Multi-topical Discussion Summarization Using Structured Lexical Chains and Cue Words

Jun Hatori[1], Akiko Murakami[2,3], and Jun'ichi Tsujii[1,3,4,5]

[1] Graduate School of Information Science and Technology, University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan
{hatori,tsujii}@is.s.u-tokyo.ac.jp
[2] IBM Research – Tokyo
1623-14 Shimotsuruma, Yamato, Kanagawa, Japan
akikom@jp.ibm.com
[3] Graduate School of Interdisciplinary Information Studies, University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan
[4] School of Computer Science, University of Manchester
131 Princess Street, Manchester, M1 7DN, UK
[5] National Centre for Text Mining (NaCTeM), UK
131 Princess Street, Manchester, M1 7DN, UK

Abstract. We propose a method to summarize threaded, multi-topical texts automatically, particularly online discussions and e-mail conversations. These corpora have a so-called reply-to structure among the posts, where multiple topics are discussed simultaneously with a certain level of continuity, although each post is typically short. We specifically focus on the multi-topical aspect of the corpora, and propose the use of two linguistically motivated features: lexical chains and cue words, which capture the topics and topic structure. Particularly, we introduce the *structured lexical chain*, which is a combination of traditional lexical chains with the thread structure. In experiments, we show the effectiveness of these features on the Innovation Jam 2008 Corpus and the BC3 Mailing List Corpus based on two task settings: key-sentence and keyword extraction. We also present detailed analysis of the result with some intuitive examples.

1 Introduction

Online discussion has become a popular tool for collaboration among people as they discuss various topics online. However, with its increasing popularity, problems have arisen with information overload, which makes it difficult for people to catch up with up-to-date topics and central points of the discussion. Particularly, if organizers intend to draw out useful findings from the whole discussion, they often encounter a problem with obtaining the big picture of the content that is distributed among a large number of posts. Therefore, great demand exists for systems that provide users with an overview of the discussion.

Posts in an online discussion are typically organized in either a sequential or a tree-structured thread. Although the former has simpler structure, the latter allows division of many topics into smaller branches. For this reason, the tree-structured thread has been

A. Gelbukh (Ed.): CICLing 2011, Part II, LNCS 6609, pp. 313–327, 2011.

adopted in many large discussion fora (e.g. Slashdot[1]) as well as in internal discussions in enterprises. Fig. 1 is an excerpt of an online discussion thread in the IBM Corporation, where the main topic of the thread, "leave pool," is branched into two subtopics, "leave accumulation" and "maternity leave," which are clearly identified in the two distinct branches in the thread tree. In larger threads, it is even common that multiple topics are discussed alternately in the same sequential branch. This multi-topicality of texts is a challenge for both parcitipants and systems to comprehend the whole content of the discussion. Therefore, our approach to the overview of the discussion is twofold: we first try to recognize the topics discussed (*topic extraction*), and then incorporate the information of the topics into the task of *key-sentence extraction* (extractive summarization).

To address the problem of *multi-topicality* of texts, some researches have introduced *lexical chains* for the task of summarization (e.g. [1]). Lexical chains are chains of semantically related words; each is considered to render a topic in the document. Recently, the lexical chains have also been successfully applied to the task of multi-document summarization [2,3]. However, to the extent of our knowledge, they have never been applied to threaded texts such as online discussions and e-mail conversations.

To apply the lexical chains to summarization of online discussions, we focus on the use of the thread structure, by which we can infer the flow of the arguments and topics. In Fig. 1, we can observe that the chains of semantically related words, such as "leave (pool)," "accumulate(d)," and "maternity, paternity" characterize the topics in the thread, capturing the cohesive property of topics in the thread structure. This motivates the use of lexical chains with the thread structure: we introduce the *structured lexical chain*s, by which we can combine the traditional lexical chains with a newly proposed scoring scheme that evaluates the importance of each sentence in the context of the thread structure.

Another characteristic of discussion corpora is that the writers tend to use typical expressions to clarify their statements in short posts. In Fig. 1, many underlined key sentences include (italicized) characteristic expressions that typically appear in sentences stating the writer's main opinion or proposal. For example, auxiliary verbs such as "should" and "could," and verbs such as "suggest" and "think" are examples of these expressions. These are considered to be examples of *cue words*, which have been discussed in the linguistic literature [4]. We propose to model these expressions explicitly with scores reflecting how strongly they contribute for a sentence to be a key sentence. In experiments, we explore a set of cue words that are effective for this task in both manual and automatic ways, and evaluate them using the proposed summarization model.

Because numerous online discussions exist with different domains and characteristics, it is not practical to construct a supervised system. For that reason, we construct an unsupervised model based on the graph-based multi-document summarization model presented by [5]. We then further extend this model to incorporate the structured lexical chains and cue words. The proposed model works with minimal supervision; we show that the almost-unsupervised, graph-based model with a few manually selected cue words works comparably with the supervised counterpart.

[1] http://slashdot.org

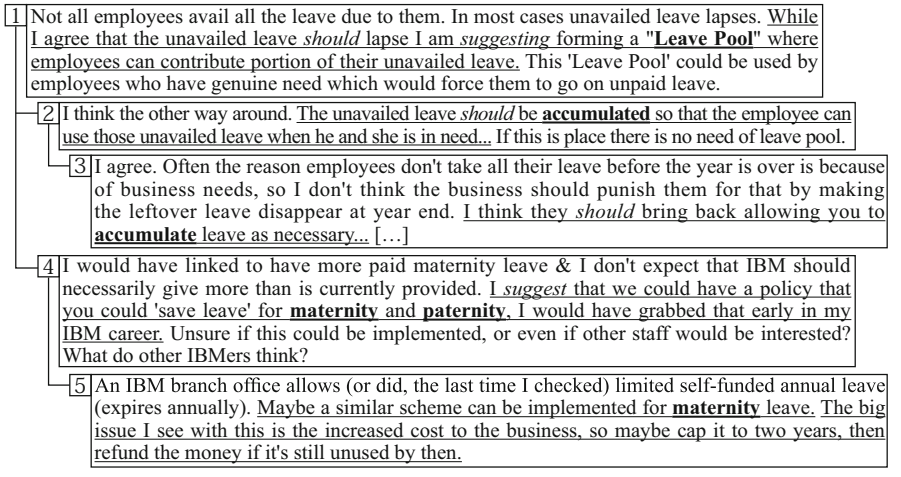

Fig. 1. A thread example from the Innovation Jam 2008 Corpus

As datasets, we mainly address the IBM's internal discussion, the "Innovation Jam 2008 Corpus" (hereinafter called "I-Jam 2008 Corpus"). We manually annotated key sentences and topics information on this corpus, and then used them to evaluate our model. To validate and compare the results, we also perform experiments on the BC3 Corpus, which is a collection of mailing list threads and is expected to share the multi-topical nature and conciseness of the expression with the I-Jam 2008 Corpus.

Here are several key terms that will be used throughout this paper.

forum: Discussion board with a specific theme for discussion.

thread: Series of posts which are mutually connected by reply-to relations.

post: Message written by a participant.

In what follows, in Section 2.2, we first introduce related works. We describe our model in detail in Section 3. We present our experimental settings and results in Section 4, and our conclusions in Section 5.

2 Background

2.1 Corpora

Innovation Jam (I-Jam) 2008 Corpus. The Innovation Jam (I-Jam) 2008 Corpus is a collection of online discussion called "Innovation Jam 2008," which was held by the IBM Corporation in 2008. Up to now, the company has held several sessions of a short-term, intensive discussion called "Jam." The Innovation Jam is one of those sessions, and is intended for not only IBM employees, but also for the customers and families of the employees and customers. In the Innovation Jam 2008, the participants discussed various topics related to the company's future plan; the session attracted 29,498 posts by 8,937 participants within five days. Such a relatively concentrated nature of the discussion naturally encouraged people to use simple and concise expression, which are

even clarified using topical words and cue words. Also, the I-Jam Corpus is a *brainstorming*-type discussion in which the participants discuss various topics from various viewpoints in an attempt to obtain novel and inspiring ideas. This contrasts starkly with standard discussion corpora that have been investigated to date [6,7], which include *question–answer* and *problem–solution* type discussion. Hence, we believe that the targeted corpus of our research is also interesting to the community.

BC3 Corpus. As another corpus used for experiment, we used the BC3 Corpus [8], which is already annotated with extractive summaries. This is a collection of e-mail posts in the W3C Corpus. The annotation is done by three annotators, with a kappa agreement of 0.50 for the extractive summary sentences. The BC3 Corpus comprises 41 threads, which include 200 documents.

2.2 Related Work

Our method for extracting key sentences and topics is closely related to extractive summarization and keyword extraction research, particularly that for web texts, such as blogs, mailing lists, and discussion fora. The primary characteristics of these corpora are that the threads are updated dynamically as the discussion proceeds; also, they consist of documents linked by reply-to relations.

To reduce the number of documents that must be read to comprehend the ongoing discussion, some researchers (e.g. [9]) have emphasized evaluation of the importance of each document. Other researchers directly examined the summarization of threads: to date, research efforts have investigated blogs [10,11], e-mails [12,7,13], and discussion fora [11]. However, these studies have not explicitly emphasized the multi-topical aspects of the corpora.

Some models exploit corpus-specific reply-to structures. [14] exploits the thread structure to summarize mailing lists. In this method, the ancestral messages of a post are regarded as its context and are used in the summarization process. For summarization of a discussion thread, [15] used the thread structure indirectly to find successive appearances of the same *clue word*s. In this context, our method is more advanced in that we use the structural information to recognize *subchain*s of a lexical chain, with novel ideas of *subchain*s and *locality*, which we describe in Section 3.3 in detail.

Although the clue words are merely repetitions of the same word, a *lexical chain* considers semantically related words as well. Several researchers [16,17] have used this approach for the summarization of single documents. More recently, the lexical chain has also been applied to the multi-document summarization [2,3]. For keyword extraction, [18] reported success in applying lexical chains to topic extraction from a single document. They considered strong lexical chains to be prominent topics of a document. However, lexical chains were used without consideration of the structural information. Consequently, they have never been applied to the summarization of e-mail conversations nor online discussions. The *structured lexical chain*, which we propose in this paper, is the first method to combine lexical chains with a thread structure.

3 Model Description

In this section, we describe our model for the key-sentence and topic extraction task. We first describe a graph-based summarization model by [19] in Section 3.1; then introduce two features we propose: cue words and lexical chains, in Section 3.2 and Section 3.3, respectively. Finally, we briefly describe a supervised model that we use for comparison with the proposed (almost-)unsupervised model.

3.1 Graph-Based Summarization Model

First of all, let us briefly describe the graph-based models proposed by [19] and its extension by [5]. In these models, we first construct a graph, where each node represents a sentence and each vertex represents a word shared by two sentences. By calculating the PageRank [20] for the vertices in the graph, one can find which sentence is most likely to be a key sentence, based on the assumption that a sentence that includes more information shared by other *important* sentences is important. Despite the simple framework, their model achieved scores comparable to those of state-of-the-art models.

The extension by [5] is to incorporate the importance of documents and sentence–document correlations as modifications to the edge weights. Because the incorporation of the importance of the documents did not improve the performance in our preliminary experiment, we only used the sentence–document correlation in our model. The resulting PageRank value is given as

$$R(s) = (1-d) + d \sum_{s' \in \mathcal{S}} \frac{f(s,s')R(s')}{\sum_{s'' \in \mathcal{S}} f(s,s'')} \tag{1}$$

$$f(s,s') = \mathrm{Sim}(s,s') \cdot \frac{1}{2}(\mathrm{Imp}(s) + \mathrm{Imp}(s')) \tag{2}$$

$$\mathrm{Imp}(s) = \mathrm{Sim}(s, doc(s)) \ , \tag{3}$$

where we set $d = 0.5$ based on our preliminary experiment on the development set.

3.2 Cue Word

A cue word [4] is a characteristic expression that affects the extract-worthiness of a sentence. It is either a *bonus* word or a *stigma* word, which is respectively the indicator of an important or an unimportant sentence. Words and phrases such as 'important,' 'should,' and 'I propose' are examples of bonus words (phrases), whereas those such as 'for instance' and 'example' are considered to be stigma words. In the graph-based model, we incorporated information from cue words as a modification to the edge weights as

$$\mathrm{Imp}(s) = \mathrm{Sim}(s, doc(s)) \cdot \prod_{c \in \mathcal{W}(s)} \mathrm{CueScore}(c) \ , \tag{4}$$

where $\mathcal{W}(s)$ denotes the set of cue words in sentence s.

3.3 Structured Lexical Chain

A *lexical chain* [1] is a sequence of semantically related words in a text. As described by [18], we assume that each lexical chain characterizes a topic of the thread. Because it captures a considerable part of the lexical cohesiveness in natural language texts and is easily incorporated, it has been widely used for various tasks including text summarization [16] and key phrase extraction [18].

We extended this by incorporating the information of thread structure, thereby introducing the idea of *structured lexical chain*s. In constructing a structured chain, we first segment each chain into local substructures called *subchain*s, and score each subchain with respect to the strength of the local structure. We describe this newly proposed method for constructing and scoring the subchains in Section 3.3 and Section 3.3.

Considering the contribution of the lexical chains, the score of an edge connecting sentences s and s' is modified as

$$f(s,s') = \mathrm{Sim}(s,s')\mathrm{Rel}(s,s') \cdot \frac{1}{2}(\mathrm{Imp}(s) + \mathrm{Imp}(s')) + \lambda \sum_{c \in \mathcal{LC}(s,s')} \mathrm{Score}(c) \quad , (5)$$

where $\mathcal{LC}(s,s')$ is the set of lexical chains that includes words in both sentences s and s'. Based on results of a preliminary experiment, we set $\lambda = 2.5$.

Chain construction. First, we describe a general method for constructing lexical chains that is also applicable for constructing the structured lexical chains. [21] proposed an efficient linear-time algorithm for recognizing lexical chains, which performs simple word sense disambiguation simultaneously. Their method comprises two steps: the first calculates the scores of all possible chains with no sense disambiguation; the second removes each word instance from any chain in which it does not maximally contribute in terms of the relation scores (i.e. simple word sense disambiguation). As semantic relations, we used synonym, hypernym, hyponym, and sibling relations in WordNet [22] following their approach; we additionally exploited holonym, antonym, and nominalization links. The weights are modified slightly from their original work: 0.95 for nominalizations, 0.9 for antonyms, 0.5 for siblings, 0.3 for hypernyms and hyponyms, and 0.2 for holonyms, which are set on the development set.

Subchain. We introduced the concept of *subchain*s, which are maximal local structures of a lexical chain. Consequently, one lexical chain consists of one or more subchains. A subchain for a lexical chain c is a local subgraph of documents, all of which include any element in the chain c, as shown in Fig. 2. It is constructed as follows. We connect, with a *direct edge*, each pair of directly connected documents that both include one or more words in the chain c. To increase the coverage, we also connect with an *indirect edge* each pair of documents that is connected via one intervening document node in the thread structure. A subchain is merged with other subchains until no more subchains can be merged via a direct or indirect edge. Eventually, the example in Fig. 2 consists of two subchains.

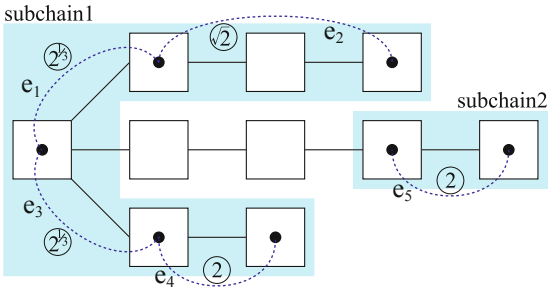

Fig. 2. Illustration of lexical chain scoring, where each box with a dot denotes a post that includes a word in the target chain

Scoring. After constructing the subchains for a lexical chain c, the chain score $\mathrm{Score}(c)$ is calculated as

$$\mathrm{Score}(c) = \mathrm{Strength}(c) + \mathrm{Locality}(c) \tag{6}$$

$$\mathrm{Locality}(c) = \ln \sum_{c' \in \mathcal{SC}(c)} \prod_{e \in \mathcal{E}(c')} \mathrm{EdgeScore}(e) \ , \tag{7}$$

where $\mathcal{SC}(c)$ stands for the set of subchains in the lexical chain c, $\mathcal{E}(c')$ signifies the set of (direct and indirect) edges in the subchain c', $n(e)$ denotes the number of children of the document that includes the first word (e_{start}) of the edge e, and $\mathrm{EdgeScore}(e)$ represents $2^{\frac{1}{n(e)}}$ if e is a direct edge and $2^{\frac{1}{2n(e)}}$ if e is an indirect edge. Here, we introduced *locality*, which measures the strength of the locally connected structure of the lexical chain. Because an actively discussed topic is more likely to have a more locally-concentrated structure, this metric helps differentiate a strong, topical lexical chain from unimportant chains (or chains with frequent but general words). The chain strength $\mathrm{Strength}(c)$ is calculated similarly as [21]. Fig. 2 portrays locality calculation. For this thread structure, the chain locality is calculated as $\ln\left[(2^{\frac{1}{3}} \cdot \sqrt{2}) \cdot (2^{\frac{1}{3}} \cdot 2) + 2\right]$.

3.4 Regression-Based Summarization Model

We also construct a supervised regression-based summarization model based on the approach by [23]. In the experiment on the I-Jam 2008 Corpus, this is used to see the degree to which supervised training with lexical features can further improve the model over that with manually chosen cue words. On the other hand, on the BC3 Corpus, this framework is used as a main framework for the experiment because we need to evaluate our model on the extractive summaries in the corpus. Because the task of summarization requires the generation of fixed-length summaries, [23] mentioned that the regression-based approach is more suitable for this task than other frameworks. We use the support vector regression (SVR) classifier and Bagging with the RSTTree classifier, by which they reported superior results among several machine learning techniques.

For summarization on the BC3 Corpus, we used the same feature set as [23], including the position of the sentence and the post, number of words and recipients, and the

average and sum of the tf-idf vector elements. For summarization on the I-Jam Corpus, we used lexical features including bag-of-words (unigrams, bigrams, and trigrams) in addition to the PageRank scores generated using the graph-based model.

4 Experiment

In this section, we describe our experimental settings and results. For proprocessing, we first performed a standard step including the lemmatization and part-of-speech tagging of words. We implemented the models described in Section 3 in Java using two machine learning libraries, Amis [24] and Weka[2].

4.1 Task Setting

Our models are evaluated on two task settings: key-sentence extraction (extractive summarization) and keyword extraction.

A *key sentence* is defined as a sentence that describes or which is most closely related to the main argument of a post. Intuitively, an important proposal or a new idea, which is most likely to be included in the summary of the whole thread, shall be a key sentence. We did not create human-annotated summaries because the scarcity of annotators (only two) complicates the creation of summaries with reasonable agreement.

A *topic* is a subject or theme that is discussed in a thread. Because each thread is allotted a theme for discussion, these topics are considered as subtopics related to the main theme of the thread. We define each topic using a set of key words or phrases, as exemplified in Table 1.

Table 1. Examples of annotated topics and their definitions with key words and phrases

Topic	Definition
Desalination of sea water	desalination, desalinate
Water leakage from supply piping	dispersion, leak, leakage
Semantic web	semantic web, semiotic web

4.2 Annotation

Because no human-annotated data for the I-Jam 2008 Corpus were available, we first created an annotated corpus from the corpus.

First, we annotated key sentence(s) of each post. We annotated at least one key sentence to each post. Although summarization and key-sentence extraction are fundamentally different tasks, a key-sentence extraction system can be evaluated by considering that the collection of extracted sentences comprises the (extractive) summary. We also noticed some cases in which multiple key sentences should be annotated. In this case, the annotators are allowed to annotate multiple key sentences when they think it really is necessary (e.g., cases in which multiple major arguments exist).

[2] http://www.cs.waikato.ac.nz/ml/weka/

Second, we annotated the major topics of each thread. The annotators were told to choose all the topics that they thought were discussed actively in the thread. As the guide for active topics, and to prevent the proliferation of minor topics, any topic they think is described in fewer than three posts was ignored. The maximum number of keywords for each topic is five, including variations of inflected forms.

4.3 Datasets

I-Jam 2008 Corpus. As the dataset used for the experiment on the I-Jam 2008 Corpus, we selected 10 threads from 10 fora in the corpus. From each forum, a thread was randomly selected from those with 15–80 posts. This is because smaller threads might have unclear, noisy thread structure, while larger threads are expensive to annotate. The average number of posts in these threads is 36.3, and each post consists of 6.7 sentences on average. The average number of key sentences per post was 1.54; the average number of topics per thread was 4.10. We performed a simple test of inter-annotator agreement between two annotators. The result was roughly 70%[3] for the key sentence extraction and 60% for the topic extraction.

We used two different experimental settings: HALF–HALF split and five-fold cross validation. For the HALF–HALF split setting, we divided the dataset into two halves, one for training (5 threads, 170 posts) and the other for evaluation (5 threads, 193 posts). In the five-fold cross validation setting, we divided the dataset into five parts: three for training, one for development, and one for evaluation. The reason that we use two different settings is that because the hand-coded cue words were taken from the training portion of HALF–HALF setting, it cannot be evaluated in the five-fold cross validation setting, although the results with the cross validation is more reliable. Because the graph-based models require no supervised training, the training sets are used only in the supervised model. The development set that we used in the preliminary experiment consists of two threads other than any of the 10 threads described above.

BC3 Corpus. Among information of various kinds annotated in the BC3 Corpus, we only use information of the extractive summaries to evaluate the performance of our summarization model. We used the same normalization as [25], such as converting "I" and "us" into "[person]." We did not use the locality measure for calculating the lexical chain score because this corpus has no explicit tree structure.

The dataset is split into five balanced portions (A, B, C, D, and E). Each part is used either as training, development, or a test set by turns in a five-fold cross validation scheme. In each trial, three sets are used as the training set, one set as the development set, and the other set as the testing set. Each set includes eight threads with roughly 600–700 sentences. Using the development set, the regularization coefficient σ for the regression-based model is set.

[3] In the measurement of the inter-annotator agreement, two annotators were requested to select only one sentence from a post if they annotated more than one sentence. This requirement is stricter than the annotation scheme, and is therefore lowering the agreement rate.

4.4 Baseline and Evaluation

For key sentence extraction, the first baseline we used is a simple but powerful classifier that extracts the first sentence from each post. We also reimplemented the graph-based method by [5], and used this as the second baseline. Each model outputs the sentence with the highest score as the key sentence for each post; the evaluation is based on whether or not this sentence is included in the gold standard sentences. In the BC3 Corpus, because annotations by three annotators exist, we used the average weighted recall, known as the pyramid precision [26], to calculate the final score[4]. The weighted recall is given as

$$\mathrm{WeightedRecall} = \frac{\sum_{i \in \mathrm{Sent_{Summary}}} \mathrm{score}_i}{\sum_{i \in \mathrm{Sent_{Gold}}} \mathrm{score}_i} \ , \tag{8}$$

where score_i is the number of annotators who selected the sentence in the extractive summary, normalized by the sentence length.

For the topic extraction task, we used two baselines: the TF-IDF and the *edge score*s. TF-IDF is a widely used metric for keyword extraction; it is calculated by the term frequency multiplied by the logarithm of the inverse document (post) frequency. Another baseline we propose to use is the *edge score* of a keyword w, which is calculated with the edge scores in the graph. After the PageRank is calculated for each vertex, the *edge score* of a word w is calculated as

$$S(w) = \ln \frac{\#doc}{\mathrm{DF}(w)} \sum_{e \in \mathcal{E}(w)} R(s_{\mathrm{start}}) R(s_{\mathrm{end}}) \ , \tag{9}$$

where $\mathcal{E}(w)$ represents the set of edges associated with word w, and $\mathrm{DF}(w)$ denotes the document (post) frequency of the word w. In the topic-extraction task, the recall is used as the evaluation measure because the number of the topics in a thread is given to the model (i.e., The model always outputs the same number of topics as the gold standard.). Recall is calculated based on how many of the output topics are actually included in the gold summaries.

4.5 Cue Words

From the development set in the HALF–HALF setting, we chose 31 cue words and heuristically set weights for these words, as listed in Table 2. Most of these seems to be general expressions used in a braimstorming-type discussion. For example, conjunctions, such as 'therefore' and 'for this reason,' are obviously good indicators of concluding sentences, and phrases, such as "point/problem is" and "one thought is," are used to draw reader's attention.

[4] As [27] mentioned, the ROUGE score, which has been widely used in the summarization of newswire texts, reportedly does not correlate well with human evaluations in the meeting domain [28]. Therefore, we used the standard measure in the domain of the e-mail summarization, following [27].

4.6 Results and Discussion

Key-sentence extraction on I-Jam 2008 Corpus. Table 3 shows experimental results for key-sentence extraction on the I-Jam 2008 Corpus. (a)–(e) are the models without supervised training, while (f) is a supervised model. † denotes statistically significant improvement[5] over "(c) Graph (MDS)." Our model with cue words outperforms the baseline model by a substantial margin of 4.97%, even though the cue words we used were hand-coded and limited in size. The use of lexical chains further improved the performance by 3.32%. These results underscore the effectiveness of the proposed use of structured lexical chains and cue words. The SVR-based supervised model with lexical features showed a slight improvement over the graph-based models. However, this improvement is marginal compared to the improvement that is provided by use of the hand-coded cue words. This difference suggests that the manual annotation of a small number of cue words is effective, and that the unsupervised model with minimum human effort works sufficiently well.

Table 4 shows a list of the highest-weighted features for the I-Jam 2008 Corpus when we use a maximum-entropy classifier[6] with exactly the same feature set as in the

Table 2. Cue words and the associated weights used in the experiment on the I-Jam 2008 Corpus

Bonus words	should (1.3), would (1.1), could (1.1), important (1.2), significant (1.2), real (1.2), now (1.2), proposal (1.1), idea (1.1), challenge (1.1), conclusion (1.3), suggest (1.2), propose (1.1), believe (1.1), need (1.1), thus (1.2), therefore (1.3), so (1.1), for this reason (1.3), I/my (1.1), so there (1.1), problem is (1.2), point is (1.2), is/are to (1.1), one thought is (1.3), it 's (1.1), I think (1.1), need to (1.2)
Stigma words	example (0.6), for example (0.7), for instance (0.4), agree (0.3)

Table 3. Experimental results for key-sentence extraction on the I-Jam 2008 Corpus. The † denotes statistically significantly improvement over "(c) Graph (MDS)".

	HALF–HALF	five-fold
(a) Baseline	40.88%	43.11%
(b) Graph (SDS)	45.86%	46.11%
(c) Graph (MDS)	60.22%	60.18%
(d) +Cue word	65.19%†	-
(e) +Lex. chain	68.51%†	61.98%
(f) SVR + (e)	69.61%†	62.87%

Table 4. Highest-weighted features for the I-Jam 2008 Corpus

Expression	Label	α Value
the idea	F	17.40
translate	T	9.19
question	T	8.59
suggest	T	8.20
I would	F	6.30
idea	T	5.73
could you	T	5.51
you can	F	5.44
as I	F	5.36
such	F	4.91

[5] All significance tests are based on McNemar's test.

[6] Note that an SVR model outputs no weight information.

SVR-based model. The first column shows the word forms of extracted expressions. The second column shows whether the feature is associated with *true* (i.e. included in the summary) or *false* (i.e. excluded from the summary). The third column shows the α weights in the maximum-entropy classifier. The result seems quite reasonable. The phrase "the idea" is shown to be stigmatic because it is typically used to mention the content of the last message in a precursive expression before stating the author's own idea. In contrast, expressions such as "question" and "suggest" are bonus words which are used to state the author's own question and suggestion. Thus, it seems apparent that the lexical features captured the importance of cue words, and contributed to the result.

Topic extraction on I-Jam 2008 Corpus. Table 5 presents the results for topic extraction. Our model with the structured lexical chains outperforms the two baselines by a large margin, and shows that the structured chains captured the topical information of the thread. However unfortunately, because the annotated data are too few, we were unable to infer the statistical significance of the improvements.

Table 6 presents an example of generated lexical chains. This example is taken from a thread on the I-Jam 2008 Corpus. Chains consisting only of the same word occurrences are excluded; the structure of chains is also omitted. It is apprent from this example that most chains seem to express a topic or theme of a discussion thread, and lexical chains are appropriately capturing semantically related words, such as near-synonyms "abuse–use" and antonyms "pessimists–optimists." In this example, only the bottom one seems wrong because the word "rules" is misclassified as having an incorrect sense "formulae."

Table 5. Experimental results for topic extraction on the I-Jam 2008 Corpus

	Micro Avg.	Macro Avg.
TF-IDF	19.51% (8/41)	23.08%
Edge score	24.39% (10/41)	25.92%
Proposed	36.59% (15/41)	34.17%

Table 6. An example of generated lexical chains on the I-Jam 2008 Corpus

Score	Chain
1.39	salary wage salary pay wage
0.74	abuse abuse misuse
0.69	alumni graduate
0.64	pessimists optimists
0.10	rules formulae formulae

Extractive Summarization on BC3 Corpus. Table 7 presents the experimental results for key-sentence extraction on the BC3 Corpus. "ME" and "BAG" respectively corresponds to the maximum-entropy and bagging classifiers. "no-lex" models do not use lexical features, while "lex" models do. "lex-lc" models do use both lexical and lexical-chain features. Both in the maximum-entropy and bagging models, the use of lexical feature improved the performance by around 1.2%. The use of lexical chains further improved the model by 2.0%. The performance of "BAG (lex-lc)" was better than "BAG (no-lex)" with the statistical significance level of $p < 0.05$. In Table 7, "GOLD Avg." is the average of weighted recalls for three gold annotations, which is considered to be an upper limit of the score. Considering this fact, the highest recall of 67.54% is a fairly good result. The score seems to be lower than that of [27], who

Table 7. Experimental results for key-sentence extraction on the five-fold cross validation on the BC3 Corpus

(a) Baseline	43.24%
(b) Graph (MDS)	57.42%
(c) ME (no-lex)	61.40%
(d) ME (lex)	62.61%
(e) BAG (no-lex)	65.33%
(f) BAG (lex)	65.50%
(g) BAG (lex-lc)	67.54%†
GOLD Avg.	74.60%

reported approximately 80% weighted recall. However, considering that we used almost identical feature sets as those, and considering that 80% is higher than the performance of "GOLD Avg.," this difference is attributed to the difference in the evaluation criteria, probably the calculation of the weighted average recall. Therefore, we can conclude that the performance of our model is comparable to or better than the performance of [27]. Even if this were not the case, we at least demonstrated that the use of cue words and lexical chains is effective in both discussion and mailing list corpora.

5 Conclusion

In this paper, we have presented a key-sentence and topic extraction model for multi-topical, threaded corpora, using structured lexical chains and cue words. Particularly, we proposed to use the structured lexical chains, which can incorporate the locality and continuity of the topics with a thread structure. Evaluation of the model was performed on the two datasets: The Innovation Jam 2008 Corpus and the BC3 E-mail Conversation Corpus. On the I-Jam 2008 Corpus, the use of cue words greatly improved the extractive summarizer. The use of structured lexical chains further improved the performance. The experiment on the keyword extraction task also revealed the effectiveness of the structured lexical chains, which is also confirmed by manual analysis. It is remarkable that even a few cue words improved the model significantly, although the further improvement by a supervised machine learning technique was marginal. This represents a hopeful finding for constructing a model with minimal supervision. We also conducted an experiment on the BC3 Mailing List Corpus, again demonstrating that the use of lexical features and lexical chains improved the model. As a whole, we conclude that the summarization of structured discussion corpora can be accomplished using an unsupervised model with structured lexical chains and cue words, and manual selection of a handful of cue words is effective, saving time used for creating training data for supervised learning. In future works, we are planning to conduct the experiment on larger and more diverse corpora, to validate the current result and to analyze the domain dependence of the model further. Also, we think that a more probabilistic formalization is necessary to achieve better performance.

Acknowledgement

We are grateful to the anonymous reviewers for their valuable comments. This work was partially supported by Grant-in-Aid for Specially Promoted Research (MEXT, Japan), and JSPS (Japan Society for the Promotion of Science) Research Fellowship.

References

1. Morris, J., Hirst, G.: Lexical cohesion computed by thesaural relations as an indicator of the structure of text. Computational Linguistics 17(1), 21–43 (1991)
2. Chen, Y.M., Wang, X.L., Liu, B.Q.: Multi-document summarization based on lexical chains. In: International Conference on Machine Learning and Cybernetics (2005)
3. Li, J., Sun, L.: A lexical chain approach for update-style query-focused multi-document summarization. In: Li, H., Liu, T., Ma, W.-Y., Sakai, T., Wong, K.-F., Zhou, G. (eds.) AIRS 2008. LNCS, vol. 4993, pp. 310–320. Springer, Heidelberg (2008)
4. Edmundson, H.: New methods in automatic extracting. Journal of the ACM 16(2), 264–285 (1969)
5. Wan, X.: An exploration of document impact on graph-based multi-document summarizaion. In: EMNLP 2008: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Morristown, NJ, USA, pp. 755–762. Association for Computational Linguistics (2008)
6. Farrell, R., Fairweather, P.G., Snyder, K.: Summarization of discussion groups. In: CIKM 2001: Proceedings of the Tenth International Conference on Information and Knowledge Management, pp. 532–534. ACM, New York (2001)
7. Mckeown, K., Shrestha, L., Rambow, O.: Using question-answer pairs in extractive summarization of email conversations. In: Gelbukh, A. (ed.) CICLing 2007. LNCS, vol. 4394, pp. 542–550. Springer, Heidelberg (2007)
8. Ulrich, J., Murray, G., Carenini, G.: A publicly available annotated corpus for supervised email summarization. In: AAAI 2008 EMAIL Workshop, Chicago, USA. AAAI, Menlo Park (2008)
9. Klaas, M.: Toward indicative discussion fora summarization. In: UBC CS TR-2005-04 (2005)
10. Hu, M., Sun, A., Lim, E.P.: Comments-oriented blog summarization by sentence extraction. In: CIKM 2007: Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, pp. 901–904 (2007)
11. Zhou, L., Hovy, E.: On the summarization of dynamically introduced information: Online discussions and blogs. In: AAAI Symposium on Computational Approaches to Analysing Weblogs (AAAI-CAAW), pp. 237–242 (2006)
12. Zajic, D.M., Dorr, B.J., Lin, J.: Single-document and multi-document summarization techniques for email threads using sentence compression. Inf. Process. Manage. 44(4), 1600–1610 (2008)
13. Rambow, O., Shrestha, L., Chen, J., Lauridsen, C.: Summarizing email threads. In: HLT-NAACL 2004: Proceedings of HLT-NAACL 2004: Short Papers on XX, Morristown, NJ, USA, pp. 105–108. Association for Computational Linguistics (2004)
14. Lam, D., Rohall, S.L., Schmandt, C., Stern, M.K.: Exploiting e-mail structure to improve summarization. In: ACM 2002 Conference on Computer Supported Cooperative Work (CSCW 2002) (2002)
15. Carenini, G., Ng, R.T., Zhou, X.: Summarizing email conversations with clue words. In: WWW 2007: Proceedings of the 16th International Conference on World Wide Web, pp. 91–100 (2007)

16. Barzilay, R., Elhadad, M.: Using lexical chains for text summarization. In: Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization, pp. 10–17 (1997)
17. Brunn, M., Chali, Y., Pinchak, C.: Text summarization using lexical chains. In: Document Understanding Conference (DUC), pp. 135–140 (2001)
18. Ercan, G., Cicekli, I.: Using lexical chains for keyword extraction. Inf. Process. Manage. 43(6), 1705–1714 (2007)
19. Mihalcea, R.: Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (2005)
20. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web (1999)
21. Silber, H.G., McCoy, K.F.: Efficiently computed lexical chains as an intermediate representation for automatic text summarization. Computational Linguistics 28(4), 487–496 (2002)
22. Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press, Cambridge (1998)
23. Ulrich, J., Carenini, G., Murray, G., Ng, R.: Regression-based summarization of email conversations. In: 3rd Int'l AAAI Conference on Weblogs and Social Media (ICWSM 2009), San Jose, CA. AAAI, Menlo Park (2009)
24. Miyao, Y., Tsujii, J.: Maximum entropy estimation for feature forests. In: Proc. of Human Language Technology Conf. (HLT) (2002)
25. Carvalho, V.R., Cohen, W.W.: Improving "email speech acts" analysis via n-gram selection. In: ACTS 2009: Proceedings of the HLT-NAACL 2006 Workshop on Analyzing Conversations in Text and Speech, Morristown, NJ, USA, pp. 35–41. Association for Computational Linguistics (2006)
26. Carenini, G., Ng, R.T., Zhou, X.: Summarizing emails with conversational cohesion and subjectivity. In: Proceedings of ACL 2008: HLT, Columbus, Ohio, pp. 353–361. Association for Computational Linguistics (2008)
27. Ulrich, J.: Supervised machine learning for email thread summarization. Master's thesis, University of British Columbia (2008)
28. Liu, F., Liu, Y.: Correlation between rouge and human evaluation of extractive meeting summaries. In: HLT 2008: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies, Morristown, NJ, USA, pp. 201–204. Association for Computational Linguistics (2008)