Automatic Generation of Review Matrices as Multi-document Summarization of Scientific Papers

Hayato Hashimoto[1]*, Kazutoshi Shinoda[2], Hikaru Yokono[3], and Akiko Aizawa[4]

[1] The University of Tokyo
hayat.hashimoto@gmail.com
[2] The University of Tokyo
kazutoshi.shinoda0516@gmail.com
[3] Fujitsu Laboratories Ltd.
yokono.hikaru@jp.fujitsu.com
[4] National Institute of Informatics, The University of Tokyo
aizawa@nii.ac.jp

Abstract. A synthesis matrix is a table that summarizes various aspects of multiple documents. In our work, we specifically examine a problem of automatically generating a synthesis matrix for scientific literature review. As described in this paper, we first formulate the task as multi-document summarization and question-answering tasks given a set of aspects of the review based on an investigation of system summary tables of NLP tasks. Next, we present a method to address the former type of task. Our system consists of two steps: sentence ranking and sentence selection. In the sentence ranking step, the system ranks sentences in the input papers by regarding aspects as queries. We use LexRank and also incorporate query expansion and word embedding to compensate for tersely expressed queries. In the sentence selection step, the system selects sentences that remain in the final output. Specifically emphasizing the summarization type aspects, we regard this step as an integer linear programming problem with a special type of constraint imposed to make summaries comparable. We evaluated our system using a dataset we created from the ACL Anthology. The results of manual evaluation demonstrated that our selection method using comparability improved performance.

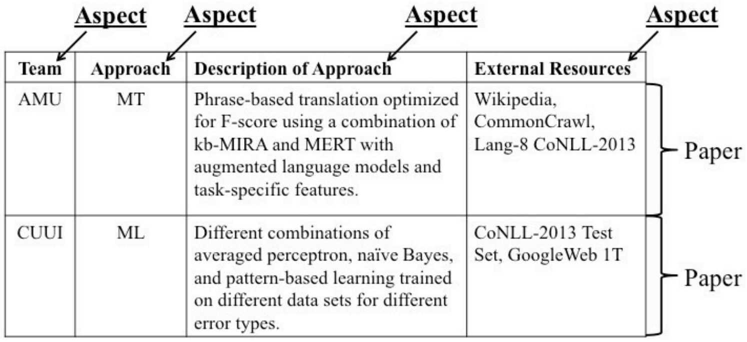
Keywords: multi-document summarization, review matrix, scientific paper mining

1 Introduction

Literature surveys are a fundamentally important part of research. Nevertheless, the increasing amounts of scientific literature demand a great deal of time for finding and reading all relevant papers. Although survey articles are often

* Currently at Google.

available for major topics, they are not always available for new or small topics. To address and mitigate these issues in surveying, scientific summarization has been widely studied. In scientific summarization, the input is scientific papers related to a certain topic. The goal is to generate a summary of them.



Aspect
Aspect
Aspect
Aspect
Paper
Paper

Team	Approach	Description of Approach	External Resources
AMU	MT	Phrase-based translation optimized for F-score using a combination of kb-MIRA and MERT with augmented language models and task-specific features.	Wikipedia, CommonCrawl, Lang-8 CoNLL-2013
CUUI	ML	Different combinations of averaged perceptron, naïve Bayes, and pattern-based learning trained on different data sets for different error types.	CoNLL-2013 Test Set, GoogleWeb 1T

Fig. 1. Example of a synthesis matrix. The matrix is from an overview paper: the CoNLL 2014 shared task [16]. Only a few rows and columns are presented here.

A synthesis matrix, or a (literature) review matrix, is a table showing a summary of multiple sources in different aspects. Synthesis matrices, which are regarded as effective tools for literature review [7], allow readers to analyze and compare source documents from different points of view. For example, an overview paper for a shared task typically includes a table that presents comparison of systems participating in the task (e.g., [16]).

Table 1 represents one example of a synthesis matrix. In this matrix, each row corresponds to a paper; each column corresponds to an aspect. For instance, the *Approach* column shows roughly what type of approach is used in each system by categorizing approaches into four types, whereas the *Description of Approach* column presents details of the approaches.

Our goal is to generate such matrices automatically. We formulate the task of automatic synthesis matrix generation as text summarization. Then we propose a model for the task. What makes synthesis matrix generation different from general summarization is that documents are mutually compared in summaries. We propose a system that is designed to capture this characteristic.

Our system is based on query-focused summarization (QFS), a variant of text summarization in which a generated summary provides an answer or a support to a query. A QFS-based approach alone, however, cannot achieve the characteristic described above because it only processes a single document at a time. To make summaries comparable, we incorporate the idea of comparative summarization, which aims to clarify and emphasize differences among documents. The proposed system consists of two steps: sentence ranking and sentence selection. The former step ranks sentences using the query focused version of LexRank

[17]. The latter selects sentences using integer linear programming (ILP) with an objective function that reflects comparability.

For evaluation, we created a dataset consisting of synthesis matrices taken from overview papers of shared tasks in the ACL Anthology, a database of papers on NLP. We conducted automatic evaluation using the evaluation metric ROUGE as well as manual evaluation by comparing the system output to the references. We experimented with various combinations of query relevance and query expansion to see the effectiveness of these techniques. We also compared our ILP-based sentence selection method with multiple greedy baseline methods. Results showed that our method is effective for synthesis matrix generation.

Our contributions can be summarized as the following. (1) We analyzed synthesis matrices in NLP and formulated the task of synthesis matrix generation. (2) We proposed a system based on LexRank and ILP for the task. (3) We showed that consideration of comparability between papers improves the performance of the proposed system.

2 Analysis of Synthesis Matrices and Task Formulation

2.1 Dataset Construction

We created a dataset from papers on the ACL Anthology[5], a full-text archive of papers on natural language processing [6]. In the construction of the dataset, we first selected the eight shared tasks listed in Table 1. For each shared task, we extracted a summary table of the participating systems, and (ii) corresponding system description papers. Here, we consider the summary table as a golden synthesis matrix for the description papers.

Next, we extracted sentences from the system description papers. We used XML format files that had been converted automatically from their original PDF versions using the SideNoter Project [1]. Because the XML files include the section structure of the papers, all extracted sentences were associate with the section titles in which they appear. Text that appears in specific regions, such as captions, footnotes, or references, was excluded. The Genia sentence splitter (GeniaSS)[7] was used for sentence splitting. Table 1 shows the fundamental statistics of our dataset. We also used the text corpus obtained from the entire ACL Anthology to calculate word embeddings used in Equations 5 and 8.

2.2 Aspect Phrasing

In the synthesis matrices we analyzed in this paper, an aspect is always phrased as a noun phrase: e.g., *System Architecture* and *Verb*. Because aspects are used in the header of a synthesis matrix, they are often very brief and ambiguous

[5] http://aclanthology.info/

[6] Because our method does not rely on any external knowledge related to the domain, we expect that the proposed framework is applicable to other domains as well.

[7] http://www.nactem.ac.uk/y-matsu/geniass/

Table 1. Dataset statistics.

Task Name	Summary Paper ID	#Aspects	#Papers	Average #Sentences
CoNLL-2011	W11-1901	3	19	110.2
CoNLL-2012	W12-4501	3	11	146.1
CoNLL-2013	W13-3601	3	14	178.1
CoNLL-2014	W14-1701	3	10	153.9
CoNLL-2015	K15-2001	2	14	116.6
CACS 2014	W14-3907	1	7	127.3
SemEval-2010-1	S10-1001	3	4	99.0
SemEval-2014-8	S14-2008	1	9	111.7

(e.g. *Syntax* and *Error*). Such aspects are sometimes extremely difficult to understand, even for humans, when presented with no context. We can regard these phrases as shortened, condensed version of the actual aspects, which can be expressed precisely in longer phrases or sentences: *Error* in the previous example actually is a short version of *Error types the system handles*, or more specifically, *Grammatical error types the system attempts to detect and correct.*

When considering a system that generates a synthesis matrix, users would give more specific aspects rather than such *header-style* aspects. In fact, questions or queries in datasets for query-focused summarization are worded much more clearly and in greater detail as examples from the DUC 2006 dataset [3] shows: *Describe theories related to the causes and effects of global warming and arguments against these theories.* If brief, unclear aspects are the only clue about what a system is presumed to find. It would be safe to say that the task of synthesis matrix generation is a considerably difficult task to address. In this work, however, we use header-style aspects in the dataset we created for experiments because it is not trivial how we should elaborate the original aspects.

2.3 Aspect Types

First, we analyzed the synthesis matrices to ascertain what kind of aspect synthesis matrices typically have, and what kind of answer they expect. We categorized aspects into the following four types:

1. Description: Sentences or phrases are anticipated as an answer. (36%)
   e.g., *Description of approach – Phrase-based translation optimized for...*
2. Item: Identify entities or concepts given a factoid type question. This includes numerical entities such as performance scores. (31%)
   e.g., *Learning method [used in the system] – Naive Bayes, MaxEnt*
3. Choice: Selection from a predefined vocabulary set. Multiple choice is often allowed. (24%)
   e.g., *Error [types that the system handles] – SVA, Vform, Wform*
4. Binary: The answer is yes or no. (9%)
   e.g., *[Whether the system uses] external resources – No*

The examples presented above are actual aspect–answer pairs from the matrices. Words in brackets are added for clarification.

Description and Item, the two most frequent types, can be handled within a summarization framework. Description can naturally be regarded as abstractive summarization. For Item, sentences that provide information about the answer can be extracted in a summarization approach. For instance, if the expected answer to an aspect *external resources used* is *Wikipedia*, then a sentence including the information that *Wikipedia* is used as an external resource can also be regarded as an answer. Based on the observation, we specifically examine Description and Item type aspects in this paper.

In total, we collected 218 summaries, which we divided into a development set (4 matrices, 101 summaries) for parameter tuning and a test set (4 matrices, 117 summaries). The development set has four Description queries and three Item queries. The test set has seven Description queries and five Item queries. The average length of a query is 1.7 words for the development set and 2.2 words for the test set. The average length of a reference summary is 5.9 words for the development set and 8.9 words for the test set.

3 Approach

3.1 Overview of the Proposed Method

Assuming that aspects are mutually independent, we define the task of synthesis matrix generation as described below.

- **Input**: K documents $\{D_i\}_{1\leq i\leq K}$ and an aspect a_j
- **Output**: K summaries of input documents based on a_j $(1 \leq j \leq K)$

Our method is based on extractive summarization, where the objective is to select a set of sentences in a document given the maximum length of the summary.

Figure 2 presents an overview of the proposed framework. Our system consists of two steps: sentence ranking and sentence selection. In the sentence-ranking step, the system ranks sentences in the input papers by regarding aspects as queries. In the sentence selection step, the system selects sentences that remain in the final output from the rankings.

3.2 Sentence Ranking

Query-Focused LexRank LexRank, a graph-based sentence ranking method presented by Erkan and Radev [5], is widely used for summarization. This method first constructs a graph in which each node represents a sentence. Then to each edge it assigns similarity between the sentences the adjacent nodes represent. It then ranks the nodes by considering a random walk on the graph and by finding the stationary distribution.

Actually, LexRank was demonstrated as useful for query-focused summarization with a small modification to the algorithm [17], which we will call Q-LexRank. Q-LexRank adds query relevance to edge weights to value sentences

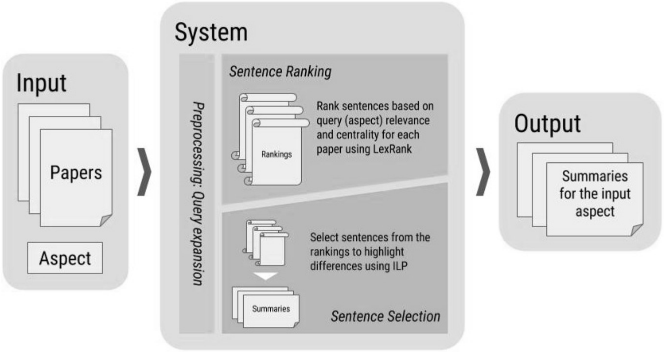

Input
Papers
Aspect
System
Preprocessing: Query expansion
Sentence Ranking
Rankings
Rank sentences based on query (aspect) relevance and centrality for each paper using LexRank
Select sentences from the rankings to highlight differences using ILP
Summaries
Sentence Selection
Output
Summaries for the input aspect

Fig. 2. Overview of the proposed method.

that are related to the query. The score $p(s \mid q)$ of a sentence s given a query q is defined as

$$p(s \mid q) = d\frac{\mathrm{rel}(s \mid q)}{\sum_{s' \in D} \mathrm{rel}(s' \mid q)} + (1-d)\sum_{s' \in D} \frac{\mathrm{sim}(s, s')}{\sum_{s'' \in D} \mathrm{sim}(s', s'')} p(s' \mid q), \tag{1}$$

where D is the input document. The first term represents how relevant the sentence s is to the query q. The second term represents how similar s is to the other sentences. Here, d functions as a query bias, which balances these terms.

We use the cosine measure defined in the original LexRank to compute sentence similarity as

$$\mathrm{sim}_0(x, y) = \frac{\sum_{w \in x,y} \mathrm{tf}_{w,x}\mathrm{tf}_{w,y}\mathrm{idf}_w^2}{\sqrt{\sum_{w \in x}(\mathrm{tf}_{w,x}\mathrm{idf}_w)^2}\sqrt{\sum_{w \in y}(\mathrm{tf}_{w,y}\mathrm{idf}_w)^2}}, \tag{2}$$

where x and y are sentences, $\mathrm{tf}_{w,x}$ is the number of times w appears in x, and

$$\mathrm{idf}_w = \log\left(\frac{n+1}{0.5 + |\{s \in D \mid w \in s\}|}\right). \tag{3}$$

When the model constructs a graph, this similarity value is set to zero when it is less than a similarity threshold t: Using the Iverson bracket, $\mathrm{sim}(x, y) = [\mathrm{sim}_0(x, y) \geq t](\mathrm{sim}_0(x, y))$. We used the query bias $d = 0.95$ and the similarity threshold $t = 0.2$ following the original Q-LexRank.

Query Expansion Query expansion is a commonly used information retrieval technique. It is expected to help the system find related sentences that have low relevance to the original query. We test two query expansion methods:

- Add words that frequently co-occur with the query words in document D_i the system is processing (COOCCUR).

- In addition to the words added to COOCCUR, add frequently co-occurring words in the entire document set $D_1, \dots, D_K$ (COOCCUR+).

We add the five most frequently co-occurring words for COOCCUR. For COOCCUR+, we add five words for the current document and five words for the entire document set.

Use of Word Embedding in Query Relevance The query relevance of a sentence s to a query q is defined as follows in the original Q-LexRank paper [17] using tf-idf values:

$$\mathrm{rel}_{\mathrm{tfidf}}(s \mid q) = \sum_{w \in q} \log(\mathrm{tf}_{w,s} + 1) \times \log(\mathrm{tf}_{w,q} + 1) \times \mathrm{idf}_w. \tag{4}$$

One problem of this measure is that it becomes non-zero only when s includes at least one word in q, which yields a very small number of sentences with a non-zero query relevance value.

We use a query relevance measure based on word embedding to address this problem. We define query relevance measures using word vectors as

$$\mathrm{rel}_{\mathrm{emb}_n}(s \mid q) = \frac{1}{n} \mathrm{sumLargest}_n \{ cos(\boldsymbol{v}_w, \boldsymbol{v}_u) \mid w \in s, u \in q \} \tag{5}$$

where $\mathrm{sumLargest}_n$ is a function that returns the sum of the n largest values. We only use the largest values because smaller cosine values do not usually convey precise information about word similarity.

3.3 Sentence Selection

Integer Linear Programming Based Sentence Selection In the sentence selection step, the system selects sentences from the rankings computed in the ranking step to reduce the redundancy of the resulting summaries. We use an ILP-based model proposed by McDonald [14]. This method selects sentences by maximizing the sum of the scores of the selected sentences and minimizing similarity between them as

$$\mathrm{maximize}_{\boldsymbol{\alpha}, \boldsymbol{\beta}} \quad \lambda \sum_{i=1}^{N} p(s_i \mid q) \alpha_i - (1 - \lambda) \sum_{i<j} \mathrm{sim}_0(s_i, s_j) \beta_{i,j} \tag{6}$$

subject to

$$\begin{gathered} \alpha_i, \beta_{i,j} \in \{0, 1\}, \ \textstyle\sum_{l=1}^{N} \mathrm{len}(s_l) \alpha_l \leq L, \\ \alpha_i + \alpha_j - \beta_{i,j} \leq 1, \, \beta_{i,j} \leq \alpha_i, \quad \beta_{i,j} \leq \alpha_j \end{gathered} \tag{7}$$

for $1 \leq i < j \leq N$. Here, $\mathrm{len}(s_i)$ is the length of the sentence s_i; also, L is the maximum length of the resulting summary. The importance bias $\lambda \in [0, 1]$ is tuned in the experiments. We designate this method as ILP. To reduce the number of variables, we keep only the top 20 sentences in the rankings, i.e., $N = 20$ in Equation 6.

Comparative Summarization The goal of comparative summarization is to highlight differences among given documents. Most earlier studies treat comparative summarization as an optimization problem with an objective function that measures comparability. Comparability is typically measured as similarity between a summary pair.

Because summaries for the input papers must specifically examine the given aspect, we can expect them to have structurally and semantically similar sentences. For example, if the aspect is *Approach*, then summaries are likely to include sentences describing *what is used* or *what is applied*. Even though ***what** is used* differs for each paper, it is true for all papers that *something is **used***. We propose *action-based* similarity and incorporate it into the objective function to capture this nature of comparability and to align topics of summaries for a certain aspect.

Although it might not be readily apparent, we can identify the action of a sentence. We adopt a simple heuristic using dependency trees to ascertain which words describe the action. In the Universal Dependency Treebank for English[8], a dataset of dependency trees, 57% of sentence heads are verbs, 17% are nouns, 10% are adjectives, and 9% are proper nouns. Exploiting this knowledge, we use the sentence head head(s) of a sentence s in our system. We define action-based similarity $\mathrm{sim}_{\mathrm{act}}$ using word embedding as

$$\mathrm{sim}_{\mathrm{act}}(x, y) = \cos(\boldsymbol{v}_{\mathrm{head}(x)}, \boldsymbol{v}_{\mathrm{head}(y)}) \ . \tag{8}$$

We incorporate the action-based similarity in the objective function because it greatly increases the number of variables to assign to sentences in all input documents. Because it optimizes summaries for all documents simultaneously, we consider optimizing a single summary at a time. The system processes the input documents $D_1, \ldots, D_K$ in that order. For document D_l, we modify Equation 6 as

$$\begin{aligned}\mathrm{maximize}_{\boldsymbol{\alpha},\boldsymbol{\beta}} \quad & \lambda \sum_{i=1}^{N} p(s_i \mid q)\alpha_i + \frac{\mu}{|\mathcal{S}_l|} \sum_{i=1}^{N} \sum_{s' \in \mathcal{S}_l} \mathrm{sim}_{\mathrm{act}}(s_i, s')\alpha_i \\ & - (1 - \lambda - \mu) \sum_{i<j} \mathrm{sim}_0(s_i, s_j)\beta_{i,j}, \end{aligned} \tag{9}$$

where $\mathcal{S}_l = S_1 \cup \ldots \cup S_{l-1}$ and $\lambda, \mu \in [0, 1]$ $(\lambda + \mu \leq 1)$. This model maximizes similarity between the summary for the current document and the summaries for the already- processed documents. We designate this method as ILP+.

4 Experiments

4.1 Experimental Setup

The objectives of the experiments are the following: The first is to identify the best combination of query expansion (NO EXPANSION, COOCCUR and COOCCUR+

[8] http://universaldependencies.org/

) and relevance measure calculation ($\mathrm{rel}_{\mathrm{tfidf}}$ and $\mathrm{rel}_{\mathrm{embs}}$). The second is to investigate the applicability of the proposed comparative summarization method (ILP+) by comparing the result with (ILP) and also comparing it with two baseline methods:

- BEST: Select sentences from the top of the rankings until the summary length reaches L, skipping a sentence if adding it makes the summary exceed the limit.
- GREEDY: Select sentences such as BEST but skip sentences similar to any of the already-selected sentences within the summary. We set the threshold for this to 0.6, i.e., sentences s and s' are similar when $\mathrm{sim}_0(s, s') > 0.6$.

Before the ranking step, input sentences and queries are tokenized, lowercased, and stemmed. Stopwords are removed from both sentences and queries. We set the maximum summary length L to 30. Word vectors were learned on the entire ACL Anthology using word2vec[9] with the default parameters.

4.2 Evaluation Methods

For performance evaluation, we first applied ROUGE[10] [12], a metric set for evaluation of summarization. This report describes ROUGE-2 and ROUGE-SU4 scores.

We also evaluated the system manually, similarly to the pyramid method [15]. We first reviewed the reference summaries manually and identified *summary content units* (SCUs) for each summary. Actually, SCUs are semantically cohesive text units that are not longer than a sentence. Each item in the list is considered an SCU in a ITEM-typed summary. For the DESCRIPTION type, we made SCUs as small as possible if they make sense alone because it is possible that a system summary includes only some part of the information the reference summary has. We believe that such summaries should be evaluated positively.

We evaluated system summaries using SCUs by manually counting how many SCUs each system summary includes. This report describes macro-average and micro-average of coverage for the entire test set ($\mathrm{SCU}_{\mathrm{macro}}$ and $\mathrm{SCU}_{\mathrm{micro}}$, respectively). However, judging whether a summary covers an SCU is not trivial. We ascertained that a summary covers an SCU when the summary implies what the SCU indicates in context. Words in the SCU appearing in the summary but in different context were not counted.

4.3 Results: Sentence Ranking Parameters

Table 2 presents the system performance in different combinations of a query relevance measure and a query expansion method. Here, BEST is used as a selection method to examine the results of sentence ranking specifically. The parameters

[9] https://code.google.com/archive/p/word2vec/
[10] http://www.berouge.com/

n in $\text{rel}_{\text{emb}_n}$ were tuned in terms of ROUGE scores on the development set using grid search and set to 8.

The two best-performing combinations in terms of ROUGE scores were $\text{rel}_{\text{tfidf}}$ and cooccur+, and $\text{rel}_{\text{emb}_8}$ and no query expansion. These two combinations also performed the best in both micro-average and macro-average coverage.

As presented in Table 2, $\text{rel}_{\text{tfidf}}$ worked well with query Expansion, although $\text{rel}_{\text{emb}_8}$ worked best without query expansion. Both word-embedding-based query relevance and query expansion aim to overcome simplicity of queries: Word-embedding-based query relevance measures attempt to find sentences that have no query words in them but which are relevant by assigning high scores to them. In contrast, query expansion strives to do the same thing by adding words related to the query itself. Using both, sentences including words similar to added query words are deemed relevant by the model, which leads to not-so-relevant sentences being ranked highly.

Table 2. Performance of the system using different query relevance measures and query expansion methods.

Relevance measure	Query expansion	ROUGE -2	ROUGE -SU4	Manual $\text{SCU}_{\text{macro}}$	Manual $\text{SCU}_{\text{micro}}$
$\text{rel}_{\text{tfidf}}$	(none)	0.0635	0.0703	0.220	0.168
	cooccur	0.0656	0.0691	0.180	0.146
	cooccur+	0.0880	0.0952	0.245	0.193
$\text{rel}_{\text{emb}_8}$	(none)	0.0820	0.0945	0.246	0.234
	cooccur	0.0339	0.0400	0.096	0.078
	cooccur+	0.0359	0.0500	0.106	0.093

4.4 Results: Sentence Selection Methods

We tuned the importance bias λ and μ in terms of ROUGE scores for ilp and ilp+ using the development set. We used $\lambda = 1.0$ for ilp and $(\lambda, \mu) = (0.8, 0.2)$ for ilp+ in the following experiments. We picked the best two combinations in the previous section: (A) $\text{rel}_{\text{tfidf}}$ & cooccur+ and (B) $\text{rel}_{\text{emb}_8}$ alone.

Table 3 presents the performance of the systems using different sentence selection methods. Unlike the ROUGE scores, manual evaluation suggests that ilp+ is the best of all methods. Results show that ilp came between ilp+ and best/greedy for combination B but performed the worst for combination A.

Effect of Comparative Summarization Results showed that the term added for redundancy prevention was not effective. Redundancy reduction might not be necessary in this task because input documents typically have more than a hundred sentences and because they have few redundant sentences.

Results show that ilp+ performed better than the baseline methods in manual evaluation. Unlike the other three methods, ilp+ uses information of the summaries for the other input documents. Such information might be helpful for the system to generate a cohesive set of summaries. However, ILP+ does not consider comparability globally at the same time. It relies on already-generated summaries for other documents, which means the output depends on the order of the documents that are processed. A fast global optimization algorithm would provide better performance.

Table 3. Performance of systems using different sentence selection methods.

Relevance	Query	Sentence	ROUGE	ROUGE4	Manual	Manual
measure	expansion	selection	-2	-SU4	$\mathrm{SCU}_{\mathrm{macro}}$	$\mathrm{SCU}_{\mathrm{micro}}$
(Combination A)						
$\mathrm{rel}_{\mathrm{tfidf}}$	COOCCUR+	BEST	0.0880	0.0952	0.245	0.193
		GREEDY	0.0882	0.0939	0.245	0.193
		ILP	0.0674	0.0733	0.218	0.174
		ILP+	0.0510	0.0565	0.271	0.215
(Combination B)						
$\mathrm{rel}_{\mathrm{emb}_8}$	(none)	BEST	0.0820	0.0945	0.246	0.234
		GREEDY	0.0798	0.0923	0.248	0.237
		ILP	0.0705	0.0807	0.274	0.240
		ILP+	0.0686	0.0787	0.284	0.271

4.5 Discussion

Table 4 presents an example of summaries generated using different selection methods. In this example, the summaries by ILP+ cover more SCUs in the gold-standard summaries than the summaries by BEST do. This example shows that ILP+ picks sentences with similar actions: Sentences include verbs such as *use* and *apply*, which are often used to describe approaches.

Table 4. Example of output summaries. $\mathrm{rel}_{\mathrm{emb}_8}$ is used. No query expansion is applied. The aspect is *Description of approach.*

Paper ID	Gold-standard	BEST	ILP+
W14-1709	N-gram-based approach finds unlikely n-gram ”frames,” which are then corrected via high-scoring LM alternatives. Rule-based methods then improve the results for certain error types.	Various classification methods and statistical machine translation based methods will be investigated in the router-based approach to find the tailored methods for the given word. Because the count __Figure 2__	We use the probability n-gram Vector approach to correct Nn. The Google corpus is also used for an n-gram vector approach and for routerbased approaches.
W14-1712	External resources correct spelling errors, whereas a conditional random field model corrects comma errors. SVA errors are corrected using an RB approach. All other errors were corrected using a language model. Interacting errors are corrected using an MT system.	The method involves automatically learning translation models based on a Web-scale ngram. A dependency-based model is proposed in this paper to handle preposition errors.	We apply the CRF model proposed by Israel et al. We use a rule-based method in this module. Language models trained on well-formed text are used to validate the replacement decisions.

5 Related Work

Summarization of scientific papers has been studied widely. Some earlier studies have used citation networks [20, 2], which are based on the idea that sentences describing a cited paper have crucial information related to the cited paper. Some other works specifically examine the surmounting of the incoherence obstacle posed by summaries generated from multiple documents. Surveyor [10] combines content and discourse models to generate coherent summaries. Parveen et al. [18] proposed a graph-based approach that extracts coherence patterns from a corpus and uses them.

Actually, QFS was a shared task at the Document Understanding Conferences 2005-2006. A number of methods have been proposed for the task. The BayeSum [4] algorithm is based on a Bayesian statistical model. Liu et al. [13] proposed an unsupervised deep learning architecture and demonstrated its effectiveness. Fisher and Roark [6] used feature similarity and centrality metrics as well as query relevance and applied machine learning. Although most QFS approaches are extractive, Wang et al. [23] proposed an abstractive QFS framework using sentence compression.

A small amount of research has been done for comparative summarization. Huang et al. [9] proposed a linear-programming-based approach to comparative news summarization. Wang et al. [22] formulated a task of comparative summarization, which aims to highlight differences between multiple document groups, and proposed a discriminative sentence selection approach. Although contrastive summarization refers mainly to opinion summarization, similar ideas can be found in it. We found a limited number of studies of contrastive summarization for product reviews [11, 21] and for controversial topics [19, 8].

6 Conclusion

We analyzed synthesis matrices in NLP-related papers and formulated the task of synthesis matrix generation, and proposed a system for the task using query-focused and comparative summarization techniques. For sentence ranking, we adopted query-focused LexRank with modifications to redeem tersely expressed queries. For sentence selection, we incorporated the idea of comparability in an ILP-based sentence selection framework. By measuring sentence similarity, we attempted to align summaries for different papers to make them mutually contrastive. The results of automatic and manual evaluation suggest that our selection method, which considers comparability, is effective for the task.

We believe that our task formulation of automatic review matrix generation is worthy of additional effort. In our framework, an aspect is expressed only as a short noun phrase. As compensation, we used frequently co-occurring words or word embeddings in our query-sentence relevance calculation. We observed that using such techniques sometimes produces an unexpected sentence ranking. Introduction of more descriptive aspects or domain ontologies is one avenue that demands further investigation. In addition, this paper presents no consideration of Choice and Binary type aspects (Sec. 2.3). How to formalize these types as question-answering tasks is another issue to address.

This work was supported by JSPS KAKENHI Grant Numbers 16K12546, 16H01756.

References

1. Takeshi Abekawa and Akiko Aizawa. Sidenoter: Scholarly paper browsing system based on PDF restructuring and text annotation. In *COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference System Demonstrations, December 11-16, 2016, Osaka, Japan*, pages 136–140, 2016.
2. Arman Cohan and Nazli Goharian. Scientific article summarization using citation-context and article's discourse structure. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015.
3. Hoa Trang Dang. Overview of duc 2006. In *Proceedings of DUC 2006: Document Understanding Workshop*, 2006.
4. Hal Daumé, III and Daniel Marcu. Bayesian query-focused summarization. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 305–312, 2006.
5. Günes Erkan and Dragomir R Radev. LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479, 2004.
6. Seeger Fisher and Brian Roark. Query-focused summarization by supervised sentence ranking and skewed word distributions. In *Proceedings of the Document Understanding Conference, DUC-2006*, 2006.
7. Judith Garrard. *Health sciences literature review made easy: the matrix method.* Aspen Publishers, 1999.
8. Jinlong Guo, Yujie Lu, Tatsunori Mori, and Catherine Blake. Expert-guided contrastive opinion summarization for controversial issues. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, pages 1105–1110, 2015.
9. Xiaojiang Huang, Xiaojun Wan, and Jianguo Xiao. Comparative news summarization using linear programming. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011.
10. Rahul Jha, Reed Coke, and Dragomir Radev. Surveyor: A system for generating coherent survey articles for scientific topics. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 2167–2173. AAAI Press, 2015.
11. Kevin Lerman and Ryan McDonald. Contrastive summarization: An experiment with consumer reviews. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, June 2009.
12. Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, July 2004.
13. Yan Liu, Sheng-hua Zhong, and Wenjie Li. Query-oriented multi-document summarization via unsupervised deep learning. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, pages 1699–1705. AAAI Press, 2012.
14. Ryan McDonald. A study of global inference algorithms in multi-document summarization. In *Proceedings of the 29th European Conference on IR Research*, ECIR'07, 2007.
15. Ani Nenkova, Rebecca Passonneau, and Kathleen McKeown. The pyramid method: Incorporating human content selection variation in summarization evaluation. *ACM Trans. Speech Lang. Process.*, 4(2), May 2007.

16. Tou Hwee Ng, Mei Siew Wu, Ted Briscoe, Christian Hadiwinoto, Hendy Raymond Susanto, and Christopher Bryant. The CoNLL-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, 2014.
17. Jahna Otterbacher, Güneş Erkan, and Dragomir R. Radev. Using random walks for question-focused sentence retrieval. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT-EMNLP '05, pages 915–922, 2005.
18. Daraksha Parveen, Mohsen Mesgar, and Michael Strube. Generating coherent summaries of scientific articles using coherence patterns. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, November 2016.
19. Michael J. Paul, ChengXiang Zhai, and Roxana Girju. Summarizing contrastive viewpoints in opinionated text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 66–76, 2010.
20. Vahed Qazvinian and Dragomir R. Radev. Scientific paper summarization using citation summary networks. In *Proceedings of the 22ndd International Conference on Computational Linguistics - Volume 1*, COLING '08, pages 689–696, 2008.
21. Ruben Sipos and Thorsten Joachims. Generating comparative summaries from reviews. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management*, CIKM '13, 2013.
22. Dingding Wang, Shenghuo Zhu, Tao Li, and Yihong Gong. Comparative document summarization via discriminative sentence selection. *ACM Trans. Knowl. Discov. Data*, 6(3):12:1–12:18, October 2012.
23. Lu Wang, Hema Raghavan, Vittorio Castelli, Radu Florian, and Claire Cardie. A sentence compression based framework to query-focused multi-document summarization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, 2013.