $v \in \mathcal{V}$ a potential value $\pi(v)$, which is the weighted sum of the outgoing edges minus the weighted sum of the ingoing edges. That is, $\pi(v) = \sum_{u \in \mathcal{V}} \text{PREF}_{\text{total}}(v, u, Q) - \sum_{u \in \mathcal{V}} \text{PREF}_{\text{total}}(u, v, Q)$. The greedy algorithm then picks some node t that has maximum potential, and assigns it a rank by setting $\hat{\rho}(t) = |\mathcal{V}|$, effectively ordering it ahead of all the remaining nodes. This node, together with all incident edges, is then deleted from the graph, and the potential values π of the remaining vertices are updated appropriately. This process is repeated until the graph is empty. Because we remove one node at a time from the graph, the ranks assigned to nodes that will be removed in subsequent iterations will have progressively smaller and smaller ranks.

Couple of important points must be noted in Algorithm 1. First, note that initially the set of ordered sentences so far, Q , is null. Therefore, the topical-closeness, precedence and succession experts will all return a preference score of 0.5 for all pairs of sentences. However, the chronological and probabilistic experts will have values in the full range $[0, 1]$ which enables us to determine the ordering between two sentences even at this initial stage. Second, the set of ordered sentences so far, Q , is in fact a variable that gets updated each time when we remove a sentence t from the set \mathcal{V} (Line 10 in Algorithm 1). This means that the total preference values, $\text{PREF}_{\text{total}}(u, v, Q)$, constantly changes throughout the iteration in the *while* loop that starts at Line 6 in Algorithm 1. Consequently, all experts are re-evaluated and the weighted sums of their individual preferences are computed according to Eq. (2).

It is noteworthy that Algorithm 1 always produces a single total ordering consistent with the preference function values for all pairs of sentences even when the individual partial orderings might imply a cyclic ordering among the sentences. This desirable property holds irrespective of whether the individual preference functions satisfy transitivity. A preference function, PREF , is defined to be transitive if for any three sentences u , v , and w PREF satisfies the condition if $\text{PREF}(u, v) > 0.5$ and $\text{PREF}(v, w) > 0.5$, then $\text{PREF}(u, w) > 0.5$. Fig. 6 illustrates an example where such a situation exists among three sentences u , v , and w . When Algorithm 1 is run on the example shown in Fig. 6, in the *for-loop* in Line 3, the initial ranking scores for u , v

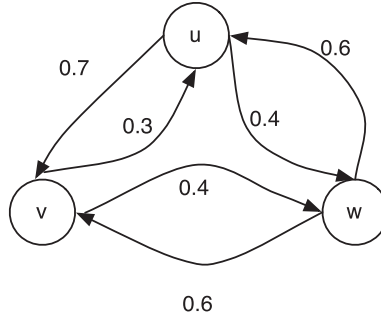


Fig. 6. Generating a total ordering between three sentences u , v , and w using Algorithm 1. The context Q is ignored to simplify the example. Here, $\text{PREF}_{\text{total}}(u, v) = 0.7$, $\text{PREF}_{\text{total}}(v, w) = 0.6$, $\text{PREF}_{\text{total}}(u, w) = 0.4$ imply partial orderings $u \succ v$, $v \succ w$, and $w \succ u$, which produces a cyclic total ordering among u , v , and w . However, Algorithm 1 resolves this issue and produces the total ordering $v \succ w \succ u$. See text for further details.

and w are computed as $\pi(u) = -0.2$, $\pi(v) = 0.2$, and $\pi(w) = 0$. Therefore, v is selected as the first sentence and subsequently removed from the graph. Next, in the *for-loop* in Line 11, ranking scores for the remaining u and w are updated to $\pi(u) = -0.6$ and $\pi(w) = -0.2$. Therefore, w is selected as the next sentence in the summary, which produces the sentence ordering $v \succ w \succ u$.

3.7. Learning algorithm

In [9], Cohen et al. propose a weight allocation algorithm that learns the weights associated with each expert in Eq. (2). We explain this algorithm in the context of our model of five experts as illustrated in Algorithm 2, in which the loss function, $\text{Loss}(\text{PREF}_e, F)$, is defined as follows,

$$\text{Loss}(\text{PREF}_e, F) = \frac{1}{|F|} \sum_{(u,v) \in F} (1 - \text{PREF}_e(u, v, Q)). \quad (22)$$

Algorithm 2. Learning the weights for each preference expert

Input: Rate of learning $\beta \in [0, 1]$, set E of experts e , number of rounds T , initial weights $w_e^1 \in [0, 1]$ for all $e \in E$, s.t.

$$\sum_{e \in E} \tilde{w}_e^1 = 1.$$

Output: Final weight w_e for expert e .

1: **for** $t = 1, 2, \dots, T$ **do**

2: Receive \mathcal{X}^t , the set of sentences to be ordered

3: Compute a total order $\hat{\rho}_t$ using Algorithm 1 that approximates, $\text{PREF}_{\text{total}}^t(u, v, Q) = \sum_{e \in E} w_e^t \text{PREF}_e(u, v, Q)$

4: Order \mathcal{X}^t using $\hat{\rho}_t$

5: Receive feedback F^t from the user

6: Evaluate the losses $\text{Loss}(\text{PREF}_e, F^t)$ for each expert as defined in Eq. (22)

7: Set the new weights $w_e^{t+1} = \frac{w_e^t \beta^{\text{Loss}(\text{PREF}_e, F^t)}}{Z_t}$, where Z_t is a normalization constant, chosen s.t. $\sum_{e \in E} w_e^{t+1} = 1$

8: **end for**

9: **return** Final weights for each expert, w_e^T

In our experiments, we set the learning rate $\beta = 0.5$, and the initial weights for all experts are set equally to $w_e^1 = 0.2$. To explain Eq. (22) let us assume that sentence u appears before sentence v in summary ordered by a human (i.e. training data). Then the expert must return the value 1 for $\text{PREF}_e(u, v, Q)$. 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. Because 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 shown in Line 7 in Algorithm 2. The feedback F is in fact a human-made ordering of the extracted set of sentences \mathcal{X} .

4. Evaluation measures

Evaluating a sentence ordering produced by an algorithm is a difficult task. Semi-automatic evaluation measures that compare a sentence ordering produced by an algorithm against the ordering made by a human annotator for those sentences

have been used in previous work [6,21]. We use three popular semi-automatic evaluation measures that have been proposed in the previous work on sentence ordering: Kendall rank correlation coefficient (τ), Spearman rank correlation coefficient (ρ), and average continuity. Next, we briefly describe those evaluation measures. For a detailed discussion and definitions of those measures refer [6].

4.1. Kendall's rank correlation coefficient

Let $S = s_1 \cdots s_N$ 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,

$$\tau = \frac{4C(\pi, \sigma)}{N(N-1)} - 1. \quad (23)$$

Here, $C(\pi, \sigma)$ is the number of concordant pairs between π and σ (i.e. the number of sentence pairs that have the same relative positions in both π and σ). For example, in Fig. 7 between T_{eval} and T_{ref} , there are six concordant sentence pairs: (a, b) , (a, c) , (a, d) , (b, c) , (b, d) , and (c, d) . These six concordant pairs yield 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 exact reverse of the other.

4.2. Spearman's rank correlation coefficient

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

$$r_s = 1 - \frac{6}{N(N+1)(N-1)} \sum_{i=1}^N (\pi(i) - \sigma(i))^2. \quad (24)$$

Here, $\pi(i)$ and $\sigma(i)$ respectively denote the i th ranked item in π and σ . Spearman's rank correlation coefficient for the example shown in Fig. 7 is 0. Spearman's rank correlation, r_s , ranges from $[-1, 1]$. Similarly to Kendall's τ , the r_s value of 1 is obtained for two identical orderings, and the r_s computed between an ordering and its reverse is -1 .

4.3. Average continuity

A text with sentences arranged in the proper order does not interrupt the process of a human reading from one sentence to the next. Consequently, the quality of a sentence ordering produced by a system can be estimated by the number of continuous sentence segments that it shares with the reference sentence ordering. However, both Spearman and Kendall coefficients do not directly take into consideration this notion of *continuous readability* in a summary. Average Continuity [6,5] is a measure that considers this desirable property in a summary.

For example, in Fig. 7 the sentence ordering produced by the system under evaluation (T_{eval}) has a segment of four sentences $(a \succ b \succ c \succ d)$, which appears exactly in that order in the reference ordering (T_{ref}). Therefore, a human can read this segment without any disfluencies and will find to be coherent.

This is equivalent to measuring a precision of continuous sentences in an ordering against the reference ordering. We define P_n as the precision of n continuous sentences in a sentence ordering as follows:

$$P_n = \frac{m}{N - n + 1}. \quad (25)$$

Here, N is the number of sentences in the reference ordering, n is the length of continuous sentences, and m is the number of continuous sentences that appear in both the evaluation and reference orderings. In Fig. 7, we have two sequences of three continuous sentences (i.e., $(a \succ b \succ c)$ and $(b \succ c \succ d)$). Consequently, the precision of three continuous sentences P_3 is calculated as

$$P_3 = \frac{2}{5 - 3 + 1} = 0.67. \quad (26)$$

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

$$AC = \exp \left(\frac{1}{k-1} \sum_{n=2}^k \log(P_n + \alpha) \right). \quad (27)$$

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

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

Table 1

Correlation between two sets of human-ordered extracts.

Metric	Mean	Std. dev.	Min	Max
Spearman	0.739	0.304	−0.2	1
Kendall	0.694	0.290	0	1
Average continuity	0.401	0.404	0.001	1

Here, k is a parameter to control the range of the logarithmic average, and α is a fixed small value. It prevents the term inside the logarithm from becoming zero in case if P_n is zero. We set $k = 4$ (i.e. more than five continuous sentences are not included for evaluation), and $\alpha = 0.001$. The average continuity is in the range $[0, 1]$. It becomes 0 when the evaluation and reference orderings share no continuous sentences, and 1 when the two orderings are identical.

5. Experiments and results

We evaluated the proposed method using the 3rd Text Summarization Challenge (TSC-3) corpus³. Text Summarization Challenge is a multiple document summarization task organized by the “National Institute of Informatics Test Collection for IR Systems” (NTCIR) project⁴. TSC-3 dataset was introduced in the 4th NTCIR workshop held in June 2–4, 2004. The TSC-3 dataset contains multi-document summaries for 30 news events. The events are selected by the organizers of the TSC task. For each topic, a set of Japanese newspaper articles are selected using some query words. Newspaper articles are selected from Mainichi Shinbun and Yomiuri Shinbun, two popular Japanese newspapers. All newspaper articles in the dataset have their date of publication annotated. Moreover, once an article is published, it is not revised or modified. Therefore, all sentences in an article bare the time stamp of the article.

Although we use Japanese text summaries for experiments, it is noteworthy that there are no fundamental differences between Japanese and English text summarization. In fact, popular summarization algorithms originally designed for English text summarization, such as the maximum marginal relevance (MMR) [7], have been successfully employed to summarize Japanese texts [32].

For each topic, the organizers of the TSC task provide a manually extracted set of sentences. On average, a manually extracted set of sentences for a topic contains 15 sentences. The participants of the workshop are required to run their multi-document summarization systems on newspaper articles selected for each of the 30 topics and submit the results to the workshop organizers. The output of each participating system is compared against the manually extracted set of sentences for each of the topics using precision, recall and F-measure. Essentially, the task evaluated in TSC is sentence extraction for multi-document summarization.

To construct the training data applicable to the proposed method, we asked two human annotators to arrange the extracts. The two human subjects worked independently and arranged sentences extracted for each topic. They were provided with the source documents from which the sentences were extracted. They read the source documents before ordering sentences in order to gain background knowledge on the topic. From this manual ordering process, we obtained $30(\text{topics}) \times 2(\text{humans}) = 60$ sets of ordered extracts. Table 1 shows the agreement of the ordered extracts between the two subjects. The correlation is measured by three metrics: Spearman's rank correlation, Kendall's rank correlation, and average continuity. Definitions of these automatic evaluation measures are described in Section 4. The mean correlation values (0.74 for Spearman's rank correlation and 0.69 for Kendall's rank correlation) indicate a strong agreement in sentence orderings made by the two subjects. In eight out of the 30 extracts, sentence orderings created by the two human subjects were identical.

We apply the leave-one-out method to the proposed method, to produce a set of sentence orderings. Specifically, we select the set of extracted sentences for one topic as test data and the remaining 29 as training data and repeat this process 30 times by selecting a different set at each round. We use the training data to compute the total preference function, $\text{PREF}_{\text{total}}$, using Algorithm 2. Subsequently, the learnt preference function is used to produce a total ordering for the test extract using Algorithm 1. We use the three evaluation measures: Kendall's coefficient, Spearman's coefficient and average continuity to compare the ordering produced by the proposed sentence ordering algorithm against the two human-made orderings for that test extract in our TSC dataset. We report the average results over the two human-made orderings in our experiments.

For comparison purposes, we ordered each extract using four methods: Random Ordering (**RO**), Probabilistic Ordering (**PO**), Chronological Ordering (**CO**) Learned Ordering (**LO**) (the method proposed in this paper) and evaluated those orderings. Next, we describe each of those sentence ordering methods.

Random ordering (RO) is the lowest anchor, in which sentences are arranged randomly. This method acts as a lower-baseline and demonstrates the performance that we would obtain if we randomly order sentences in a summary.

³ <http://ir-www.pi.titech.ac.jp/tsc/tsc3-en.html>

⁴ <http://research.nii.ac.jp/ntcir/index-en.html>

Table 2
Performance of different sentence ordering methods.

Method	Spearman	Kendall	Average continuity
Random ordering (RO)	−0.267	−0.160	0.024
Probabilistic ordering (PO)	0.062	0.040	0.029
Chronological ordering (CO)	0.774	0.735	0.511
Proposed method (LO)	0.783	0.746	0.546

Probabilistic ordering (PO) arranges sentences by using the probabilistic text structuring method proposed by Lapata [21]. We used CaboCha⁵ (a dependency parser for Japanese text) to obtain part-of-speech information and dependency structure of sentences. Using nouns, verbs, and verb-noun dependencies, we trained the language model on a corpus of 100,000 articles selected from Mainichi and Yomiuri newspapers.

Chronological ordering (CO) arranges sentences with the chronology criterion defined in Eq. (3). Sentences are arranged in chronological order of their publication date. Specifically, sentences belonging to articles published earlier are ordered ahead of sentences belonging to articles published later. Among sentences belonging to the same source article, we order them according to the order in which they appear in the original article. Chronological ordering cannot define an order for sentences belonging to articles with identical publication dates/times. Ordering among such sentences are decided randomly.

Learned ordering (LO) is the method proposed in this paper. Specifically, we use Algorithm 2 to learn the weights for each of the experts and then compute the total preference functions as the weighted sum of the individual expert's preference functions according to Eq. (2). Finally, Algorithm 1 is used to produce a total ordering for a set of sentences extracted for a topic.

Note that the experts topical-closeness, precedence and succession cannot produce a total ordering for a set of sentences only by themselves because they cannot determine the ordering between two sentences at the initial stage where we have not ordered any sentences (i.e. when $Q = \emptyset$ all three experts return the value 0.5 for any pair of sentences). Consequently, those experts have not been included as individual ranking algorithms in the above list of baselines. Of course, they are used in the proposed method (**LO**) both during training and ordering as described in the paper.

We evaluate each of the above mentioned methods using Kendall coefficient, Spearman coefficient and average continuity. Experimental results are shown in Table 2. From Table 2, we see that the proposed method (**LO**) reports the best results among the four methods compared in the table according to all evaluation measures. We performed an analysis of variance (ANOVA) test followed up with Tukey's honest significance of differences (HSDs) test [39] to evaluate the statistical significance of the results obtained. Our statistical significance tests reveal that the improvement of the proposed method (**LO**) over all the other methods compared in Table 2 are statistically significant under the confidence level of 0.05. It is noteworthy that both randomly ordering sentences (**RO**) and probabilistic ordering (**PO**) result in very poor performances. In particular, we found that data sparseness is a major problem for the probabilistic ordering method even though we used smoothing methods as described in Section 3.2. One reason for the data sparseness is that our evaluation dataset contains newspaper articles that describe novel events that have not been reported in the past. Therefore, adjacent sentence pairs in past newspaper articles rarely contain two words that appear in adjacent sentences in the extracts that must be ordered. On the other hand, chronological ordering (**CO**) works surprisingly well despite its simplicity. In particular, newspaper articles have a tendency to present information in a chronological fashion, elaborating past events using new information. The appropriateness of chronological ordering as a method to order sentences for multi-document news summarization systems have been reported also in previous work [33,2]. The ability of the proposed method to significantly improve over chronological ordering by taking into consideration other clues such as precedence and succession relations can be seen as an important contribution of our work.

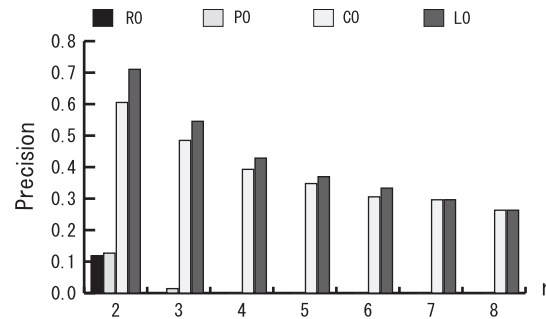
Table 3 shows the weights learnt by Algorithm 2 for the different experts discussed in the paper. Because the final total preference function used for ordering (Eq. (2)) is simply the weighted linear combinations of the individual preference functions corresponding to each expert, the weight learnt for a particular expert indicates the influence it imparts on the overall sentence ordering algorithm (Algorithm 1). From Table 3, we see that succession, chronological and precedent experts have significant contributions to the total preference functions, whereas the contributions of probabilistic and topical-closeness experts are negligible. The high weight assigned to the chronological expert is in agreement with the better performance we observed in Table 2 for the chronological ordering (**CO**) method and indicates the importance of chronological information for sentence ordering in multi-document news summarization tasks. It is interesting to note that succession expert reports the highest weight among all five experts included in our proposed method. Typically, news articles follow a logical order of events where succession relations are often satisfied. The high weight learnt for the succession expert is a consequence of

⁵ <http://chasen.org/taku/software/cabocha/>.

Table 3

Weights learned for different experts by Algorithm 2.

Expert	Chronological	Probabilistic	Topical-closeness	Precedent	Succession
Weight	0.327947	0.000039	0.016287	0.196562	0.444102

**Fig. 8.** Precision P_n vs. the length n of continuous sentence segments (refer Eq. (25)).

this phenomenon. Future work in sentence ordering using other types of texts, other than newspaper articles, will reveal whether this phenomenon is universal or unique to this genre. Although both precedence and succession experts model similar types of relations in multi-document summaries, we see that succession relations are more useful in determining the order among a set of extracted sentences. The low weight learnt for the probabilistic expert is due to the data sparseness issues that were already discussed under Table 2. Extractive summarization systems attempt to maximize the diversity of a summary by including numerous sub-topics discussed in a set of source documents in the summary. Consequently, not all sentences in a summary are closely related to the main topic of the summary, but cover other sub-topics. The relatively lower weight learnt for the topical-closeness expert in comparison to precedent and succession experts can be attributable to this nature of sentence selection in multi-document summarization.

The number of continuous sentences that appear both in an ordering produced by an algorithm as well as in an ordering made by a human annotator for a set of sentences is an indicator of the readability of a sentence ordering produced by the algorithm. Average continuity measure captures this notion of readability. To further investigate how each of the above-mentioned sentence ordering methods perform in terms of *precision*, we plot the precision scores, P_n (Eq. (25)) against the length n of continuous text segments (length is measured by the number of sentences that appear in a continuous segment of sentences) in Fig. 8. According to Fig. 8, for lengths up to six sentences, the proposed method (LO) has the highest precision among the different sentence ordering methods compared. The probabilistic ordering (PO) does not possess continuous segments of sentences with length more than two. Note that larger continuous segments are rare and as a result precision decreases with the value of n for all sentence ordering methods.

In Table 4, we compare the proposed method against two previously proposed sentence ordering methods: Okazaki et al. [33] and Bollegala et al. [6]. Both those methods are evaluated on the same dataset as the proposed method and can be directly compared against our results. Among all the methods compared in Table 4, Okazaki et al. [33] reports the best performance in all three evaluation measures. The second best set of results is obtained by the proposed method. However, a paired *t*-test performed using the Kendall coefficients shows that the difference of performance between Okazaki et al. [33] and our proposed method is statistically insignificant under the 0.05 critical level. Therefore, we conclude that our method is statistically comparable to the state-of-the-art approach by Okazaki et al. [33].

In addition to using the semi-automatic evaluation measures described in Section 4 for evaluating a sentence ordering method, we also conduct a subjective evaluation to further compare the different sentence ordering methods. We asked three human judges to rate sentence orderings according to the following criteria⁶.

Perfect: A *perfect* summary is a text that we cannot improve any further by re-ordering.

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 its 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.

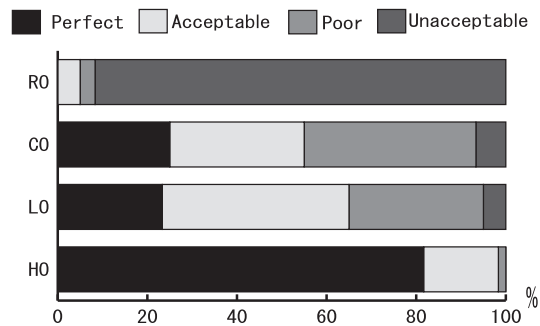
Unacceptable: An *unacceptable* summary is one that leaves much to be improved and requires overall restructuring rather than partial revision.

⁶ The human judges that participated in this evaluation are different from the two annotators that created the two sets of reference summaries. All three judges are native Japanese speakers and graduate school students, majoring in information engineering.

Table 4

Comparison against previously proposed sentence ordering methods.

Method	Spearman	Kendall	Average continuity
Proposed method (LO)	0.783	0.746	0.546
Okazaki et al. [33]	0.843	0.792	0.606
Bollegala et al. [6]	0.603	0.612	0.459

**Fig. 9.** Human evaluation.

- (i). Hurricane Gilbert, one of the strongest storms ever, slammed into the Yucatan Peninsula Wednesday and leveled thatched homes, tore off roofs, uprooted trees and cut off the Caribbean resorts of Cancun and Cozumel.
- (ii). Tropical Storm Gilbert formed in the eastern Caribbean and strengthened into a hurricane Saturday night.
- (iii). Gilbert reached Jamaica after skirting southern Puerto Rico, Haiti and the Dominican Republic.
- (iv). The Mexican National Weather Service reported winds gusting as high as 218 mph earlier Wednesday with sustained winds of 179 mph.
- (v). More than 120,000 people on the northeast Yucatan coast were evacuated, the Yucatan state government said.
- (vi). Shelters had little or no food, water or blankets and power was out.
- (vii). The storm killed 19 people in Jamaica and five in the Dominican Republic before moving west to Mexico.
- (viii). Prime Minister Edward Seaga of Jamaica said Wednesday the storm destroyed an estimated 100,000 of Jamaica's 500,000 homes when it throttled the island Monday.
- (ix). The National Hurricane Center said a hurricane watch was in effect on the Texas coast from Brownsville to Port Arthur and along the coast of northeast Mexico from Tampico north.
- (x). The National Hurricane Center said Gilbert was the most intense storm on record in terms of barometric pressure.

Fig. 10. An example of a *perfect* grade summary.

To avoid any disturbance in rating, we inform the judges that the summaries were made from the same set of extracted sentences, and that only the ordering of sentences is different. Furthermore, the judges were given access to the source documents for each summary. Fig. 10 shows a summary that obtained a *perfect* grade. The ordering 1–4–5–6–7–8–2–3–9–10 was assigned an *acceptable* grade, whereas 4–5–6–7–1–2–3–8–9–10 was given a *poor* grade. A random ordering of the ten sentences 4–7–2–10–8–3–1–5–6–9 received an *unacceptable* grade.

The results of the human evaluation of the summaries is shown in Fig. 9. For each of the sentence ordering methods shown in Fig. 9, there are 90 (30 summaries \times 3 human judges) ratings provided by the human judges. 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 $W = 0.937$. For each sentence ordering method, we aggregate the ratings for each of the four grades individually and report the percentage in Fig. 9. From Fig. 9, we see that most of the randomly ordered summaries (**RO**) are *unacceptable*. Although both chronological ordering (**CO**) and the proposed method (**LO**) have the same number of *perfect*

summaries, the acceptable to poor ratio is better in **LO**. Over 60% of summaries ordered using **LO** are either *perfect* or *acceptable*.

6. Conclusions and future work

In this paper, we studied the problem of ordering a set of extracted sentences in a multi-document summarization setting to create a coherent summary. We formalized numerous previously proposed ideas for ordering a set of sentences into experts that express preferences for ordering one sentence ahead of another in a summary. Specifically, we proposed five ordering experts: chronological expert, probabilistic expert, topical-closeness expert, precedence expert, and succession expert. We then learnt the weighted linear combination of those experts using a hedge regression algorithm. We use a set of summaries ordered by human annotators as training data. We proposed a pairwise greedy ordering algorithm that has good approximation properties and time complexities to avoid the combinatorial searching frequently associated with total ordering problems. The proposed method significantly outperformed numerous baselines and previously proposed sentence ordering methods on a publicly available dataset for multi-document summarization.

There are several natural future research directions to our current work. First, the list of ordering experts that we presented in this paper is by no means exhaustive. There are numerous other signals that one can model as ordering experts. For example, textual entailment relations [8] between two sentences can be modeled as an ordering expert. If a sentence T entails another sentence H , then it is a strong signal that we must order H after T in a summary. Second, there are numerous other classification algorithms that can be used to infer the final ordering rule. Fuzzy rules [41,40] are particularly suitable for this purpose because they can incorporate soft rules with confidence scores. Third, the sentence ordering problem is not limited to text summarization but omnipresent in numerous other natural language generation tasks. It remains to be tested whether the proposed approach is sufficient to order sentences in other natural language generation tasks such as warfighting games based on linguistic geometry [38].

References

- [1] P. Achananuparp, X. Hu, X. Shen, The evaluation of sentence similarity measures, in: 10th International Conference on Data Warehousing and Knowledge Discovery, 2008.
- [2] R. Barzilay, N. Elhadad, K. McKeown, Inferring strategies for sentence ordering in multidocument news summarization, *Journal of Artificial Intelligence Research* 17 (2002) 35–55.
- [3] R. Barzilay, L. Lee, Catching the drift: probabilistic content models, with applications to generation and summarization, in: HLT-NAACL 2004: Proceedings of the Main Conference, 2004.
- [4] D. Bollegala, N. Noman, H. Iba, Rankde: learning a ranking function for information retrieval using differential evolution, in: GECCO'11, 2011.
- [5] D. Bollegala, N. Okazaki, M. Ishizuka, A bottom-up approach to sentence ordering for multi-document summarization, in: COLING/ACL '06, 2006.
- [6] D. Bollegala, N. Okazaki, M. Ishizuka, A bottom-up approach to sentence ordering for multi-document summarization, *Information Processing and Management* 46 (1) (2010) 89–109.
- [7] J. Carbonell, J. Goldstein, The use of mmr, diversity-based reranking for reordering documents and producing summaries, in: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1998.
- [8] J.J. Castillo, A wordnet-based semantic approach to textual entailment and cross-lingual textual entailment, *International Journal of Machine Learning and Cybernetics* 2 (2011) 177–189.
- [9] W.W. Cohen, R.E. Schapire, Y. Singer, Learning to order things, *Journal of Artificial Intelligence Research* 10 (1999) 243–270.
- [10] P. Duboue, K. McKeown, Empirically estimating order constraints for content planning in generation, in: Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL'01), 2001.
- [11] P. Duboue, K. McKeown, Content planner construction via evolutionary algorithms and a corpus-based fitness function, in: Proceedings of the second International Natural Language Generation Conference (INLG'02), 2002.
- [12] N. Elhadad, K. McKeown, Towards generating patient specific summaries of medical articles, in: Proceedings of the NAACL 2001 Workshop on Automatic Summarization, 2001.
- [13] A. Feng, J. Allan, Finding and linking incidents in news, in: CIKM'07, 2007.
- [14] E. Filatova, E. Hovy, Assigning time-stamps to event-clauses, in: Proceedings of the 2001 ACL Workshop on Temporal and Spatial Information Processing, 2001.
- [15] Y. Freund, R. Iyer, R.E. Schapire, Y. Singer, An efficient boosting algorithm for combining preferences, *Journal of Machine Learning Research* 4 (2003) 933–969.
- [16] I. Good, The population frequencies of species and the estimation of population parameters, *Biometrika* 40 (1953) 237–264.
- [17] P.D. Ji, S. Pulman, Sentence ordering with manifold-based classification in multi-document summarization, in: Proceedings of Empirical Methods in Natural Language Processing, 2006.
- [18] N. Karamanis, H.M. Manurung, Stochastic text structuring using the principle of continuity, in: Proceedings of the second International Natural Language Generation Conference (INLG'02), 2002.
- [19] S.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) (1987) 400–401.
- [20] M.G. Kendall, A new measure of rank correlation, *Biometrika* 30 (1938) 81–93.
- [21] M. Lapata, Probabilistic text structuring: experiments with sentence ordering, in: Proceedings of the Annual Meeting of ACL, 2003, 2003, pp. 545–52.
- [22] M. Lapata, Automatic evaluation of information ordering, *Computational Linguistics* 32 (4) (2006).
- [23] M. Lapata, A. Lascarides, Learning sentence-internal temporal relations, *Journal of Artificial Intelligence Research* 27 (2006) 85–117.
- [24] C. Lin, E. Hovy, Neats: a multidocument summarizer, in: Proceedings of the Document Understanding Workshop (DUC).
- [25] I. Mani, M.T. Maybury (Eds.), *Advances in Automatic Text Summarization*, The MIT Press, 2001.
- [26] I. Mani, B. Schiffman, J. Zhang, Inferring temporal ordering of events in news, in: Proceedings of North American Chapter of the ACL on Human Language Technology (HLT-NAACL 2003), 2003.
- [27] I. Mani, G. Wilson, Robust temporal processing of news, in: Proceedings of the 38th Annual Meeting of ACL (ACL 2000), 2000.
- [28] W. Mann, S. Thompson, Rhetorical structure theory: toward a functional theory of text organization, *Text* 8 (3) (1988) 243–281.
- [29] C.D. Manning, H. Schütze, *Foundations of Statistical Natural Language Processing*, second ed., The MIT Press, Cambridge, Massachusetts London, England, 2002.

- [30] K. McKeown, J. Klavans, V. Hatzivassiloglou, R. Barzilay, E. Eskin, Towards multidocument summarization by reformulation: progress and prospects, *AAAI/IAAI* (1999) 453–460.
- [31] D. Metzler, S.T. Dumais, C. Meek, Similarity measures for short segments of text, in: *ECIR'07*, 2007.
- [32] T. Mori, T. Sasaki, Information gain ratio meets maximal marginal relevance – a method of summarization for multiple documents, in: *Proceedings of NTCIR Workshop 3 Meeting – Part V: Text Summarization, Challenge 2 (TSC2)*, 2002.
- [33] N. Okazaki, Y. Matsuo, M. Ishizuka, Improving chronological sentence ordering by precedence relation, in: *Proceedings of 20th International Conference on Computational Linguistics (COLING 04)*, 2004.
- [34] D. Ó Séaghdha, A. Korhonen, Probabilistic models of similarity in syntactic context, in: *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP-11)*, Edinburgh, UK, 2011.
- [35] T. Qin, T.-Y. Liu, J. Xu, W. Xiong, H. Li, Letor: A Benchmark Collection for Learning to Rank for Information Retrieval, Tech. Rep, Microsoft Research Asia, 2007.
- [36] D.R. Radev, K. McKeown, Generating natural language summaries from multiple on-line sources, *Computational Linguistics* 24 (3) (1999) 469–500.
- [37] E. Reiter, R. Dale, *Building Natural Language Generation Systems*, Cambridge University Press, 2000.
- [38] B. Stilman, V. Yakhnis, O. Umanskiy, The primary language of ancient battles, *International Journal of Machine Learning and Cybernetics* 2 (2011) 157–176.
- [39] J.W. Tukey, *Exploratory Data Analysis*, Addison-Wesley, 1977.
- [40] X.-Z. Wang, C.-R. Dong, Improving generalization of fuzzy if – then rules by maximizing fuzzy entropy, *IEEE Transactions on Fuzzy Systems* 17 (3) (2009) 556–567.
- [41] X.-Z. Wang, L.-C. Dong, J.-H. Yan, Maximum ambiguity based sample selection in fuzzy decision tree induction, *IEEE Transactions on Knowledge and Data Engineering* (DOI:10.1109/TKDE.2011.67).
- [42] F. Xia, T.-Y. Liu, J. Wang, W. Zhang, H. Li, Listwise approach to learning to rank: theory and algorithm, in: *ICML 2008*, 2008.
- [43] J. Zhang, Y. Sun, H. Wang, Y. He, Calculating statistical similarity between sentences, *Journal of Convergence Information Technology* 6 (2) (2011) 22–34.