Grouping Sentences as a Better Language Unit for Extractive Text Summarization

Mengyun Cao and Hai Zhuge*

Institute of Computing Technology, Chinese Academy of Sciences, University of Chinese Academy of Sciences, Beijing, China
Laboratory of Cyber-Physical-Social Intelligence, Guangzhou University, China
Laboratory of Semantics, Knowledge and Data, Aston University, Birmingham, UK
**Corresponding author: Hai Zhuge (e-mail: h.zhuge@aston.ac.uk).*

Abstract

Most existing methods for extractive text summarization aim to extract important sentences with statistical or linguistic techniques and concatenate these sentences as a summary. However, the extracted sentences are usually incoherent. The problem becomes worse when the source text and the summary are long and based on logical reasoning. The motivation of this paper is to answer the following two related questions: What is the best language unit for constructing a summary that is coherent and understandable? How is the extractive summarization process based on the language unit? Extracting larger language units such as a *group* of sentences or a *paragraph* is a natural way to improve the readability of summary as it is rational to assume that the original sentences within a larger language unit are coherent. This paper proposes a framework for *group*-based text summarization that clusters semantically related sentences into groups based on Semantic Link Network (SLN) and then ranks the groups and concatenates the top-ranked ones into a summary. A two-layer SLN model is used to generate and rank groups with semantic links including the *is-part-of* link, *sequential* link, *similar-to* link, and *cause-effect* link. The experimental results show that summaries composed by *group* or *paragraph* tend to contain more key words or phrases than summaries composed by sentences; and, summaries composed by *groups* contain more key words or phrases than those composed by *paragraphs*, especially when the average length of source texts is from 7,000 words to 17,000 words, which is the usual length of scientific papers. Further, we compare seven clustering algorithms for generating groups and propose five strategies for generating groups with the four types of semantic links.

Keywords: Text Summarization; Semantic Link Network; Clustering; Natural Language Processing.

1. Introduction

Automatic text summarization is an important research direction in the area of natural language processing. It is a desirable approach to deal with the rapid growth of online texts. A good summary should convey the core ideas of the source texts fluently while minimizing the redundancy [1]. Text summarization tasks can be classified into single-document or multi-document summarization tasks according to the number of source texts, and can be classified into generic or theme-focused (also known as query-focused) summarization tasks according to whether the summarization concerns all themes or specific themes of the source text. Text summarization methods are generally divided into extractive and abstractive [2]. Extractive methods usually rank sentences by analyzing statistical and linguistic features such as word/phrase frequency, part-of-speech of words/phrases, position of sentence, and so on, and extract top-k ranked sentences to compose a summary [3]. Abstractive methods attempt to discover the main events (or key words/phrases) and their relations in the source text based on background lexicon and linguistic knowledge, and rephrase these main events (or key words/phrases) into new sentences with possible new words or phrases [1].

A *Sentence* is a basic language unit that represents a complete meaning. Within a meaningful text, there are many semantic links between sentences such as cause-effect, sequential, example-of, and so on. However, most extractive summarization methods rank sentences rarely considering the semantic interdependence between sentences. Therefore, these methods essentially extract each sentence in isolation, and the summary composed by these isolated sentences usually has low readability because the context and the original semantic links between the selected sentences are lost. Figure 1 gives an example of losing context and semantic links in the summary composed by top-ranked sentences. Readers cannot know what "they" and "the learned function" refer to by just reading the generated summary, because sentences s_{26} and s_{37} that contain the correct referents are not extracted to generate the summary. We call these anaphors whose referents cannot be found from the text as dangling anaphors. Sentences s_{27} and s_{109} are the effects of two *cause-effect* links in the source text, and they are extracted to generate the summary while their corresponding causes (i.e. s_{26} and s_{108}) are not extracted due to their low ranks. This situation forms incomplete or incorrect representation in the generated summary, which confuses readers.

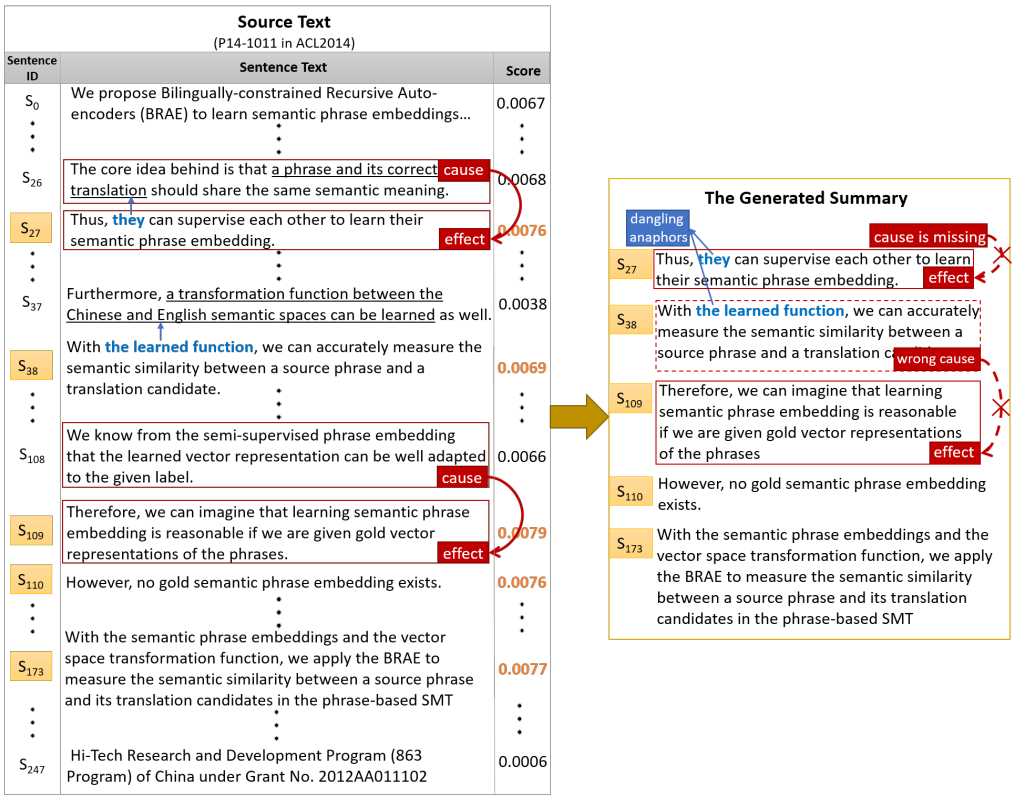


Figure 1. An example of losing context and semantic links in the summary composed by top-ranked sentences.

Since sentences in the source text are coherent, extracting a larger language unit that retains the original order of sentences is a natural way to improve the coherence of the generated summary.

A *paragraph* is a language unit where its sentences are organized by the writer(s) to convey a complete meaning to readers. It is reasonable to assume that sentences within a paragraph of a published text are coherent. However, ranking and extracting paragraphs directly may reduce the core ideas of the source text contained in the generated summary because: 1) a paragraph may discuss multiple themes but not all of them are related to the core ideas; 2) some sentences within a paragraph are just for helping complete the description of a theme (e.g., the use of examples), which should not be included in the summary; 3) discussion of a theme sometimes crosses paragraphs, i.e., several paragraphs may commonly render a theme of the source text. Therefore, ranking and extracting appropriate language units with a flexible range is more suitable to generate summaries.

A *group* is a language unit that contains a group of semantically related sentences on the same theme and the size of group is smaller than paragraph. There are different ways to organize a group. This paper focuses on using the following four types of semantic links to cluster sentences into groups.

- *Is-part-of* link, reflecting the relation between a system and its components. Within texts, sentences contribute to render the content of paragraphs, which render the content of sections, which further render the content of the whole text [1].
- *Sequential* link, reflecting a linear order of operating a process. Sentences are organized in sequential order so that readers can understand the content in the same order of representation. Some semantic links such as cause-effect link and temporal link depend on sequential order. Using the *sequential* link to organize group can reduce dangling anaphors within each group since the referents of most anaphors are located in nearby sentences [4].
- *Similar-to* link, reflecting similarity between things. It can be used to detect similar sentences when organizing group.
- *Cause-effect* link, reflecting causal relation between things. Two sentences linked by cause-effect link render the same theme. Therefore, a basic rule for organizing a group is that the sentence representing cause and the sentence representing the effect should be put into the same group.

The left-hand part of Figure 2 shows an example of grouping sentences according to the *sequential* link, *similar-to* link, and *cause-effect* link: the sentences of a group are consecutive within the source text; sentences s_{37} and s_{38} are clustered into group g_{26} with the lexical *similar-to* link; sentences s_{26} and s_{27} are clustered into group g_{19} and sentences s_{108} and s_{109} are clustered into group g_{85} with *cause-effect* links. The "paragraph ID ← sentence ID" shows the *is-part-of* link between *sentence* and *paragraph*, but it is not used as a clustering constraint in this example, so groups like g_{26} can contain sentences that belong to different paragraphs. The right-hand part of Figure 2 shows the summary generated by ranking these groups and extracting top-ranked groups. Comparing with the summary generated by ranking sentences in Figure 1, the summary generated by ranking groups remains more context and semantic links.

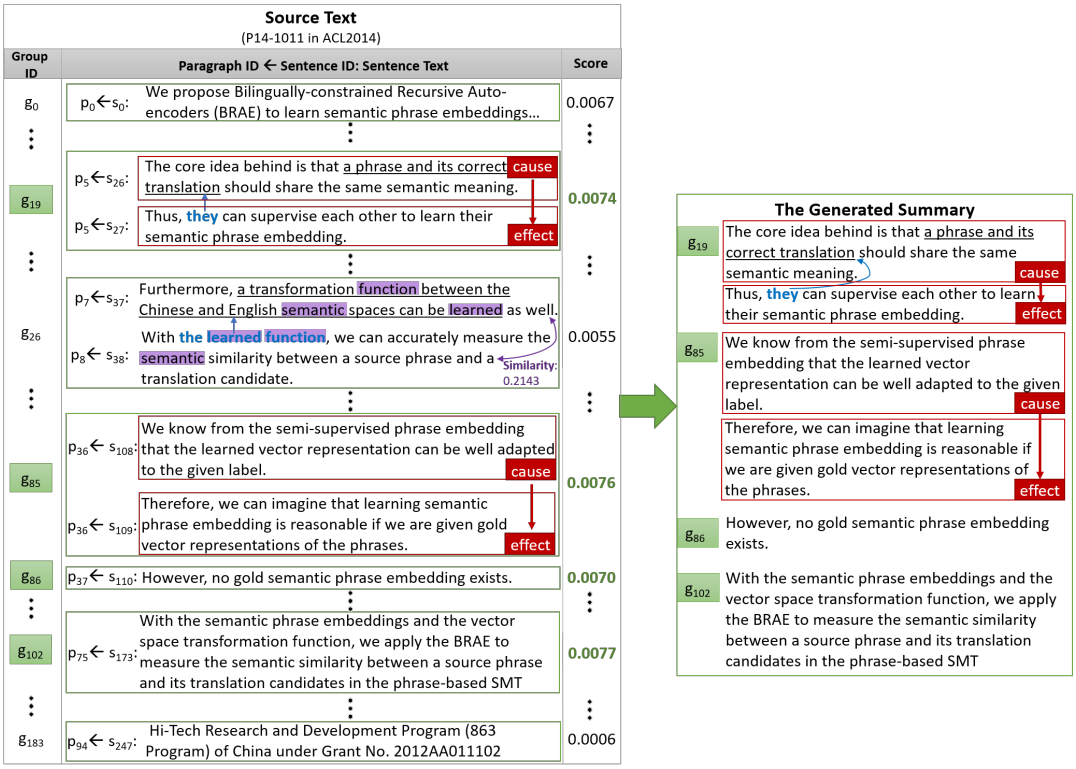


Figure 2. An example of generating a summary by ranking and extracting groups of semantically related sentences.

However, if a group contains too many sentences or the sentences within a group are less semantically linked, the summary generated by ranking and extracting such groups will contain more sentences that are not relevant to the theme of the source text. Therefore, it is necessary to investigate the suitable semantic links and the suitable clustering algorithm to generate groups.

2. Group-based Text Summarization

2.1 Framework

The *group*-based text summarization framework as shown in Figure 3 consists of four components: the Semantic Link Network (in short SLN [1, 5]) of sentences and paragraphs, the algorithm for generating groups, the SLN of groups, and the algorithm for ranking groups. Users can define their own approaches by selecting different semantic links to construct the SLNs and selecting (or designing) algorithms to generate and rank groups.

The framework provides a new approach to improve the current extractive text summarization by designing appropriate algorithms for generating groups connected by various semantic links.

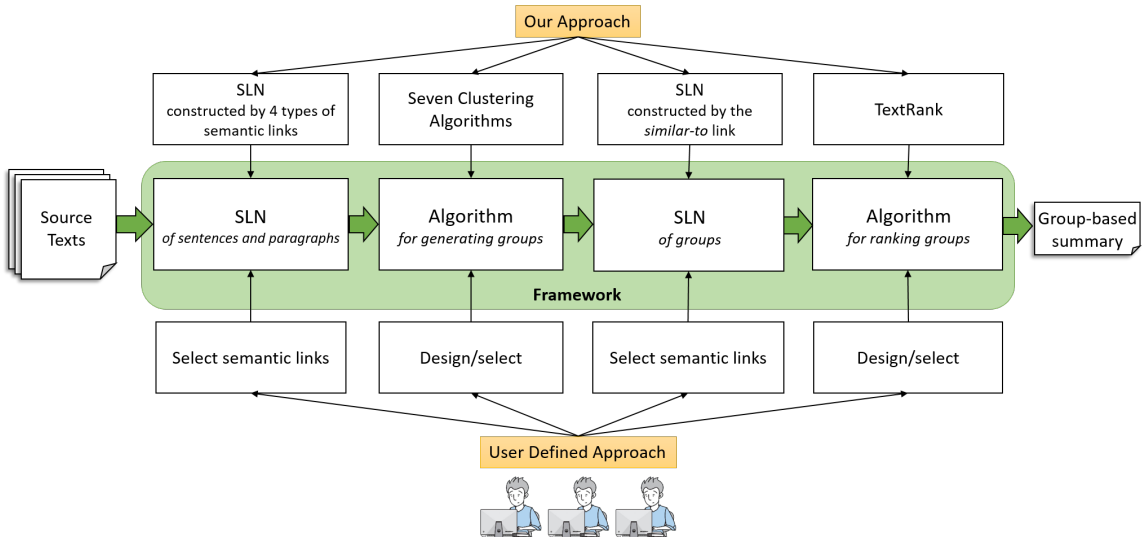


Figure 3. The framework for *group*-based text summarization.

2.2 The General Architecture of Our Approach

Basing on the framework for *group*-based text summarization, we design an approach that uses the *is-part-of*, *sequential*, *similar-to*, and *cause-effect* links to construct the SLN of sentences and paragraphs, selects one from the seven clustering algorithms we designed to generate groups, uses the *similar-to* link to construct the SLN of groups, and adopts the TextRank algorithm ([6]) to rank groups.

Figure 4 depicts the general architecture of this study. The middle of the figure is the model we designed. The top and bottom are the two baseline models that generate summaries by ranking sentences or paragraphs. The following four steps of the two-layer SLN model summarizing a source text correspond to the four components of the framework for *group*-based text summarization:

- Step 1: *Constructing the first-layer SLN*. The model splits the source text into paragraphs and sentences (nodes) connected by *is-part-of* links (in blue color), *sequential* links (in yellow color), *similar-to* links (in purple color; a thicker line represents higher similarity), and *cause-effect* links (in red color) to form the first-layer SLN.
- Step 2: *Clustering sentences into groups*. We designed seven clustering algorithms based on different combinations of the four types of semantic links. The model adopts one of the algorithms to perform on the first-layer SLN for generating groups.

- Step 3: *Constructing the second-layer SLN*. The model takes the groups generated in Step 2 as nodes and connects these nodes through *similar-to* links to form the second-layer SLN, which is also called the sim-SLN of *group* (sim-SLN is an SLN that contains only *similar-to* links).
- Step 4: *Ranking groups and then extracting top-ranked groups to compose a summary*.

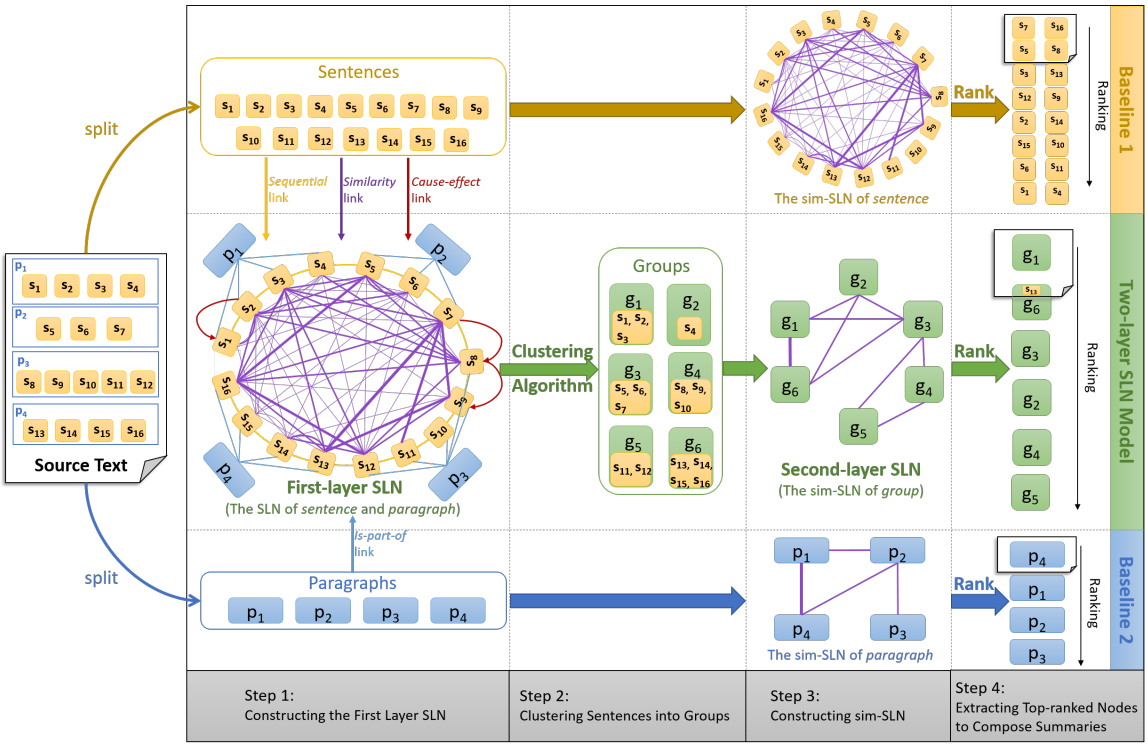


Figure 4. The general architecture of our approach.

3. Experiments

3.1 Aim and method

To verify the performance of *group* in generating summaries for long source texts, we make single-document summarization on scientific papers to compare the *group*-based summary generated by our two-layer SLN model with the *sentence*-based summary and *paragraph*-based summary generated by the two baseline models. Just as the Step 3 and Step 4 in Figure 4, the two baseline models summarize a source text by constructing the sim-SLN of *sentence* or *paragraph*, performing the TextRank algorithm to rank sentences or paragraphs, and composing a summary with the top-ranked sentences or paragraphs.

To investigate the effects of the four types of semantic links in generating and ranking groups, we further conducted experiments on the two-layer SLN model by using different clustering algorithms and using different types of *similar-to* links to construct the sim-SLN.

The seven clustering algorithms implemented for generating groups according to different combinations of the *is-part-of* link, *sequential* link, *similar-to* link, and *cause-effect* link are introduced in Appendix B. Each clustering algorithm is named according to the semantic links it uses, namely *Seq+pb/npb*, *SimSIZE*, *SimTHR*, *CE+pb/npb*, *CESim*, *SeqSim+pb/npb*, and *CESeqSim+pb/npb*.

We implemented six similarity metrics to calculate the weight of the *similar-to* link between two language units, including the lexical-based JCD metric, the embedding-based AVG, SIF and USE metrics, and the synsets-based LIN and WUP metrics. The details of the six similarity metrics are introduced in Appendix A. Each SLN instance at most contains one type of *similar-to* link.

The explicit *cause-effect* links contain causal cue words/phrases such as ‘because’ and ‘due to’. Implicit *cause-effect* links have no causal cue words/phrases, and readers can infer implicit *cause-effect* links by analyzing the sentences with their background knowledge. The two-layer SLN model uses explicit *cause-effect*

links to generate groups, because placing sentences of explicit *cause-effect* links into different groups could lead to incorrect (or incomplete) representation. We use the pattern-based algorithm [7] to discover the explicit *cause-effect* link between any pair of consecutive sentences.

3.2 Experimental Datasets

One dataset *ACL-all* contains 173 ACL2014 conference papers downloaded from the ACL Anthology. The other dataset *AI-all* contains 372 papers from the Artificial Intelligence journal from 1994 to 2018. To study the influence of paper length on the performance of extracting different language units to generate summary, the *ACL-all* dataset is divided into two subsets and the *AI-all* dataset is divided into six subsets. Table 1 shows the details of the two datasets and the eight sub-datasets, in which the column "*Average Length*" shows the average number of words in each sub-dataset.

We regard texts in *abstract, conclusion* and *introduction* of a scientific paper as three kinds of standard summaries. When the kind of standard summary is decided, the text in the standard summary is excluded from the automatic summarization process, and the number of words in the standard summary is regarded as the upper limit of the length of the generated summary. In Table 1, columns "*Abstract*", "*Conclusion*" and "*Introduction*" show the average number of words in each kind of standard summaries on different sub-datasets. We can generate longer summaries by simply using longer standard summaries.

Table 1. The details of the experimental datasets.

Datasets	Subsets	The Number of Papers	Average Length	Abstract	Conclusion	Introduction
ACL-all	*ACL-short*	85 short papers	2,823	99	127	464
	ACL-long	88 long papers	5,494	128	209	607
AI-all	*AI-less10*	59 papers less than 10,000 words	7,762	169	272	924
	AI-1013	57 papers between 10,000 and 13,000 words	11,396	184	374	1271
	AI-1316	70 papers between 13,000 and 16,000 words	14,516	201	600	1132
	AI-1619	65 papers between 16,000 and 19,000 words	17,201	205	542	1405
	AI-1925	60 papers between 19,000 and 25,000 words	21,566	198	596	1573
	AI-more25	61 papers more than 25,000 words	32,768	218	686	1893

3.3 Four Metrics for Comparing Summaries

ROUGE evaluation metrics, containing ROUGE-N ($N \in \{1, 2, 3, 4\}$) and ROUGE-L, are widely used to evaluate the lexical similarity between the generated summary and the standard summary [8]. Since the words or phrases used in standard summaries can be viewed as key words/phrases that reflect the core ideas of the source text, we design four metrics based on the ROUGE scores to compare the amount of key words/phrases contained in different kinds of generated summaries.

We connect the scores of ROUGE-1, ROUGE-2, ROUGE-3, ROUGE-4, and ROUGE-L to form a ROUGE vector for each kind of the generated summaries. Let $\boldsymbol{R_s} = \{R1_s, R2_s, R3_s, R4_s, RL_s\}$ be the ROUGE vector for the *sentence*-based summary, where Ri_s is the average F-score of ROUGE-i ($i \in \{1, 2, 3, 4, L\}$). Similarly, $\boldsymbol{R_p}$ and $\boldsymbol{R_g}$ are the ROUGE vectors for the *paragraph*-based summary and the *group*-based summary. Let $\boldsymbol{R_s}$ be the benchmark for comparison and $\boldsymbol{R}$ be the ROUGE vector of the summary to be compared, the four metrics we designed are shown below:

- ***L2*** **metric**

$$L2(\boldsymbol{R}||\boldsymbol{R_s}) = \sqrt[2]{\sum_{i\in\{1,2,3,4,L\}} (Ri)^2} - \sqrt[2]{\sum_{i\in\{1,2,3,4,L\}} (Ri_s)^2} \quad (1)$$

- ***Increase*** **metric**

$$Increase(\boldsymbol{R}||\boldsymbol{R_s}) = \sum_{i\in\{1,2,3,4,L\}} (Ri - Ri_s) \tag{2}$$

- ***Increase%*** **metric**

ROUGE-1 and ROUGE-L scores are usually much higher than ROUGE-3 and ROUGE-4 scores. In the *L2* metric and the *Increase* metric, a significant increase on $R3$ or $R4$ can be easily offset by a tiny decrease on $R1$ or RL. Therefore, *Increase%* is proposed for showing the average increment on each dimension.

$$Increase\%(\boldsymbol{R}||\boldsymbol{R_s}) = \frac{\sum_{i\in\{1,2,3,4,L\}} \frac{(Ri - Ri_s)}{Ri_s} * 100\%}{5} \tag{3}$$

- ***Diverge*** **metric**

Referring to Kullback–Leibler divergence which measures how a probability distribution diverges from the expected one [9], we propose $Diverge(\boldsymbol{R}||\boldsymbol{R_s})$ to measure the deviation of the ROUGE vector $\boldsymbol{R}$ from the basis ROUGE vector $\boldsymbol{R_s}$.

$$Diverge(\boldsymbol{R}||\boldsymbol{R_s}) = \sum_{i\in\{1,2,3,4,L\}} Ri * ln\frac{Ri}{Ri_s} \tag{4}$$

The value range of the $L2$ metric or the *Increase* metric is $[-5, 5]$, the value range of the *Increase%* metric is $[-100\%, 100\%]$, and the value range of the *Diverge* metric is $[-\infty, +\infty]$. If the values of more than two metrics are greater than zero, then we say that the summaries of $\boldsymbol{R}$ contain more key words/phrases than the summaries of $\boldsymbol{R_s}$.

3.4 Comparison based on Four Metrics

3.4.1 Preliminary Comparison

Our previous work [10] also used the framework for *group*-based text summarization to automatically generate summaries for each paper of the *ACL-all* dataset. It used the $SeqSim + pb(JCD, 0.06)$ algorithm to generate groups and used the JCD-type *similar-to* links to construct the sim-SLN.

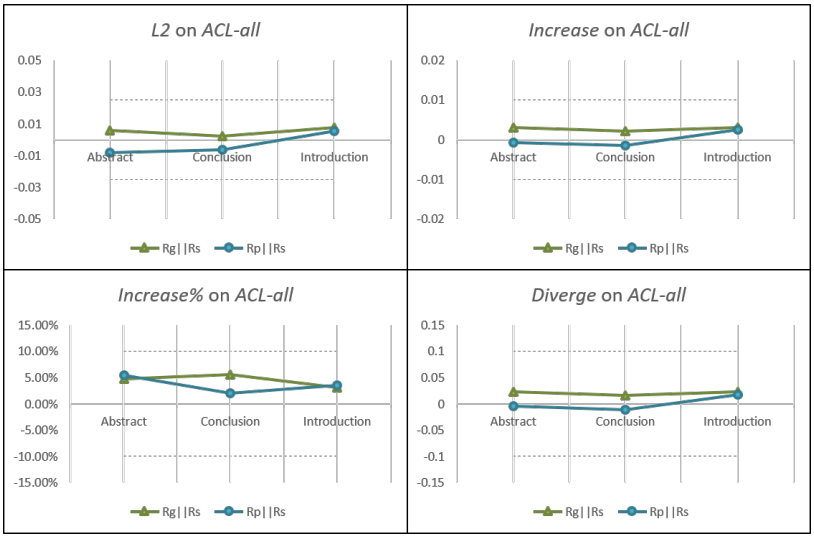


Figure 5. The values of the four metrics on the *ACL-all* dataset when using *SeqSim+pb(JCD, 0.06)* and JCD-type sim-SLN.

Figure 5 shows the values of the four metrics obtained from the experiments of the previous work. Series $\boldsymbol{R_g}||\boldsymbol{R_s}$ compares the *group*-based summary with the *sentence*-based summary. If $\boldsymbol{R_g}||\boldsymbol{R_s}$ is greater than zero on more than two metrics, we say the *group*-based summary contains more key words/phrases than the *sentence*-based summary. If $-0.025 \le L2(\boldsymbol{R_g}||\boldsymbol{R_s}) \le 0.025$, $-0.01 \le Increase(\boldsymbol{R_g}||\boldsymbol{R_s}) \le 0.01$, $-10\% \le Increase\%(\boldsymbol{R_g}||\boldsymbol{R_s}) \le 10\%$ and $-0.1 \le Diverge(\boldsymbol{R_g}||\boldsymbol{R_s}) \le 0.1$ (the value interval between two dotted lines in each

sub-figure of Figure 5), we say that the performance of *group* is similar to the performance of *sentence*. The $\boldsymbol{R_p}||\boldsymbol{R_s}$ compares the *paragraph*-based summary with the *sentence*-based summary in the same way. If the value of $\boldsymbol{R_g}||\boldsymbol{R_s}$ is greater than the value of $\boldsymbol{R_p}||\boldsymbol{R_s}$ on more than two metrics, we say that the *group*-based summary contains more key words/phrases than the *paragraph*-based summary.

For generating summaries for each paper of the *AI-all* dataset, we first use the same experimental settings used in the previous work to generate and rank groups. Figure 6 shows the values of the four metrics on the *AI-all* dataset. From Figure 5 and Figure 6, we find that the *group*-based summary contains more key words/phrases than *sentence*-based summary on both the *ACL-all* and the *AI-all* datasets. The *paragraph*-based summary contains more key words/phrases than the *sentence*-based summary on the *AI-all* dataset, but on the *ACL-all* dataset *paragraph* performs better than *sentence* only when *Introduction* is used as standard summary.

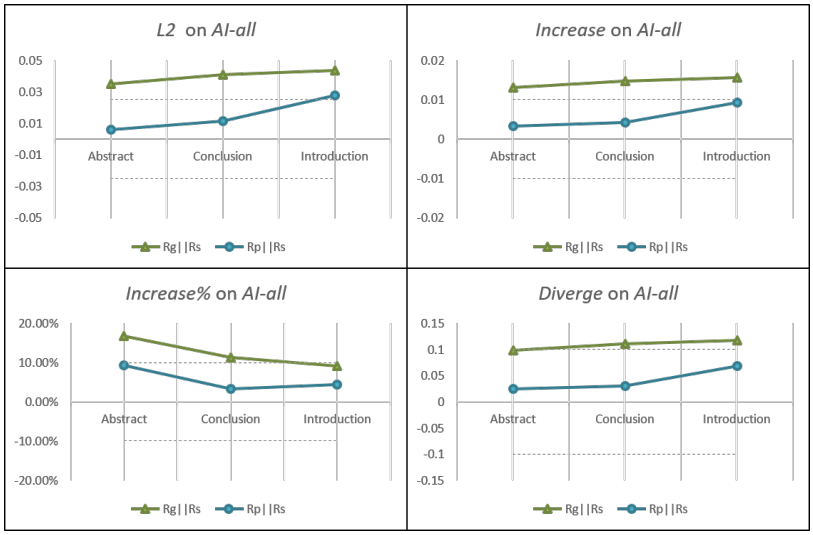


Figure 6. The values of the four metrics on the *AI-all* dataset when using *SeqSim+pb(JCD, 0.06)* and JCD-type sim-SLNs.

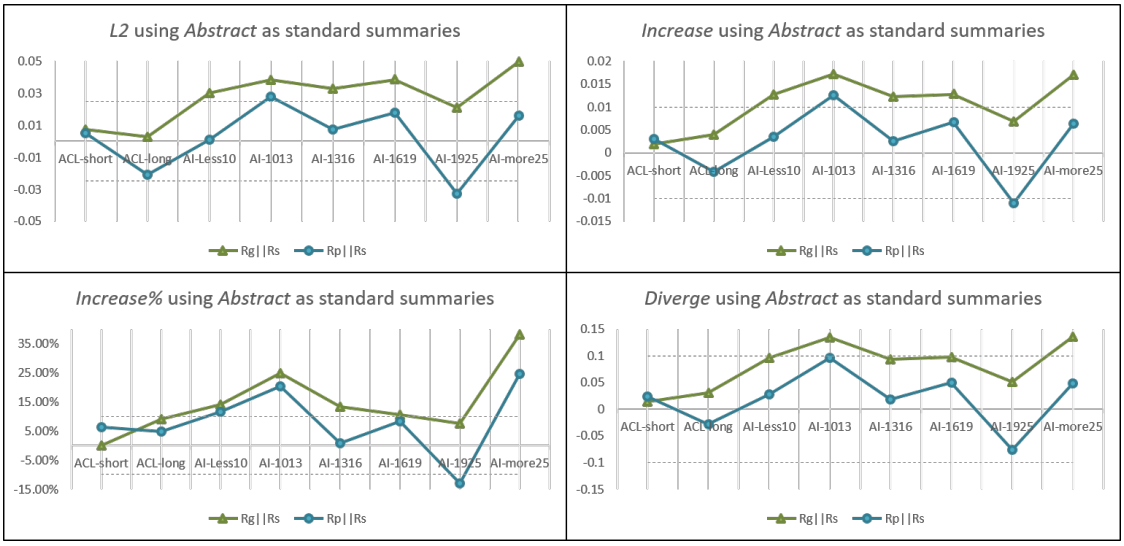


Figure 7. The values of the four metrics on each sub-dataset when using *SeqSim+pb(JCD, 0.06)* for generating groups and using JCD-type sim-SLN for ranking language units.

Figure 7 shows the values of the four metrics on each sub-dataset when using the *Abstract* as the standard summary, from which we see that the performance of *group* in generating summaries is significantly improved when the average length of the source texts is more than 7000 words. (Figure C-1 in Appendix C.1

gives the values of the four metrics on each sub-dataset when using the *Conclusion* or the *Introduction* as the standard summary, from which we draw the same conclusion).

Besides, we randomly sample some *sentence*-based, *group*-based and *paragraph*-based summaries for manually evaluating the readability of these summaries. The readability statistics shown in Appendix D verifies the assumption that the summary composed by larger language units has higher readability than the *sentence*-based summary.

3.4.2 Further Comparison

We further test the performance of *group* in generating summaries by changing the type of *similar-to* links used for constructing the sim-SLN, since different *similar-to* links guide TextRank to rank language units differently and thus cause different groups (sentences or paragraphs) are extracted to compose the summary. We still use the *SeqSim+pb(JCD, 0.06)* algorithm to generate groups, but use the JCD-type, AVG-type, SIF-type, USE-type, LIN-type, and WUP-type *similar-to* links in turn to construct the sim-SLN for ranking sentences, groups, and paragraphs.

Table C-1 to Table C-4 in Appendix C.2 shows the values of each metric on the *ACL-all* dataset and *AI-all* dataset when using different types of sim-SLN. Table 2 gives the statistics of the four metrics counted from Table C-1 to Table C-4. As shown in Table 2, when using different sim-SLNs to rank language units for extracting top-ranked ones to generate summaries, the *group-based approach* or the *paragraph-based approach* performs at least as good as the *sentence*-based approach with a probability greater than 97%, and the *group-based approach* or the *paragraph-based approach* performs better than *sentence* with a probability greater than 63%. This means that the *group*-based or *paragraph*-based summary tends to contain more key words/phrases than *sentence*-based summary even if we change the types of sim-SLN for ranking language units.

Table 2. The statistics of the four metrics in different ranges on *ACL-all* and *AI-all*.

Performance	The range of value of the four metrics	Larger Language Units	
		group	*paragraph*
At least similar to *sentence*	$L2(R\|\|R_s) > -0.025$	97.22%	97.22%
	$Increase(R\|\|R_s) > -0.01$	**100**%	97.22%
	$Increase\%(R\|\|R_s) > -10\%$	**100**%	97.22%
	$Diverge(R\|\|R_s) > -0.1$	100%	100%
Better than *sentence*	$L2(R\|\|R_s) > 0$	**75**%	63.89%
	$Increase(R\|\|R_s) > 0$	**77.78**%	66.67%
	$Increase\%(R\|\|R_s) > 0$	**83.33**%	77.78%
	$Diverge(R\|\|R_s) > 0$	**77.78**%	69.44%
Significantly better than *sentence*	$L2 (R\|\|R_s) > 0.025$	**27.78**%	25%
	$Increase(R\|\|R_s) > 0.01$	19.44%	**22.22**%
	$Increase\%(R\|\|R_s) > 10\%$	22.22%	**30.56**%
	$Diverge(R\|\|R_s) > 0.1$	13.89%	13.89%

In Table 2, the probability that "*group* is significantly better than *sentence*" is slightly less than the probability that "*paragraph* is significantly better than *sentence*". This is probably due to the existence of other concluding paragraphs within a scientific paper except for the text in *Abstract*, *Introduction* and *Conclusion*, and the sentences in a concluding paragraph are usually concluding sentences. If a concluding paragraph is extracted to compose a summary, the *paragraph*-based summary will contain more concluding sentences than the *group*-based summary since each group generated by the *SeqSim+pb(JCD, 0.06)* algorithm contains fewer sentences than the paragraph to which it belongs. However, given that the probability that "*group* is better than *sentence*" is 9% more than the probability that "*paragraph* is better than *sentence*" on average, we can still conclude that *group* tends to outperform both *sentence* and *paragraph*.

Figure 8. The percentage of cases where the four metrics are greater than zero on each sub-dataset.

Table C-5 to Table C-8 in Appendix C.2 shows the values of each metric on all the sub-datasets when using different types of sim-SLN. Figure 8 shows the percentages of cases where the four metrics are greater than zero on each sub-dataset in Table C-5 to Table C-8. From Figure 8 we can see that, when the source text becomes longer, the percentages of cases where $\boldsymbol{R_g}||\boldsymbol{R_s}$ is greater than zero first improves, then remains stable, and finally declines. The sub-datasets on which *group* has stable performance are *AI-less10, AI-1013, AI-1316, AI-1619*, so the best range of the average length of source texts for ranking groups to generate summaries is from 7000 words to 17000 words. However, the percentages of cases where $\boldsymbol{R_p}||\boldsymbol{R_s}$ is greater than zero varies frequently with the length of the source text, so it is difficult to determine the best range of the average length of source texts for the *paragraph*-based summary.

Combining the experiment results shown in the previous subsection, we can draw the following two conclusions:

Conclusion 1: *summaries composed by group or paragraph tend to contain more key words or phrases than summaries composed by sentence*.

Conclusion 2: *summaries composed by group contain more key words or phrases than those based on paragraph, especially when the average length of source texts is from 7,000 and 17,000 words*.

Figure 9 compares the performance of different types of *similar-to* links in ranking language units to generate summaries, where $L2(\boldsymbol{R_s})$, $L2(\boldsymbol{R_g})$, and $L2(\boldsymbol{R_P})$ are the L2 norms of the ROUGE-score vectors for the *sentence*-based, *group*-based, and *paragraph*-based summary respectively. We can see that the performance of the embedding-based *similar-to* link (AVG, SIF or GSE) becomes significantly better than the lexical-based (JCD) or synsets-based (LIN or WUP) *similar-to* link when the average length of source texts becomes longer.

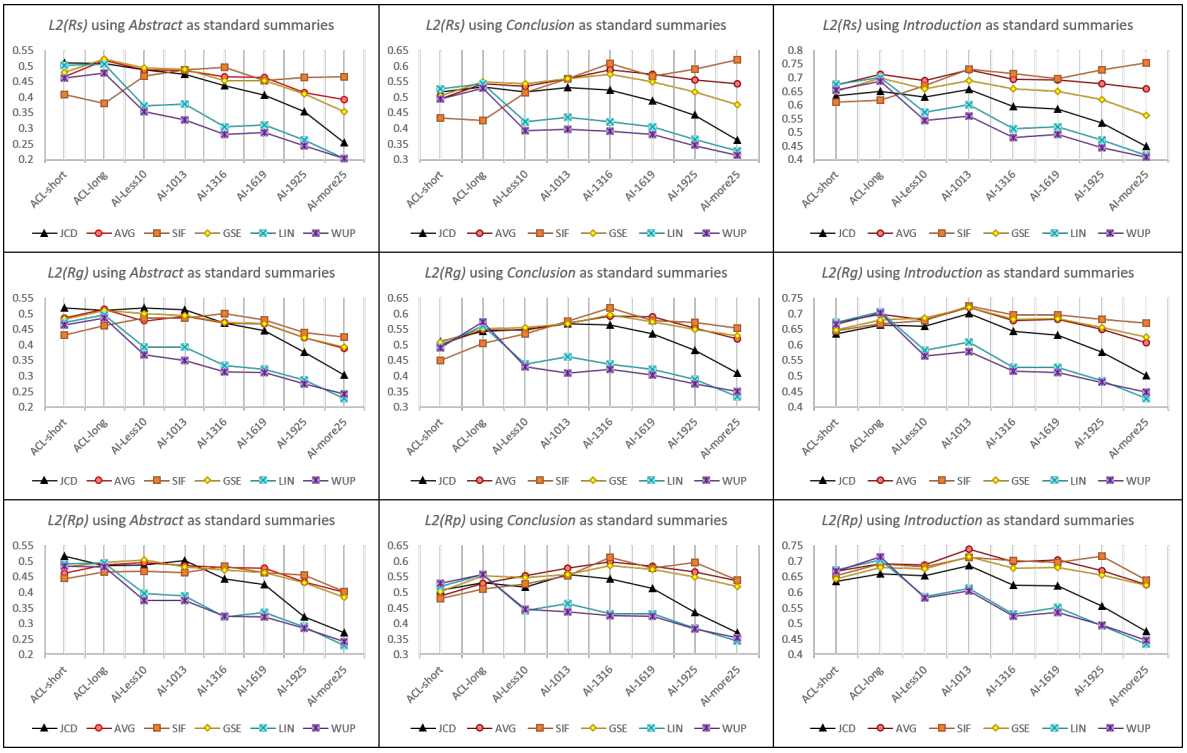


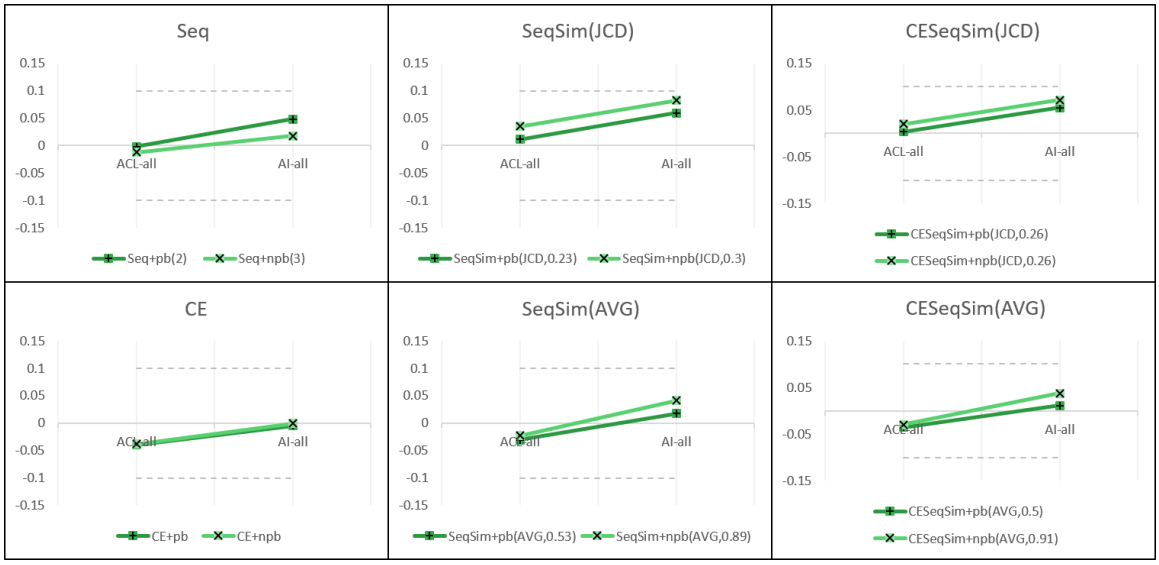
Figure 9. The L2 norms of *Rs, Rg* and *Rp* on each sub-dataset when using different types of sim-SLN to rank nodes.

3.4.3 The Role of Semantic Links in Generating Groups

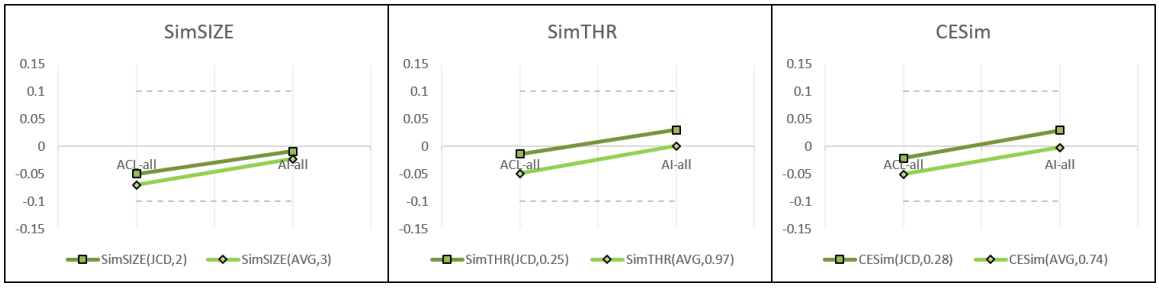
We compare the performance of the seven clustering algorithms to investigate the role of the four types of semantic links in generating groups. First, the *'+pb'* and *'+npb'* modes set in the *Seq*, *CE*, *SeqSim* and *CESeqSim* algorithms are compared to show the effect of using *is-part-of* link. Second, for *SimSIZE, SimTHR, CESim, SeqSim* and *CESeqSim* that need *similar-to* links, the lexical-based similarity and the embedding-based similarity are compared to show the effect of using different types of *similar-to* links. Third, by setting each algorithm with its suitable paragraph-bounded mode and its suitable type of *similar-to* links, the seven clustering algorithms are compared to show the best one for generating groups.

Other parameters used in the clustering algorithms are tuned on the *AI-less10* sub-dataset because on *AI-less10* the average length of a paper is moderate and the performance of *group* starts to be significantly better than the performance of *sentence* in composing summaries. The parameter *win_size* used in the *Seq* algorithm (or *size* used in *SimSIZE*) means the maximum number of sentences within a generated group. Since a group should contain fewer sentences than a paragraph and more than 95% paragraphs of the *ACL-all* and *AI-all* datasets contain less than10 sentences, we try with *win_size* (or *size*) = 2, 3… 10 to find the suitable size of *group* for the *Seq* algorithm (or the *SimSIZE* algorithm). The parameter *thr* used in *SimTHR, CESim, SeqSim* and *CESeqSim* algorithms refers to the similarity threshold to merge two language units. Since the weight of a *similar-to* link ranges from 0 to 1, we try *thr* = 0.01, 0.02, …, 0.99 to find the suitable similarity threshold for each of the four algorithms.

The AVG-type *similar-to* links are used for constructing the sim-SLNs and the text in the *abstract* of each paper is used as the standard summary. We only show the results of the *Diverge* metric here, since the four metrics show the same conclusions on the performance of different kinds of groups.



(a) Using '+*pb*' or '+*npb*' mode in each clustering algorithm.



(b) Using JCD-type or AVG-type *similar-to* links in each clustering algorithm.

Figure 10. The *Diverge* metric results of each kind of groups on the *ACL-all* and *AI-all* datasets when using AVG-type sim-SLNs to rank nodes and using *Abstract* as standard summaries.

Figure 10 shows the *Diverge* metric of each kind of group on the *ACL-all* and *AI-all* datasets, where the series *Seq+pb(2)* is short for *Diverge*($R_{Seq+pb(2)}||R_s$), and so on. We can see from Figure 10(a) that the *Seq*, *CE*, *SeqSim* and *CESeqSim* algorithms have different performance under the '+*pb*' mode and the '+*npb*' mode:

- For the *Seq* algorithm, the groups generated under the '+*pb*' mode get higher *Diverge* results than the groups generated under the '+*npb*' mode. This shows that when only using the *sequential* links to generate groups, the paragraph-bounded restriction can help reduce the cases that unrelated sentences within consecutive paragraphs are clustered into the same group.
- For the *CE* algorithm, there is almost no difference between the '+*pb*' mode and the '+*npb*' mode. This is because only 2.73% *cause-effect* links within the *ACL-all* and *AI-all* datasets cross paragraphs, and fewer of these paragraph-crossing *cause-effect* links are among the top-ranked groups that are used to compose summaries.
- For the *SeqSim* and *CESeqSim* algorithms, the groups generated under the '+*npb*' mode get higher *Diverge* results than the groups generated under the '+*pb*' mode. This phenomenon can be explained in two aspects. (1) Using *sequential* and *similar-to* links simultaneously has a similar effect as the paragraph-bounded restriction because the first sentence of a paragraph is often dissimilar to the last few sentences of the preceding paragraph. (2) When the first sentence of a paragraph is similar to the last few sentences of the preceding paragraph, this sentence usually plays a role in connecting the preceding paragraph to the paragraph to which it belongs. This makes sense to cluster the last few sentences of the preceding paragraph with this connective sentence into the same group because they usually talk about the same theme.

From *SeqSim* and *CESeqSim* in Figure 10(a) and from Figure 10(b), we can see that using JCD-type *similar-to* links gets higher *Diverge* results than using AVG-type *similar-to* links for the five clustering algorithms that need *similar-to* links. This shows that the lexical-based *similar-to* link outperforms the embedding-based *similar-to* link in generating groups.

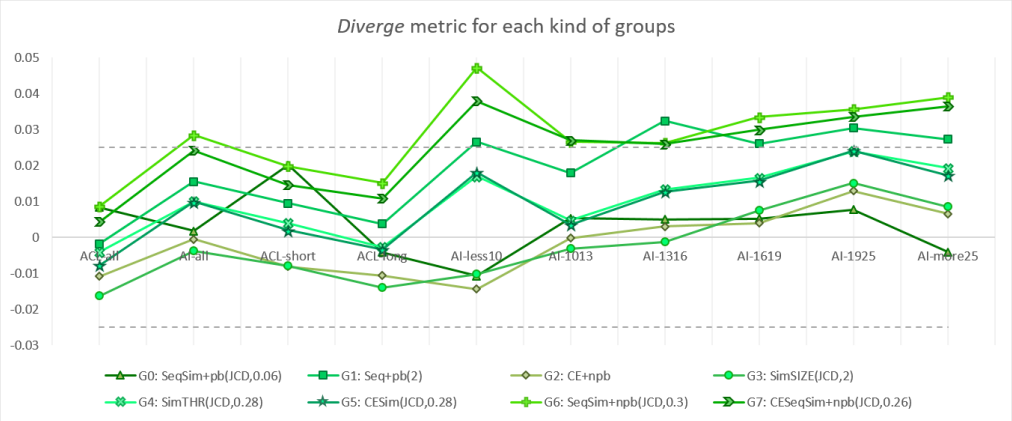


Figure 11. The *Diverge* metric results of different kinds of groups on all datasets when using AVG-type sim-SLNs to rank nodes and using *Abstract* as standard summaries.

Figure 11 shows the *Diverge* metric of using each clustering algorithm when the paragraph-bounded restriction and the type of *similar-to* links are suitable. The series named as *G0: SeqSim+pb(JCD, 0.06)* shows the *Diverge* results of the clustering algorithm we used in Section 3.4.1 and 3.4.2. As shown in Figure 11, it is a baseline to show the *Diverge* results of other clustering algorithms (or the *SeqSim* algorithm under different parameters). From Figure 11 we can see that:

- G6 is superior to G1, G3 and G4. This shows that the *sequential* link and the *similar-to* link are quite suitable for being used together to generate groups.
- There is no big difference between G4 and G5 or between G6 and G7, and the *Diverge* results of G2 are relatively lower than many other clustering algorithms. As shown in Figure 2, clustering sentences that convey a *cause-effect* link into a group can reduce incorrect or incomplete *cause-effect* links in the generated summary. This suggests that the *cause-effect* link can improve the readability of the *group*-based summary without reducing the number of key words/phrases contained in the summaries.
- G4 is significantly better than G3. This shows that, when using the *similar-to* link to generate groups, it is better not to limit the number of sentences within each group but to limit the degree of similarity between sentences within each group.
- In previous experiments, we use G0 to generate groups and draw the conclusion that "*group* outperforms *sentence* and *paragraph* in producing extractive summaries". However, both G6 and G7 outperform G0, so this conclusion still holds when using G6 or G7 to generate groups.
- Comparing the *Diverge* results of G6 or G7 on each sub-dataset, we can find that the *group*-based summary becomes significantly better than the *sentence*-based summary when the source text longer than the average length of papers in *AI-less10*. Thus, the conclusion that "summaries composed by *group* contain more core ideas especially when the average length of source texts is longer than 7,000 words" still holds when using G6 or G7 to generate groups.

Based on the above analysis, the *SeqSim*+*npb*(*JCD*) algorithm and the *CESeqSim*+*npb*(*JCD*) algorithm are the best of the seven clustering algorithms for generating groups. Accordingly, we propose the following strategies for generating groups using the *is-part-of*, *sequential*, *similar-to* and *cause-effect* links:

Strategy 1: *The is-part-of link is a suitable clustering constraint when only using the sequential link to generate groups, while in other cases of using the is-part-of link as a clustering constraint has no effect or even has an adverse effect for generating groups.*

Strategy 2: *The lexical-based similar-to link is more suitable for generating groups, while the embedding-based similar-to link is more suitable for ranking groups.*

Strategy 3: *When only using the similar-to link to generate groups, it is better not to restrict the number of sentences within each group but to restrict the degree of similar-to between sentences within each group.*

Strategy 4: *The sequential link is well suitable for being used together with the similar-to link for generating groups.*
Strategy 5: *The cause-effect link improves the readability of the group-based summary without reducing the amount of key words/phrases contained in the summaries*.

The above five strategies complement our proposed framework when the four types of semantic links are used in clustering algorithms.

Appendix C.3 gives more experiment results to support the above strategies.

4. Related Works

The Semantic Link Network (SLN) was initially proposed for organizing and operating Web resources in a semantic space [11, 12]. It has been developed as a systematic theory and method for representing and operating the semantic structure of various complex systems [5, 13, 14]. An instance model of SLN mainly contains semantic nodes, semantic links, rules on semantic links, and operations on nodes and links. Its nodes represent categories of resources and links between nodes represent the semantic relations. Different from the traditional Semantic Net, SLN emphasizes on self-organized "link", on the basic self-organization operations of a complex system, on the emerging semantics [13], and on the automatic discovery of semantic links. The theory and method of SLN have been applied to various application areas, such as building and maintaining Peer-to-Peer networks [15, 16], discovering and representing Knowledge Flow [17], supporting Cyber-Physical-Social Intelligence [18-21], and serving as a methodology for extractive or abstractive text summarization or even for multimedia summarization [1, 7, 10, 22-24].

Summarizing citations can be regarded as a kind of group-based extractive summarization, which can also generate coherent and readable summaries. An approach to automatically generating related work through summarizing citations was proposed [22]. But this kind of work is only suitable for scientific papers with citations. The approach proposed in this paper is a general framework that can be applied to not only scientific paper but also other types of texts (especially for those texts with length ranging from 7,000 words to 17,000 words). Reference [23] proposes an extractive text summarization method that constructs SLNs with different types of nodes and links and then performs reinforcement ranking on these SLNs, which verifies the effectiveness of SLN in rendering the core of texts. Reference [7] further verifies the role of SLN in representing the core of scientific papers by investigating the distribution of *cause-effect* links, the statistics of keywords within cause-effect sentences, and the improvement of the summarization model by adding *cause-effect* links [23]. Although these two models construct hierarchical SLNs with different language units such as *word*, *sentence*, *paragraph* and *section*, only *sentence* is the target language unit for extraction while other language units are used for reinforcing the ranking of sentences through *is-part-of* links in each step of the iteration.

Most previous extractive summarization models also extract isolated sentences to generate a summary, such as graph-based models [6, 25-28], matrix factorization based models [29, 30], machine learning based models [31-35], neural network based models [36, 37] and so on.

Based on the frame semantics [38], Semantic Role Labelling (SRL) is proposed as a phrase-level semantic analysis task [39]. It is to determine "*who* did *what* to *whom where when how* and *why*" from a sentence. So it is more suitable for analyzing event-based texts like news. It is limited in ability to infer rich semantic relations between larger language units such as sentences and paragraph. For extractive text summarization application, it is hard to process semantic interdependence such as the cause-effect link between sentences when extracting sentences to compose a summary.

With the development of deep learning, applying deep learning to text summarization become a new approach. Deep learning techniques are especially suitable for processing images. An extractive summarization approach to summarizing multi-media based on deep learning was proposed [40]. Deep learning can also be used to discover semantic link within texts through training process.

Some extractive summarization models cluster sentences into sets according to categories, topics [41-43], similarity [44-46] and adjacency between sentences [47]. Reference [44] aims to generate a citation-based summary for single scientific papers. It first collects suitable citation sentences from other papers that cite the target paper as input, then classifies each citation sentence into five categories (*background*, *problem statement*, *method*, *results*, and *limitations*) by a Support Vector Machine classifier, next clusters citation sentences in the same category by performing a community finding on the cosine similarity graph of these sentences, and finally ranks the sentences within each cluster and extracts the top-ranked sentence in each cluster to generate the summary. Some approaches are for enhancing the coverage of topics in the generated summaries. The model introduced in [41] first clusters sentences into topical groups according to the mutual

reinforcement of term-sentence and the sequences of sentences in the source text, and then selects sentences with the highest saliency scores in each topical group to compose a summary. The model introduced in [42] divides sentences into topical clusters by mutual reinforcements of both terms-sentences and sentences-documents, and ranks topic clusters as well as sentences in each cluster to organize the top-ranked sentences selected from each cluster. Sentences are clustered into topical clusters by performing three classical clustering algorithms on the cosine similarity graph of sentences, then the sentence-to-cluster relation is used to construct a two-layer link graph, and finally a conditional Markov random walk model or a HITS model is performed on the two-layer link graph to rank all sentences [43]. Some approaches aim to decrease redundancy in multi-document summarization by clustering sentences into groups and extracting top-ranked sentences in each group. The model introduced in [45] first generates a summary for every single document, and then cluster sentences of the generated summaries according to syntactic similarity and semantic similarity between sentences. The model introduced in [46] creates a graph with four kinds of links (lexical similarity, semantic similarity, co-reference, and discourse relation) connecting all sentences of the input documents, then performs TextRank on this graph to find the leader sentences which act as the core of clusters, and then uses the Dijkstra's algorithm to cluster each sentence to the nearest cluster. Two sentence-extraction tasks for solving the feature sparseness problem were introduced in [47]. It combines two consecutive sentences into a bi-gram pseudo sentence to enhance the statistical features for selecting salient bi-gram pseudo sentences, separates each selected bi-gram pseudo sentence into two sentences, and selects the salient sentences for producing the final summary. In the above works, *sentence* is the only language unit for generating summaries.

Our previous work [10] preliminarily studied the performance of the language unit *group* for extractive summarization. This paper significantly extends the work. Appendix E summarizes the aspects of extensions.

5. Conclusion

This paper proposes a *group*-based extractive text summarization framework by using *group* to replace *sentence* as a basic language unit to generate summaries for the first time. It is based on the assumption that summaries composed by larger language units that retain semantic relations between sentences naturally have higher readability. Experiments show that the proposed approach is effective in summarizing texts.

Research also reaches the following results:

- S*ummaries composed by group or paragraph tend to contain more key words or phrases than summaries composed by sentences*.
- S*ummaries composed by group contain more key words or phrases than those based on paragraph, especially when the average length of source texts is from 7,000 words to 17,000 words*.
- Adopting different clustering algorithms can affect the formation of groups, which in turn influences the *group*-based summary. We suggest the following strategies when the following four types of semantic links are adopted*: 1) the is-part-of link is a suitable clustering constraint when only using the sequential link to generate groups, while in other cases of using the is-part-of link as a clustering constraint has no effect or even has adverse effect for generating groups; 2) the lexical-based similar-to link is more suitable for generating groups, while the embedding-based similar-to link is more suitable for ranking groups; 3) when only using the similar-to link to generate groups, it is better not restrict the number of sentences within group but restrict the degree of similar-to link between sentences within group; 4) the sequential link is suitable to be used together with the similar-to link for generating groups; and, 5) the cause-effect link can improve the readability of the group-based summary without reducing the number of key words/phrases contained in the summaries*.

This work makes a significant contribution to extractive text summarization and it also verifies the role of semantic links in representing and understanding texts, which are the basis for text summarization.

Acknowledgement

Professor Hai Zhuge is the corresponding author of this paper.

References

[1] H. Zhuge, Multi-Dimensional Summarization in Cyber-Physical Society, Morgan Kaufmann, 2016.

[2] M. Gambhir, V. Gupta, Recent automatic text summarization techniques: a survey, Artificial Intelligence Review, 47 (2017) 1-66.

[3] V. Gupta, G.S. Lehal, A survey of text summarization extractive techniques, Journal of emerging technologies in web intelligence, 2 (2010) 258-268.

[4] S. Lappin, H.J. Leass, An algorithm for pronominal anaphora resolution, Computational Linguistics, 20 (1994) 535-561.

[5] H. Zhuge, The Knowledge Grid: Toward Cyber-Physical Society, World Scientific Publishing Co., 2004 (1st edition), 2012 (2nd edition).

[6] R. Mihalcea, P. Tarau, TextRank: Bringing order into text, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2004, pp. 404-411.

[7] M. Cao, X. Sun, H. Zhuge, The contribution of cause-effect link to representing the core of scientific paper -- The role of Semantic Link Network, PloS one, 13 (2018) e0199303.

[8] C.-Y. Lin, Rouge: A package for automatic evaluation of summaries, in: Proceedings of the ACL-04 Workshop on Text Summarization Branches Out, Barcelona, Spain, 2004, pp. 74-81.

[9] S. Kullback, R.A. Leibler, On information and sufficiency, Annals of Mathematical Statistics, 22 (1951) 79-86.

[10] M. Cao, H. Zhuge, What size of language unit is more appropriate for text summarization?, in: Proceedings of the 14th International Conference on Semantics, Knowledge and Grids (SKG), IEEE, Guangzhou, China, 2018, pp. 196-202.

[11] H. Zhuge, L. Zheng, N. Zhang, X. Li, An automatic semantic relationships discovery approach, in: Proceedings of the 13th International World Wide Web Conference, ACM, 2004, pp. 278-279.

[12] H. Zhuge, The Web Resource Space Model, Springer, 2008.

[13] H. Zhuge, Communities and emerging semantics in Semantic Link Network: Discovery and learning, IEEE Transactions on Knowledge and Data Engineering, 21 (2009) 785-799.

[14] H. Zhuge, Interactive semantics, Artificial Intelligence, 174 (2010) 190-204.

[15] H. Zhuge, J. Liu, L. Feng, X. Sun, C. He, Query routing in a P2P Semantic Link Network, Computational Intelligence, 21 (2005) 197-216.

[16] H. Zhuge, X. Li, Peer-to-Peer in metric space and semantic space, IEEE Transactions on Knowledge and Data Engineering, 19 (2007) 759-771.

[17] H. Zhuge, Discovery of knowledge flow in science, Communications of the ACM, 49 (2006) 101-107.

[18] H. Zhuge, Socio-natural thought Semantic Link Network: A method of semantic networking in the Cyber Physical Society, in: Proceedings of 24th IEEE International Conference on Advanced Information Networking and Applications, IEEE, 2010, pp. 19-26.

[19] H. Zhuge, Semantic linking through spaces for Cyber-Physical-Socio Intelligence: A methodology, Artificial Intelligence, 175 (2011) 988-1019.

[20] H. Zhuge, Y. Xing, Probabilistic Resource Space Model for managing resources in Cyber-Physical Society, IEEE Transactions on Service Computing, 5 (2012) 404-421.

[21] H. Zhuge, Cyber-Physical-Social Intelligence on Human-Machine-Nature Symbiosis, Springer, 2019.

[22] J. Chen, H. Zhuge, Automatic generation of related work through summarizing citations, Concurrency and Computation: Practice and Experience, 31 (2019).

[23] X. Sun, H. Zhuge, Summarization of scientific paper through reinforcement ranking on Semantic Link Network, IEEE Access, 6 (2018) 40611-40625.

[24] W. Li, H. Zhuge, Abstractive multi-document summarization based on Semantic Link Network, IEEE Transactions on Knowledge and Data Engineering, (2019) DOI: 10.1109/TKDE.2019.2922957.

[25] G. Erkan, D.R. Radev, LexRank: Graph-based lexical centrality as salience in text summarization, Journal of Qiqihar Junior Teachers College, 22 (2004) 457-479.

[26] X. Zhang, G. Cheng, Y. Qu, Ontology summarization based on rdf sentence graph, in: Proceedings of the 16th international conference on World Wide Web, ACM, 2007, pp. 707-716.

[27] L. Antiqueira, O.N. Oliveira, L.d.F. Costa, M.d.G.V. Nunes, A complex network approach to text summarization, Information Sciences, 179 (2009) 584-599.

[28] E. Baralis, L. Cagliero, N. Mahoto, A. Fiori, GraphSum: Discovering correlations among multiple terms for graph-based summarization, Information Sciences, 249 (2013) 96-109.
[29] Y. Gong, X. Liu, Generic text summarization using relevance measure and latent semantic analysis, in: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, New Orleans, Louisiana, USA, 2001, pp. 19-25.
[30] J.-H. Lee, S. Park, C.-M. Ahn, D. Kim, Automatic generic document summarization based on non-negative matrix factorization, Information Processing & Management, 45 (2009) 20-34.
[31] D. Shen, J.T. Sun, H. Li, Q. Yang, Z. Chen, Document summarization using conditional random fields, in: Proceedings of the International Joint Conference on Artifical Intelligence, Morgan Kaufmann Publishers Inc., 2007, pp. 2862-2867.
[32] Y. Ouyang, W. Li, R. Zhang, S. Li, Q. Lu, A progressive sentence selection strategy for document summarization, Information Processing & Management, 49 (2013) 213-221.
[33] M.A. Fattah, A hybrid machine learning model for multi-document summarization, Applied Intelligence, 40 (2014) 592-600.
[34] M. Mendoza, S. Bonilla, C. Noguera, C. Cobos, E. León, Extractive single-document summarization based on genetic operators and guided local search, Expert Systems with Applications, 41 (2014) 4158-4169.
[35] L. Li, K. Zhou, G.R. Xue, H. Zha, Y. Yu, Enhancing diversity, coverage and balance for summarization through structure learning, in: Proceedings of the 18th International Conference on World Wide Web, 2009, pp. 71-80.
[36] M.A. Fattah, F. Ren, GA, MR, FFNN, PNN and GMM based models for automatic text summarization, Computer Speech & Language, 23 (2009) 126-144.
[37] Z. Cao, F. Wei, L. Dong, S. Li, M. Zhou, Ranking with recursive neural networks and its application to multi-document summarization, in: Proceedings of the 29th AAAI Conference on Artificial Intelligence, AAAI Press, Austin, Texas, 2015, pp. 2153-2159.
[38] C.J. Fillmore, Frame semantics and the nature of language, in: Annals of the New York Academy of Sciences: Conference on the origin and development of language and speech, 1976, pp. 20-32.
[39] D. Gildea, D. Jurafsky, Automatic labeling of semantic roles, Computational linguistics, 28 (2002) 245-288.
[40] J. Chen, H. Zhuge, Extractive summarization of documents with images based on multi-modal RNN, Future Generation Computer Systems, 99 (2019) 186-196.
[41] H. Zha, Generic summarization and keyphrase extraction using mutual reinforcement principle and sentence clustering, in: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, Tampere, Finland, 2002, pp. 113-120.
[42] L. Yang, X. Cai, Y. Zhang, P. Shi, Enhancing sentence-level clustering with ranking-based clustering framework for theme-based summarization, Information Sciences, 260 (2014) 37-50.
[43] X. Wan, J. Yang, Multi-document summarization using cluster-based link analysis, in: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, ACM, Singapore, 2008, pp. 299-306.
[44] A. Abu-Jbara, D.R. Radev, Coherent citation-based summarization of scientific papers, in: Proceedings of the Association for Computational Linguistics, 2011, pp. 500-509.
[45] V.K. Gupta, T.J. Siddiqui, Multi-document summarization using sentence clustering, in: Proceddings of the 4th International Conference on Intelligent Human Computer Interaction (IHCI), Kharagpur, 2012, pp. 1-5.
[46] R. Ferreira, L.d.S. Cabral, F. Freitas, R.D. Lins, G.d.F. Silva, S.J. Simske, L. Favaro, A multi-document summarization system based on statistics and linguistic treatment, Expert Systems with Applications, 41 (2014) 5780-5787.
[47] Y. Ko, J. Seo, An effective sentence-extraction technique using contextual information and statistical approaches for text summarization, Pattern Recognition Letters, 29 (2008) 1366-1371.

Appendix A. Six Metrics for Calculating the Similarity between Text Segments

A.1 Lexical-based Similarity

If two text language units have many words in common, then they are likely to talk about the same thing. The similarity based on the common words is called the lexical-based similarity. Jaccard distance (JCD) can be used to calculate the proportion of common words in two language units.

Let s be a sentence, g be a group, p be a paragraph and $u \in \{s, g, p\}$. Let $u_i = \{w_{i,1}, w_{i,2}, \dots w_{i,m}\}$ be a language unit containing m words, and $u_j = \{w_{j,1}, w_{j,2}, \dots w_{j,n}\}$ be another language unit containing n words. As shown in formula (1), the JCD-type similarity between u_i and u_j is calculated by dividing the number of common words to the total number of words.

$$JCDsim(u_i, u_j) = \frac{\left|\{w_{i,1}, \dots, w_{i,m}\} \cap \{w_{j,1}, \dots, w_{j,m}\}\right|}{m + n} \quad (1)$$

We remove stop words from two language units before calculating their similarity, because stop words frequently appear in many sentences and convey no meaning. The number of common words is counted by the exact matching of words.

A.2 Embedding-based Similarity

Word embedding (also known as word representation) has become widely used in various Natural Language Processing tasks. Word embedding models represent each word as a continuous vector by capturing the contexts of this word in a large corpus. The similarity between words can be easily measured by the distance between the vectors of the corresponding words. In the same way, the similarity between language units can be computed by first embedding language units into vectors and then calculating the distance between the embedding vectors.

The GloVe is an unsupervised learning algorithm for obtaining vectors of words by aggregating word-word co-occurrence statistics from a corpus [48], and it has become a standard approach for embedding words. However, there is no standard approach for embedding sentences, groups or paragraphs yet. We implement two unsupervised models and one pre-trained model in this paper for embedding language units into vectors.

- **Average (AVG)**

  AVG is an unsupervised model that simply takes the average of the word embedding vectors as the embedding of a language unit. For the language unit $u_i = \{w_{i,1}, w_{i,2}, \dots w_{i,m}\}$, let $\boldsymbol{wv}_{i,k}$ be the embedding vector of the word $w_{i,k}$ $(k \in \{1,2, \dots m\})$, the AVG-type embedding vector of u_i can be calculate by the formula (2).

$$\boldsymbol{uv}_{i_{AVG}} = \frac{\sum_{k=1}^{m} \boldsymbol{wv}_{i,k}}{m} \quad (2)$$

  The $\boldsymbol{wv}_{i,k}$ is initialized by the publicly available 300-dimensional GloVe vectors[1]. Stop words and words that have no embedding vector are removed from u_i before embedding.

- **Smooth Inverse Frequency (SIF)**

  SIF is another unsupervised model proposed in [49]. SIF model first computes the weighted average of the word embedding vectors as the initial embedding vector of each sentence, and then modifies these initial embedding vectors with the Principal Component Analysis to get the final embedding vector of each sentence. The experimental results in [49] show that this model improves about 10% to 30% than some baseline models, and can even beat some sophisticated supervised methods including RNN-based method and LSTM-based method.

---

[1] The package named as 'glove.840B.300d.zip' is downloaded from https://nlp.stanford.edu/projects/glove/.

Let $D = \{u_1, u_2, \dots, u_t\}$ be a source text that contains t language units, $u_i = \{w_{i,1}, w_{i,2}, \dots w_{i,m}\}$ be the i^{th} language unit in D, and $\boldsymbol{wv}_{i,k}$ be the embedding vector of word $w_{i,k}$. Let $freq(w_{i,k})$ be the frequency of the word $w_{i,k}$, which is obtained by counting the word within a dataset of English Wikipedia[2] in our implementation. Let $\boldsymbol{uv}_{i_{SIF\#}}$ be the initial SIF-type embedding of u_i, and it is calculated by the formula (3). The constant parameter α in formula (3) is set as 10^{-3} both in [49] and in our implementation.

$$\boldsymbol{uv}_{i_{SIF\#}} = \frac{1}{m}\sum_{k=1}^{m} \frac{\alpha}{freq(w_{i,k}) + \alpha} \boldsymbol{wv}_{i,k} \tag{3}$$

Matrix $\boldsymbol{X}$= $[\boldsymbol{uv}_{1_{SIF\#}}, \boldsymbol{uv}_{2_{SIF\#}}, \dots, \boldsymbol{uv}_{t_{SIF\#}}]$ is formed after getting the initial SIF-type embedding vector for each language unit. Then the Principal Component Analysis is conducted to assign the first singular vector of $\boldsymbol{X}$ to $\boldsymbol{y}$. Finally, the SIF-type embedding vectors for all the language units are calculated by formula (4).

$$[\boldsymbol{uv}_{1_{SIF}}, \boldsymbol{uv}_{2_{SIF}}, \dots, \boldsymbol{uv}_{t_{SIF}}] = \boldsymbol{X} - \boldsymbol{y}\boldsymbol{y}^T \cdot \boldsymbol{X} \tag{4}$$

- **Universal Sentence Encoder (USE)**

  USE model is based on the deep averaging network where the embedding vectors for words and bi-grams are averaged together and passed through a feed-forward deep neural network [50]. This model has been pre-trained and made freely available on Tensorflow Hub[3]. We can easily obtain the embedding vector of any language unit by loading and calling this open-source package.

  After we get the embedding vectors of two language units u_i and u_j, the similarity between them can be measured by formula (5), which is a variant of cosine similarity to ensure the similarity value to be positive. Note that $\boldsymbol{uv}_i$ and $\boldsymbol{uv}_i$ in formula (5) should be the same type of embedding vectors. We name the type of similarity between u_i and u_j as AVG-type if AVG model is used for calculating $\boldsymbol{uv}_i$ and $\boldsymbol{uv}_j$. The SIF-type similarity and the USE-type similarity can be got in a similar way.

$$sim(u_i, u_j) = \frac{1}{2} \times \left(\frac{\boldsymbol{uv}_i \cdot \boldsymbol{uv}_j}{|\boldsymbol{uv}_i| \times |\boldsymbol{uv}_j|} + 1 \right) \tag{5}$$

A.3 Synsets-based Similarity

Many English words have more than one meaning. However, both the lexical-based similarity and the embedding-based similarity introduced above cannot detect the influence of polysemy on the similarity of two language units. The lexical-based similarity is based on the morphological matching of words without distinguishing the senses of words in different contexts. The embedding-based similarity is based on the word embedding. Although the word embedding tries to represent the sense of a word by the distances from this word to other words in a vector space, a single vector cannot represent different senses of a word at the same time. For example, if the embedding vector of 'apple' is close to the embedding vectors of 'company', 'products', 'computer' and 'smartphone', then the embedding vector of 'apple' should not be close to the embedding vectors of 'fruit', 'banana', 'red' and 'sweet'. Otherwise, the concept of 'electronic equipment manufacturer' and the concept of 'fruit' will be close in the embedding vector space, meaning that these two concepts are similar to each other.

The synsets-based similarity aims to not only reduce the similarity between two language units that contain polysemous words, such as "Mary likes apples" and "Mary likes the Apple Inc.", but also intensify the similarity between two language units that contain the synonyms, such as "this car uses a lot of oil" and "this automobile consumes a lot of petrol".

The synsets-based similarity between two language units is calculated by first assigning a sense to each word, then matching senses of words from the two language units according to the similarity of senses, and finally taking the average similarity of the matched sense pairs as the similarity between the two language units. Figure A-1 shows an example to calculate the synsets-based similarity between two sentences.

The synsets in WordNet are used as the senses of words [51]. Pywsd [52], a WordNet-based open-source word sense disambiguation package, is used to assign a specific synset to each word according to the adapted

---

[2] The English Wikipedia dataset we used is downloaded from https://dumps.wikimedia.org/enwiki/20181101/.

[3] The instructions of USE model can be found at https://www.tensorflow.org/hub/modules/google/universal-sentence-encoder/1

Lesk word sense disambiguation algorithm [53]. As for pairing words from two language units, we select two metrics to measure the similarity between two synsets.

- **Lin**

  This metric is based on information content (IC) of synsets obtained from an extended corpus [54]. Let syn_i be the synset for word w_i, syn_j be the synset for word w_j, and $lcs(syn_i, syn_j)$ be the least common subsume of syn_i and syn_j in the WordNet taxonomy. The LIN metric calculates the similarity between syn_i and syn_j as shown in formula (6).

$$LINsim(syn_i, syn_j) = \frac{2 \times IC(lcs(syn_i, syn_j))}{IC(syn_i) + IC(syn_j)} \tag{6}$$

- **Wu and Palmer (WUP)**

  As shown in formula (7), the WUP metric calculates the similarity between two synsets based on the depth of synsets in the WordNet taxonomy [55].

$$WUPsim(syn_i, syn_j) = \frac{2 \times depth(lcs(syn_i, syn_j))}{depth(syn_i) + depth(syn_j)} \tag{7}$$

Let $[syn_{i,1}, syn_{i,2}, \dots, syn_{i,m}]$ be the synsets list for u_i, and $[syn_{j,1}, syn_{j,2}, \dots, syn_{j,n}]$ be the synsets list for u_j. After getting the similarity for each pair like $(syn_{i,x}, syn_{j,y})$ $(1 \le x \le m, 1 \le y \le n)$ and storing them into a similarity matric *SynM*, we perform the Algorithm 1 to match synsets of u_i and u_j. The similarity between u_i and u_j is the average similarity of the matched synsets. We name the type of the similarity between u_i and u_j LIN-type if the Lin metric is used to calculate the similarity for each synset pair, otherwise name WUP-type if the WUP metric is used.

Algorithm 1 A Greedy Algorithm for Matching Synsets

Input: $SynM$:A m*n matrix stores the similarity scores of synset pairs.
Output: $MatchSyn$: a list of the matched synset pairs.

```
MatchSyn ← [ ]
m, n ← SynM.shape
BaseSide ← 0
if n > m then
    BaseSide ← 1
end if
Visited ← Initiate a list contians max(m, n) False.
SynRank ← Reversely rank elements of SynM in the form like (row, col, value)
for elem in SynRank do
    if not Visited[elem(BaseSide)] then
        MatchSyn.append(elem)
        Visited[elem(BaseSide)] = True
    end if
end for
return MatchSyn
```

Supplementary References

[48] J. Pennington, R. Socher, C. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532-1543.

[49] S. Arora, Y. Liang, T. Ma, A simple but tough-to-beat baseline for sentence embeddings, in: Proceedings of the International Conference on Learning Representations (ICLR), 2017.

[50] D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R.S. John, N. Constant, M. Guajardo-Céspedes, K. Yuan, C. Tar, Y.-H. Sung, B. Strope, R. Kurzweil, Universal sentence encoder, arXiv:1803.11175 [cs.CL], (2018).

[51] G.A. Miller, WordNet: A lexical database for English, Communications of the ACM, 38 (11) (1995) 39-41.

[52] L. Tan, Pywsd: Python implementations of Word Sense Disambiguation (WSD) technologies [software]. Retrieved from https://github.com/alvations/pywsd in GitHub, 2014.

[53] S. Banerjee, T. Pedersen, An adapted lesk algorithm for word sense disambiguation using WordNet, in: Proceedings of international conference on intelligent text processing and computational linguistics, 2002, pp. 136-145.
[54] D. Lin, An information-theoretic definition of similarity, in: Proceedings of the 15th international conference on machine learning, Morgan Kaufmann Publishers Inc., 1998, pp. 296-304.
[55] Z. Wu, M. Palmer, Verbs semantics and lexical selection, in: Proceedings of the 32nd annual meeting on Association for Computational Linguistics (ACL), 1994, pp. 133-138.

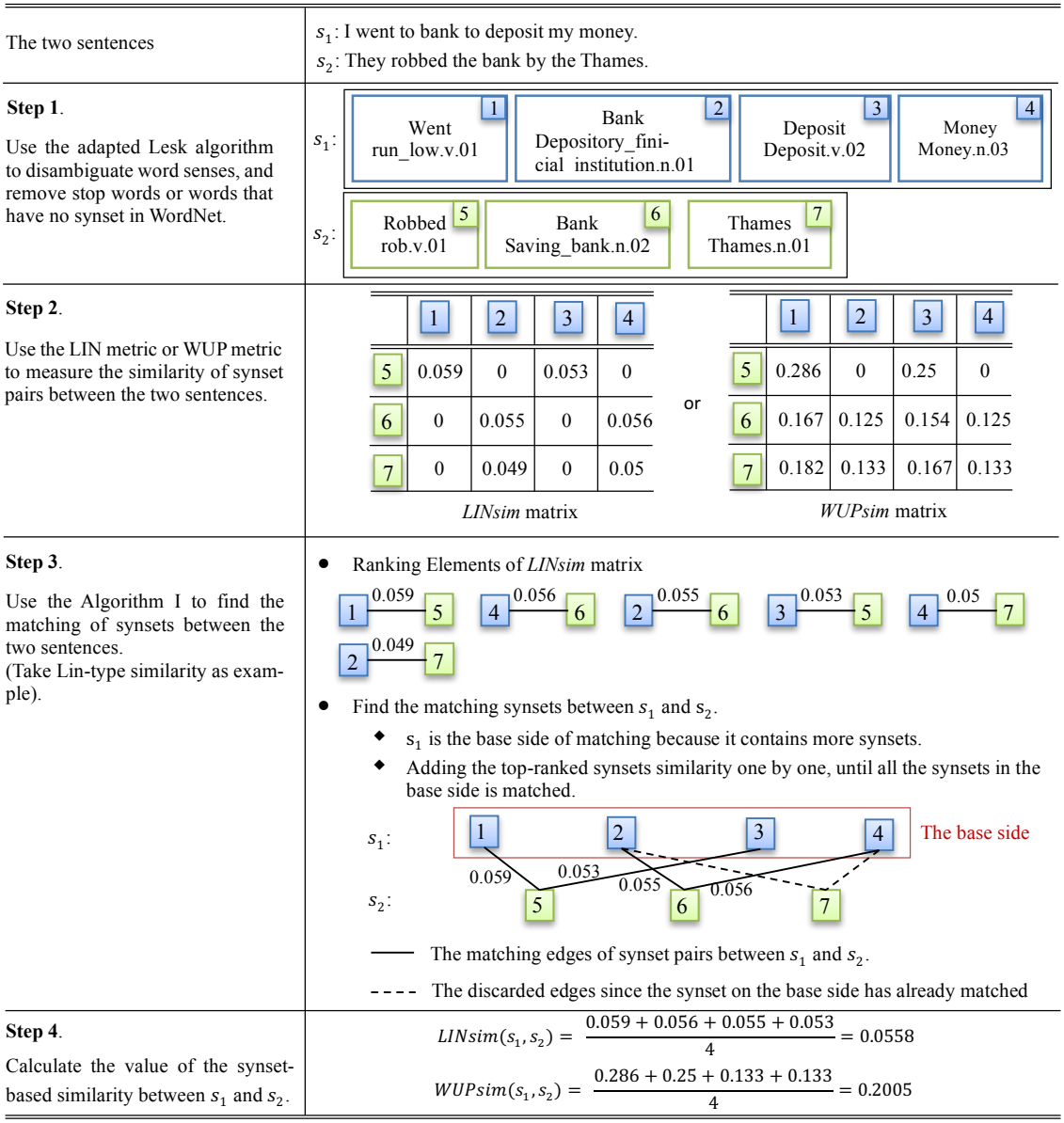


Figure A-1. The example of calculating the synsets-based similarity.

Appendix B. Seven Clustering Algorithms for Generating Groups

Let D be a source text that contains 4 paragraph $p_{1\sim4}$ and 16 sentences $s_{1\sim16}$. If we know s_2 $-cause\rightarrow$ s_1, s_7 $-cause\rightarrow$ s_8 and s_8 $-cause\rightarrow$ s_9 and select the JCD-type similarity metric to calculate the weight of the *similar-to* links, then the first layer SLN for the source text D is constructed as shown in Figure B-1.

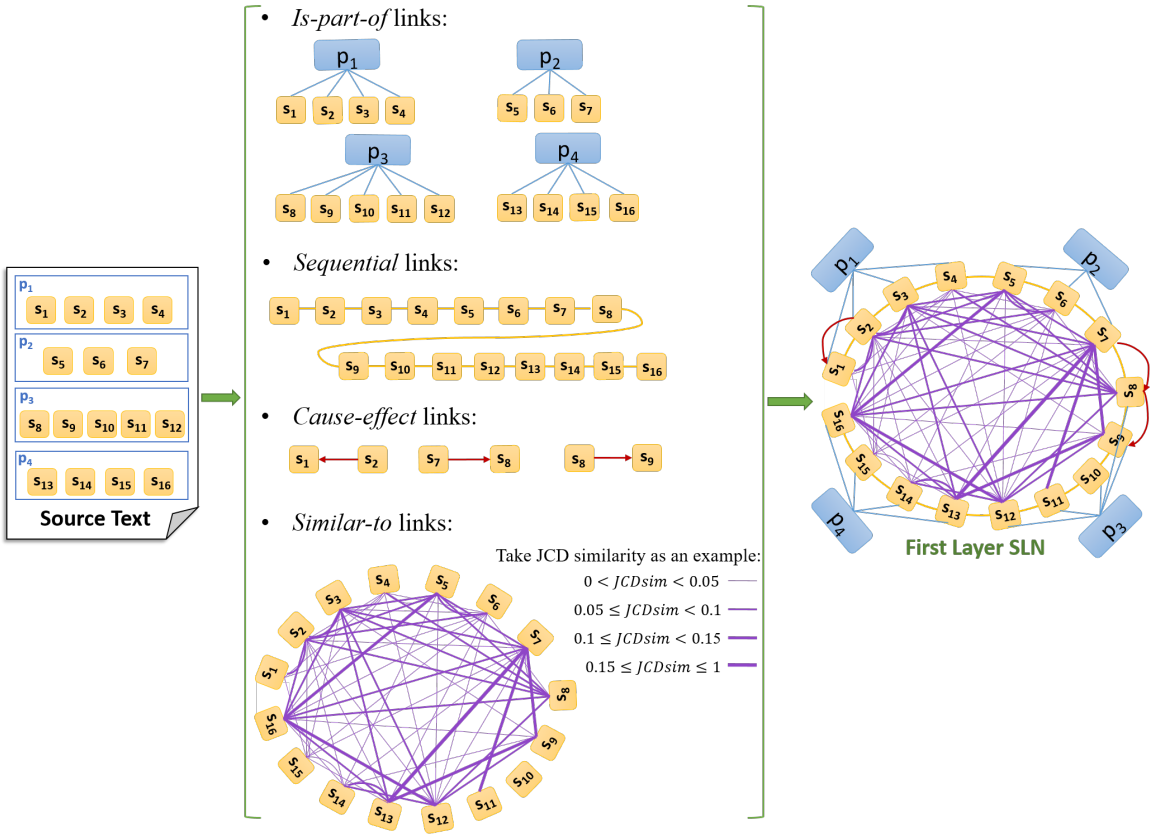


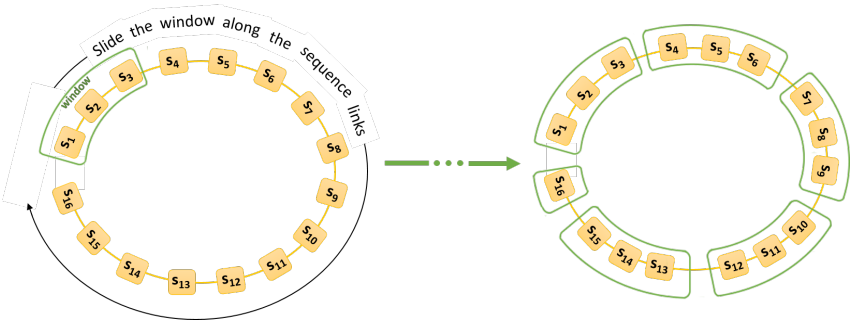
Fig B-1. An example of constructing the first layer SLN for a source text.

Seven clustering algorithms are designed for generating groups according to the combinations of the *is-part-of link*, *sequential link*, *similar-to link,* and *cause-effect* link:

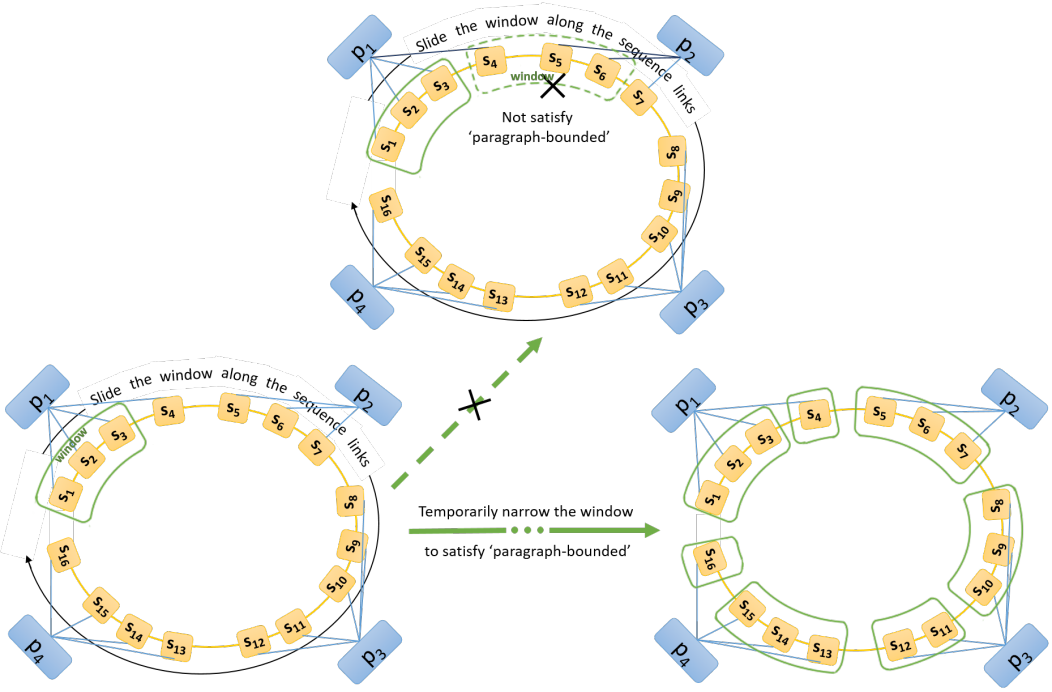
- **Algorithm 1: *Seq+pb/npb (win_size* $\in N^+$*)***

  As shown in Figure B-1, this clustering algorithm slides a window along with the *sequential* links and clusters sentences within the window as a group. The parameter '*win_size*' is a positive integer specifying the number of sentences covered by the sliding window. The '*+pb*' represents the 'paragraph-bounded' mode, in which the *is-part-of* link is used to check the subordination between sentences and paragraphs to ensure that the sentences within a group belong to the same paragraph. In contrast, the '*+npb*' represents the 'not paragraph-bounded' mode, in which sentences from two consecutive paragraphs can be clustered into the same group. Either the '*+pb*' mode or the '*+npb*' mode should be chosen when using this algorithm.

  When composing a summary with top-ranked groups, we just put the first few sentences of the last extracted group into the summary if the group contains too many sentences to be completely put into the summary. Therefore, the order of the sentences within a group affects the sentences contained in the generated summary. However, ordering the sentences within a group differently from the source text disrupts the semantic link between sentences, so we arrange the sentences of each group according to the *sequential* link if it is used to generate groups.



(a) $Seq + npb(win\_size = 3)$



(b) $Seq + pb(win\_size = 3)$

Figure B-2. The clustering process of the $Seq + pb/npb(win\_size = 3)$ algorithm.

- **Algorithm 2: *SimSIZE* ($sim\_type, size \in N^+$)**

  This algorithm clusters sentences into groups of fixed size according to the *similar-to* links between sentences. The parameter 'sim_type' takes one in {JCD, AVG, SIF, USE, LIN, WUP}, referring to the type of the *similar-to* links used to cluster sentences. The parameter '*size*' specifies the maximum number of sentences within a group.

  As shown in Figure B-3, the clustering process of the *SimSIZE* algorithm is iterative. For each iteration, the algorithm first sorts all *similar-to* links in descending order, next checks the *similar-to* link in turn until finding the first *similar-to* link whose nodes contain no more than '*size*' sentences, then merges the two nodes of this *similar-to* link to form a new node, and finally calculates the average *similar-to* links to connect the newly formed node with other nodes. Perform the above iteration until no *similar-to* link can be merged, at which point each node is a group.

  In order to determine the order of sentences within each group, we add an attribute named 'tightness' to each node during the clustering process. The tightness of a node containing only one sentence is set zero, and the tightness of a node containing more than one sentence is equal to the sum of the weights of the *similar-to* links that are used to form this node divided by the number of sentences contained in this node. When merging two nodes to form a new node, we put the sentences of the high tightness node before the sentences of the low tightness node.

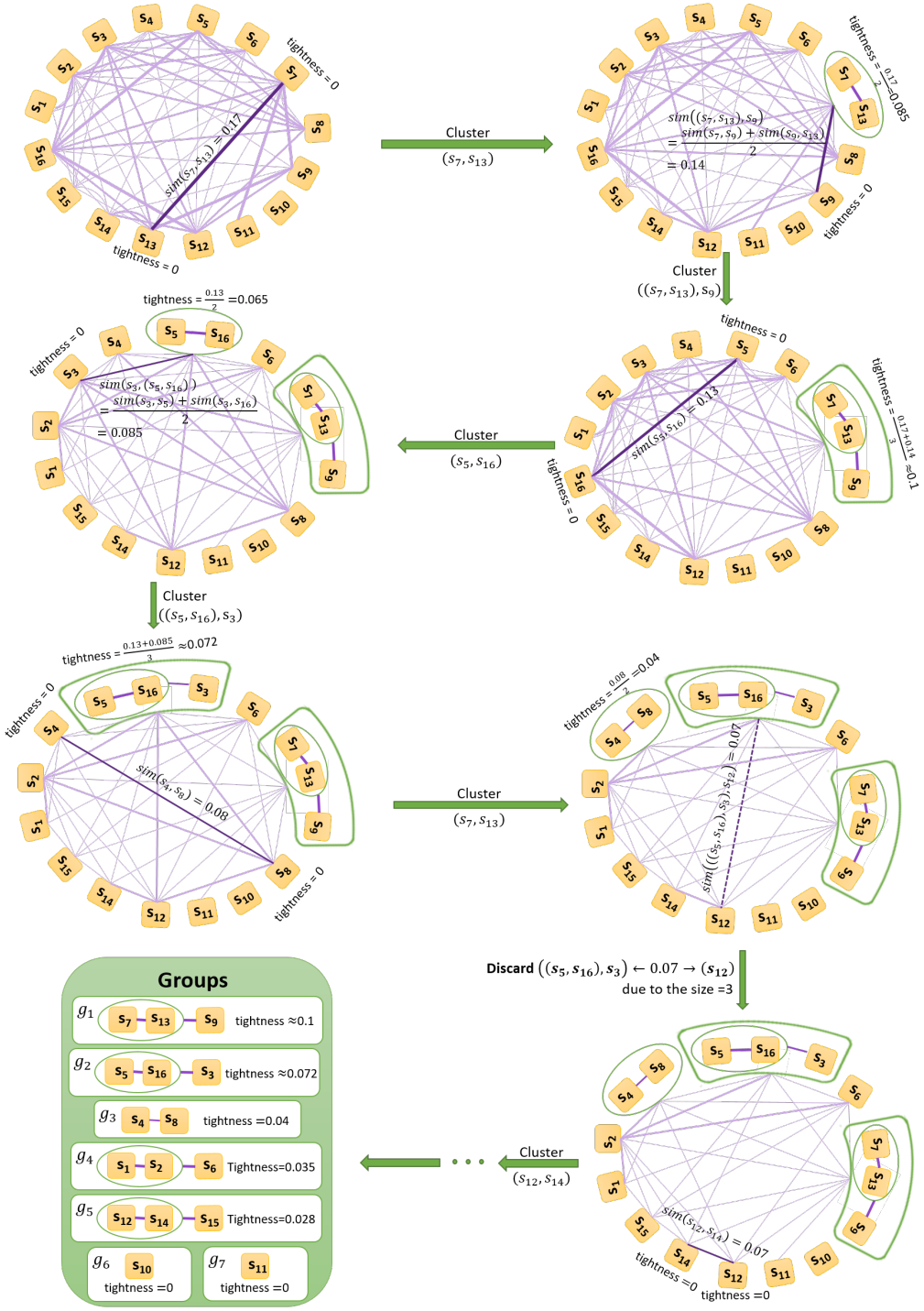


Figure B-3. The clustering process of the $Seq + pb/npb(win\_size = 3)$ algorithm.

- **Algorithm 3: *SimTHR* $(sim\_type, thr \in R^+)$**

  The *SimTHR* algorithm is similar to the *SimSIZE* algorithm, except that it uses a threshold called 'thr' to restrict the weight of the *similar-to* link that is selected to form the new node in each iteration, without limiting the number of sentences in the newly formed node. The iteration will be stopped when there is no *similar-to* link whose weight is larger than the threshold 'thr', at which point each node is a group. Figure B-4 shows the clustering process of the *SimTHR* algorithm.

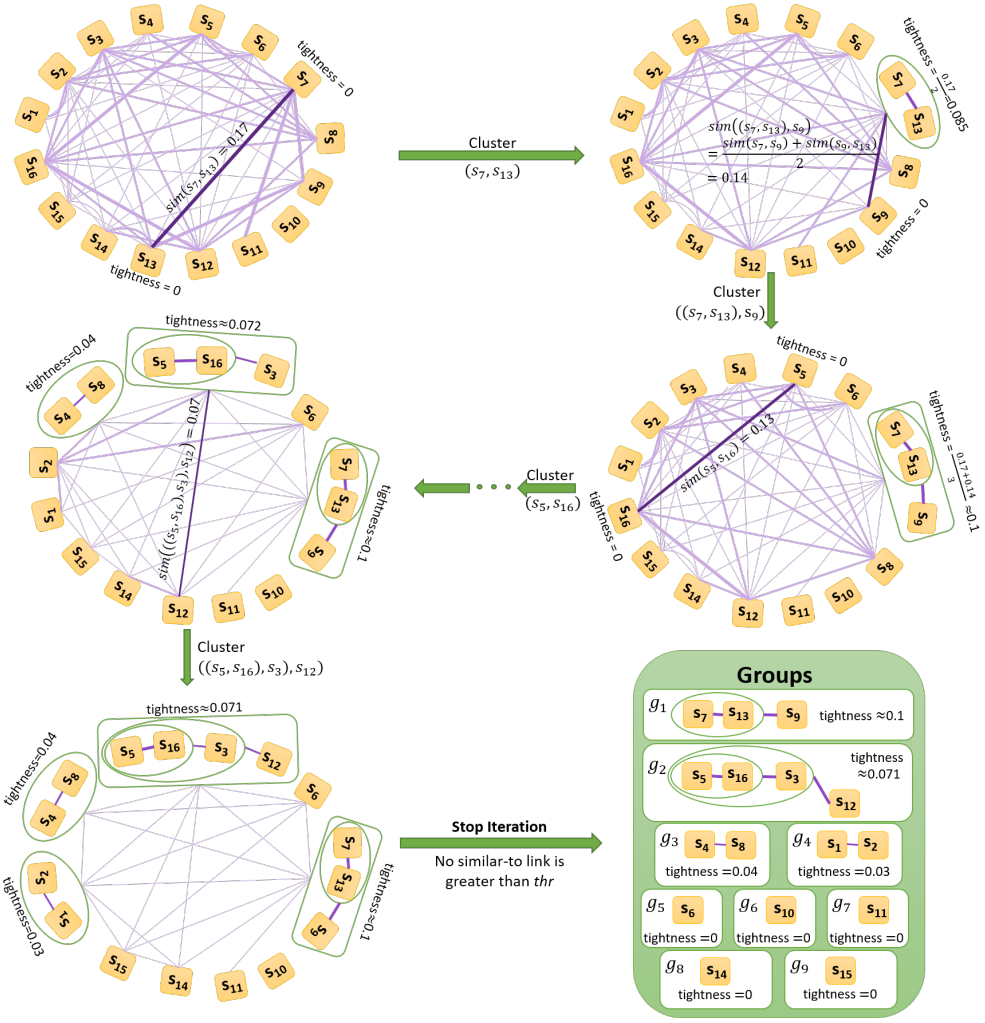
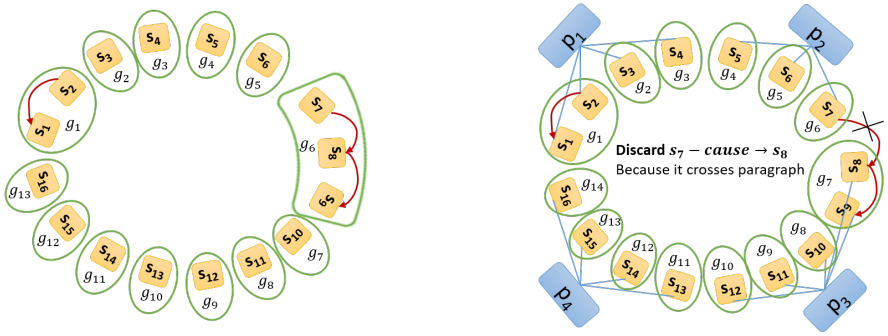


Figure B-4. The clustering process of the $SimTHR(JCD, thr = 0.05)$ algorithm.

- **Algorithm 4: *CE+pb/npb***

  This algorithm clusters sentences that are connected by *cause-effect* links into a group, and views each sentence that is not linked by a *cause-effect* link as a group. In the '*+npb*' mode all the *cause-effect* links are used to cluster sentences into groups, while in the '*+pb*' mode the *cause-effect* links whose sentences do not belong to the same paragraph are discarded before clustering. Figure B-5 shows examples of generating groups with the *CE+pb/npb* algorithm.



(a) *CE+npb* (b) *CE+pb*

Figure B-5. The clustering process of the $CE + pb/npb$ algorithm.

- **Algorithm 5: *CESim(sim_type, thr ∈ R^+)***

As shown in Figure B-6, the *CESim* algorithm first performs the *CE+npb* algorithm to cluster sentences into temporary groups, then connects these temporary groups by averaging the *similar-to* links between sentences, and then performs the *SimTHR* algorithm to cluster temporary groups into the final groups. The '*+npb*' mode is selected for the *CE* algorithm because the paragraph-bounded restriction is useless in the clustering process of the *SimTHR* algorithm.

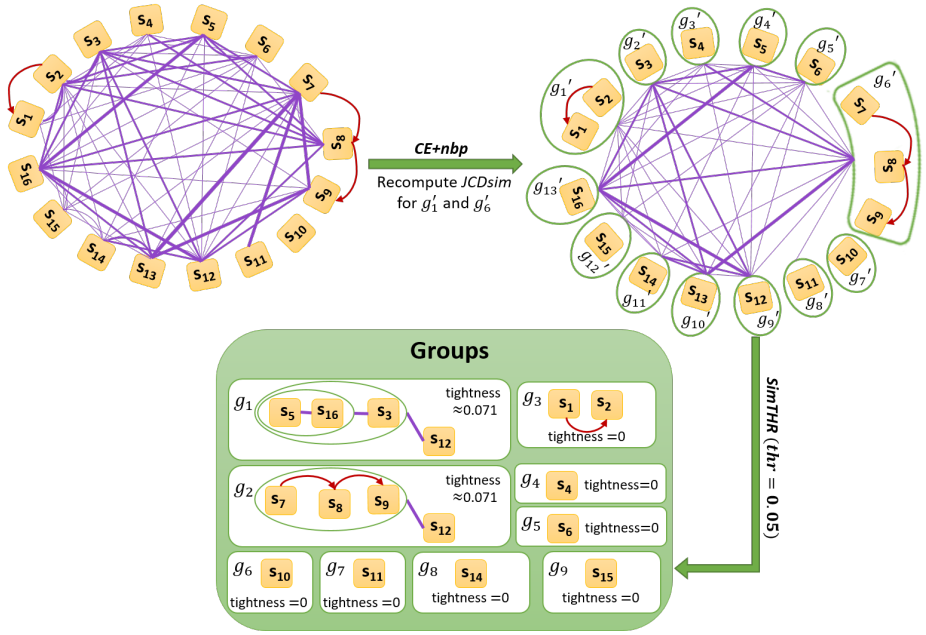


Figure B-6. The clustering process of the $CESim(JCD, thr = 0.05)$ algorithm.

- **Algorithm 6: *SeqSim+pb/npb(sim_type, thr ∈ R^+)***

Figure B-7 and Figure B-8 respectively show the clustering process of the *SeqSim* algorithm in the modes of '*+npb*' and '*+pb*'. At first, the algorithm initializes the first sentence of a source text as a group. Then, from the second sentence, the algorithm checks each sentence along the *sequential* link to determine whether the sentence should be added to the previous group or should be initialized as a new group according to the *similar-to* links between this sentence and the sentences in the previous group. For example, let s_{c+1} be the sentence being checked and $g_b = (s_{c-2}, s_{c-1}, s_c)$ be the group before s_{c+1}. If the weight of a *similar-to* link between s_{c+1} and any sentence in g_b is greater than the threshold 'thr', then s_{c+1} should be appended to g_b. If not, the algorithm initializes s_{c+1} as a new group $g_{b+1} = (s_{c+1})$.

The '*+pb*' mode requires that the sentences within a group belong to the same paragraph, so the $SeqSim + pb$ algorithm initializes the first sentence of each paragraph as a new group regardless of the weights of the *similar-to* links.

- **Algorithm 7: *CESeqSim+pb/npb(sim_type, thr ∈ R^+)***

The *CESeqSim* algorithm is the combination of the *CE* algorithm and the *SeqSim* algorithm. As shown in Figure B-9, it first performs the *CE* algorithm to cluster sentences into temporary groups while remaining the *sequential* links and the *is-part-of* links on these temporary groups, then connects these temporary groups by the average *similar-to* links, and finally performs the *SeqSim* algorithm to cluster the temporary groups into the final groups. Note that the mode of the paragraph-bounded restriction must be consistent in the clustering processes of both the *CE* algorithm and the *SeqSim* algorithm.

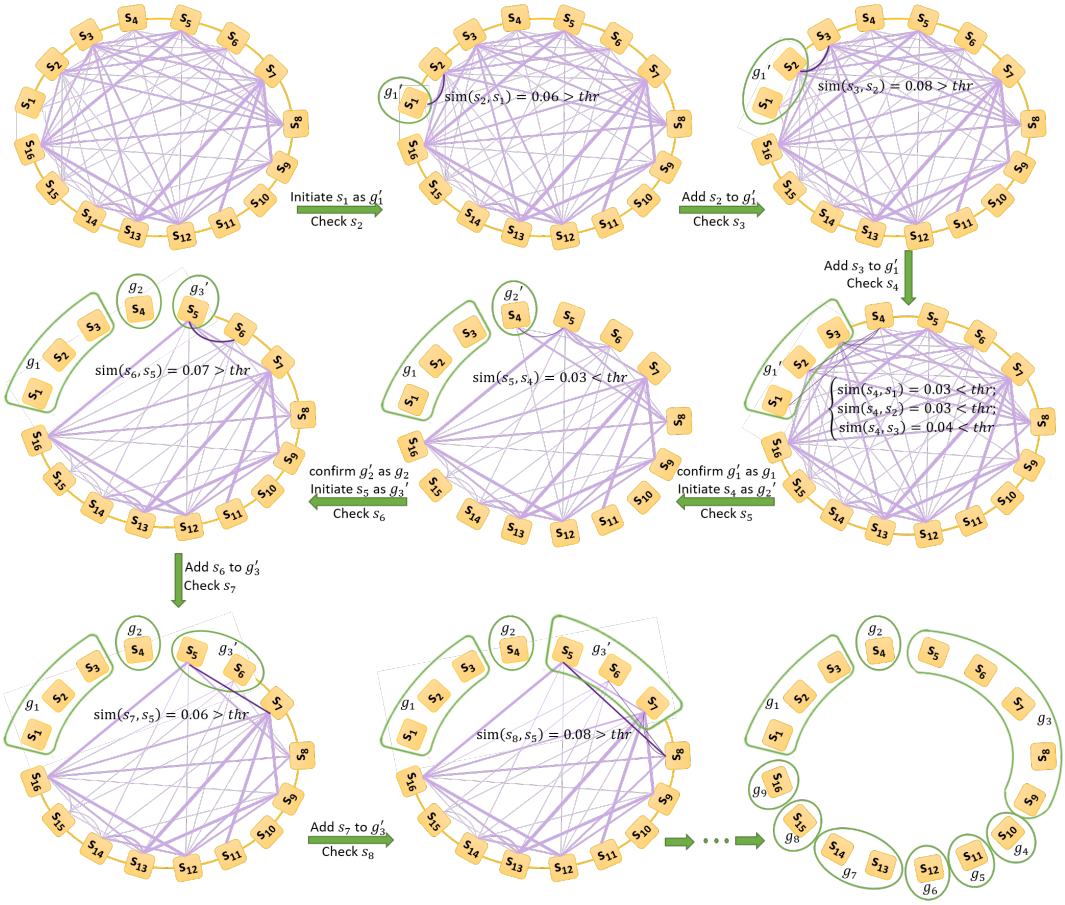


Figure B-7. The clustering process of the $SeqSim + npb(JCD, thr = 0.05)$ algorithm.

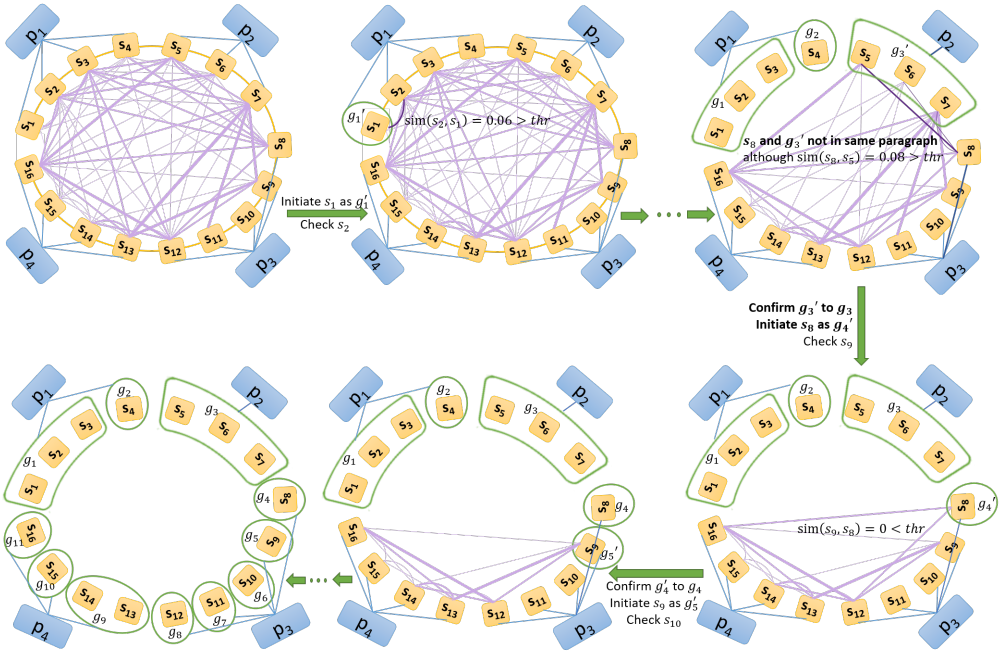
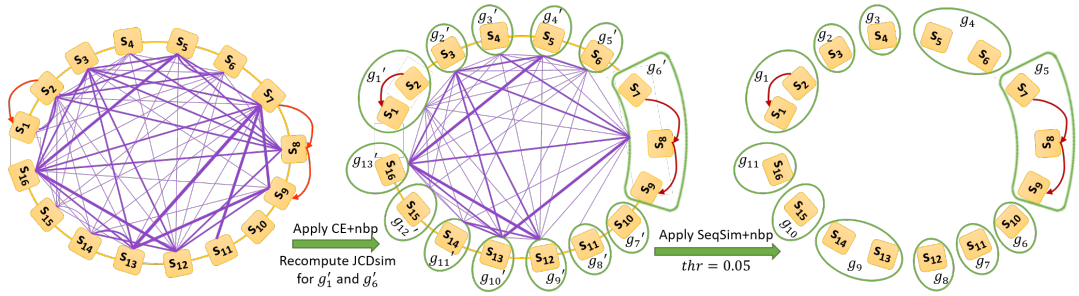
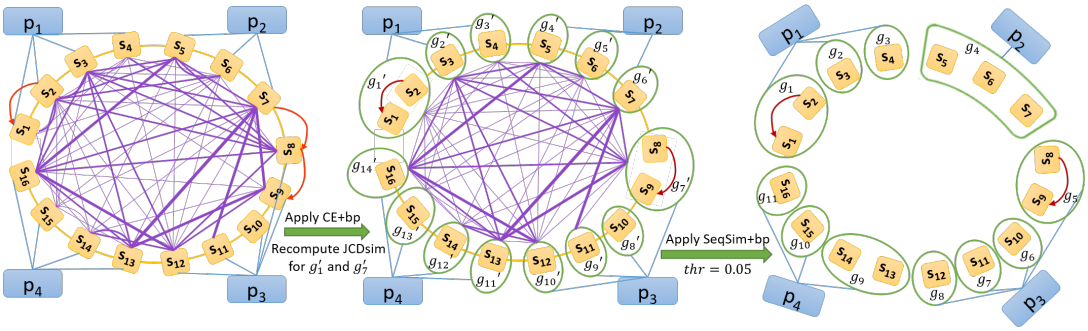


Figure B-8. The clustering process of the $SeqSim + pb(JCD, thr = 0.05)$ algorithm.



(a) $CESeqSim + npb(JCD, thr = 0.05)$



(b) $CESeqSim + pb(JCD, thr = 0.05)$

Figure B-9. The clustering process of the $CESeqSim(JCD, thr = 0.05)$ algorithm.

Appendix C. The Supplemental Results of the Four Metrics

C.1 The Preliminary Comparison

We first use the $SeqSim + pb(JCD, 0.06)$ algorithm to generate groups and use the JCD-type sim-SLN to rank sentences, groups or paragraphs. Figure C-1 shows the values of the four metrics on each sub-dataset when using the *Conclusion* or the *Introduction* of each paper as the standard summaries.

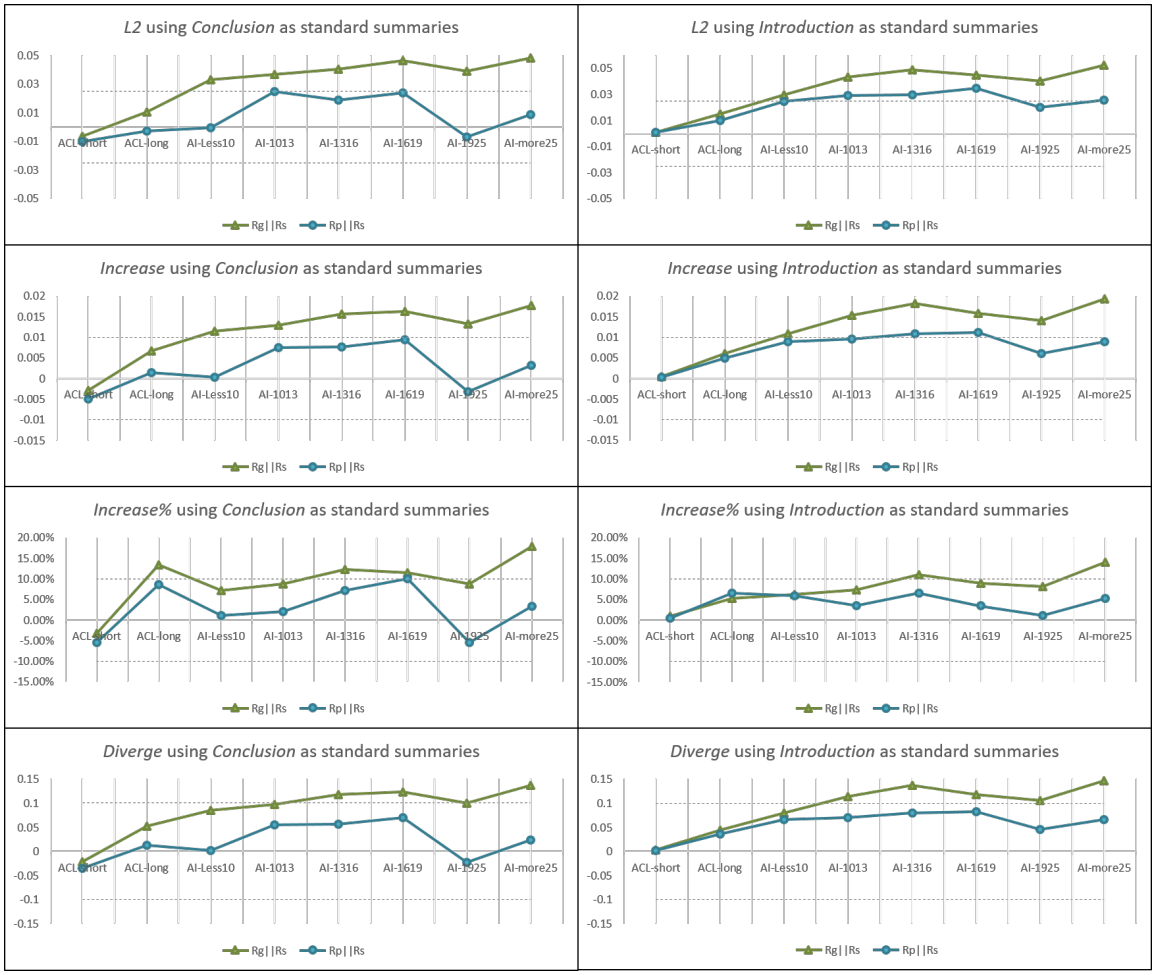


Figure C-1. The four metrics on each sub-dataset when using *SeqSim+pb(JCD, 0.06)* for generating groups and using JCD-type sim-SLNs for ranking.

C.2 The Further Comparison by Changing Types of sim-SLN

We still use the $SeqSim + pb(JCD, 0.06)$ algorithm to generate groups, but change the types of sim-SLN for ranking language units. Table C-1 to Table C-4 separately shows the results of the four metrics on the *ACL-all* dataset and the *AI-all* dataset when changing the types of sim-SLN and the kinds of standard summaries. Table C-5 to Table C-8 separately shows the results of the four metrics on each sub-dataset when changing the types of sim-SLN and the kinds of standard summaries.

Table C-1. The *L2* metric on the *ACL-all* and *AI-all* datasets when using different sim-SLNs and different standard summary.

L2 metric	The *ACL-all* dataset	The *AI-all* dataset

		Abstract	*Conclusion*	*Introduction*	*Abstract*	*Conclusion*	*Introduction*
JCD-type sim-SLNs	$R_g \| \| R_s$	0.0057	0.0024	0.0076	0.0348	0.0409	0.0433
	$R_p \| \| R_s$	-0.0084	-0.0063	0.0052	0.0061	0.0115	0.0275
AVG-type sim-SLNs	$R_g \| \| R_s$	0.0083	0.0035	-0.0085	0.0018	0.0025	-0.0205
	$R_p \| \| R_s$	-0.0175	-0.0115	-0.0132	0.0112	0.0100	-0.0019
SIF-type sim-SLNs	$R_g \| \| R_s$	0.0524	0.0499	0.0415	-0.0033	-0.0047	-0.0263
	$R_p \| \| R_s$	0.0607	0.0685	0.0589	-0.0156	-0.0079	-0.0250
GSE-type sim-SLNs	$R_g \| \| R_s$	-0.0042	0.0018	-0.0105	0.0155	0.0247	0.0348
	$R_p \| \| R_s$	-0.0124	0.0004	-0.0126	0.0138	0.0188	0.0310
LIN-type sim-SLNs	$R_g \| \| R_s$	-0.0199	-0.0045	0.0014	0.0201	0.0178	0.0109
	$R_p \| \| R_s$	-0.0102	0.0041	-0.0009	0.0211	0.0199	0.0191
WUP-type sim-SLNs	$R_g \| \| R_s$	0.0056	0.0208	0.0125	0.0272	0.0275	0.0282
	$R_p \| \| R_s$	0.0132	0.0318	0.0189	0.0375	0.0405	0.0427

Table C-2. The *Increase* metric on the *ACL-all* and *AI-all* datasets when using different sim-SLNs and different standard summary.

Increase metric		The *ACL-all* dataset			The *AI-all* dataset		
		Abstract	*Conclusion*	*Introduction*	*Abstract*	*Conclusion*	*Introduction*
JCD-type sim-SLNs	$R_g \| \| R_s$	0.0031	0.0021	0.0031	0.0130	0.0147	0.0156
	$R_p \| \| R_s$	-0.0007	-0.0015	0.0025	0.0033	0.0042	0.0093
AVG-type sim-SLNs	$R_g \| \| R_s$	0.0030	0.0031	-0.0025	0.0006	0.0008	-0.0082
	$R_p \| \| R_s$	-0.0073	-0.0027	-0.0047	0.0033	0.0025	-0.0022
SIF-type sim-SLNs	$R_g \| \| R_s$	0.0207	0.0195	0.0155	-0.0014	-0.0012	-0.0081
	$R_p \| \| R_s$	0.0236	0.0246	0.0223	-0.0080	-0.0046	-0.0102
GSE-type sim-SLNs	$R_g \| \| R_s$	-0.0001	0.0032	-0.0033	0.0053	0.0086	0.0125
	$R_p \| \| R_s$	-0.0033	0.0028	-0.0038	0.0055	0.0062	0.0111
LIN-type sim-SLNs	$R_g \| \| R_s$	-0.0072	0.0002	0.0012	0.0061	0.0055	0.0028
	$R_p \| \| R_s$	-0.0003	0.0052	0.0012	0.0063	0.0059	0.0055
WUP-type sim-SLNs	$R_g \| \| R_s$	0.0040	0.0096	0.0066	0.0086	0.0083	0.0092
	$R_p \| \| R_s$	0.0073	0.0147	0.0099	0.0119	0.0123	0.0142

Table C-3. The *Increase%* metric on the *ACL-all* and *AI-all* datasets when using different sim-SLNs and different standard summary.

Increase% metric		The *ACL-all* dataset			The *AI-all* dataset		
		Abstract	*Conclusion*	*Introduction*	*Abstract*	*Conclusion*	*Introduction*
JCD-type sim-SLNs	$R_g \| \| R_s$	0.0475	0.0557	0.0312	0.1671	0.1124	0.0921
	$R_p \| \| R_s$	0.0546	0.0204	0.0359	0.0924	0.0335	0.0441
AVG-type sim-SLNs	$R_g \| \| R_s$	0.0236	0.0871	0.0066	0.0325	0.0124	-0.0560
	$R_p \| \| R_s$	-0.0969	0.0283	-0.0182	0.0368	-0.0015	-0.0383
SIF-type sim-SLNs	$R_g \| \| R_s$	0.4885	0.3132	0.1313	-0.0169	0.0009	-0.0126
	$R_p \| \| R_s$	0.5103	0.3184	0.1950	-0.1615	-0.0745	-0.0672
GSE-type sim-SLNs	$R_g \| \| R_s$	0.0557	0.1309	-0.0090	0.0631	0.0548	0.0725
	$R_p \| \| R_s$	0.0071	0.1272	-0.0075	0.1432	0.0374	0.0651
LIN-type sim-SLNs	$R_g \| \| R_s$	-0.0553	0.0566	0.0286	0.0759	0.0319	-0.0137
	$R_p \| \| R_s$	0.1073	0.1575	0.0456	0.0332	0.0193	0.0018
WUP-type sim-SLNs	$R_g \| \| R_s$	0.1242	0.1339	0.0922	0.0878	0.0345	0.0408

	$R_p \| \| R_s$	0.1686	0.2017	0.1326	0.1229	0.0491	0.0744

Table C-4. The *Diverge* metric on the *ACL-all* and *AI-all* datasets when using different sim-SLNs and different standard summary.

Diverge metric		The *ACL-all* dataset			The *AI-all* dataset		
		Abstract	*Conclusion*	*Introduction*	*Abstract*	*Conclusion*	*Introduction*
JCD-type sim-SLNs	$R_g \| \| R_s$	0.0232	0.0160	0.0230	0.0989	0.1109	0.1170
	$R_p \| \| R_s$	-0.0040	-0.0106	0.0180	0.0247	0.0306	0.0685
AVG-type sim-SLNs	$R_g \| \| R_s$	0.0222	0.0235	-0.0179	0.0050	0.0060	-0.0582
	$R_p \| \| R_s$	-0.0502	-0.0191	-0.0335	0.0245	0.0187	-0.0155
SIF-type sim-SLNs	$R_g \| \| R_s$	0.1671	0.1536	0.1170	-0.0097	-0.0086	-0.0570
	$R_p \| \| R_s$	0.1915	0.1947	0.1706	-0.0538	-0.0317	-0.0714
GSE-type sim-SLNs	$R_g \| \| R_s$	-0.0001	0.0256	-0.0233	0.0394	0.0633	0.0932
	$R_p \| \| R_s$	-0.0236	0.0220	-0.0271	0.0413	0.0453	0.0820
LIN-type sim-SLNs	$R_g \| \| R_s$	-0.0506	0.0025	0.0092	0.0454	0.0408	0.0202
	$R_p \| \| R_s$	0.0004	0.0415	0.0089	0.0470	0.0437	0.0402
WUP-type sim-SLNs	$R_g \| \| R_s$	0.0309	0.0726	0.0496	0.0648	0.0622	0.0683
	$R_p \| \| R_s$	0.0563	0.1134	0.0747	0.0912	0.0932	0.1071

Table C-5. The *L2* metric on each sub-dataset when using different sim-SLNs and different standard summary.

L2 metric		JCD-type sim-SLNs		AVG-type sim-SLNs		SIF-type sim-SLNs		GSE-type sim-SLNs		LIN-type sim-SLNs		WUP-type sim-SLNs	
		$R_g \| \| R_s$	$R_p \| \| R_s$	$R_g \| \| R_s$	$R_p \| \| R_s$	$R_g \| \| R_s$	$R_p \| \| R_s$	$R_g \| \| R_s$	$R_p \| \| R_s$	$R_g \| \| R_s$	$R_p \| \| R_s$	$R_g \| \| R_s$	$R_p \| \| R_s$
ACL-short	*Abs.*	0.007	0.005	0.020	-0.004	0.023	0.036	0.003	0.002	-0.029	-0.010	0.002	0.021
	Conc.	-0.006	-0.010	0.005	-0.007	0.018	0.048	0.004	-0.001	-0.031	-0.007	-0.005	0.035
	Intro.	0.001	0.001	-0.001	-0.006	0.034	0.043	-0.001	-0.006	-0.003	-0.004	0.010	0.012
ACL-long	*Abs.*	0.003	-0.021	-0.004	-0.031	0.081	0.084	-0.011	-0.027	-0.010	-0.011	0.008	0.004
	Conc.	0.010	-0.003	0.003	-0.015	0.079	0.087	0.000	0.003	0.020	0.014	0.045	0.028
	Intro.	0.015	0.010	-0.015	-0.020	0.049	0.074	-0.018	-0.018	0.006	0.002	0.015	0.026
AI-less10	*Abs.*	0.030	0.001	-0.011	0.009	0.016	0.000	0.006	0.008	0.019	0.023	0.013	0.019
	Conc.	0.033	-0.001	0.011	0.017	0.019	0.012	0.012	0.004	0.018	0.021	0.037	0.053
	Intro.	0.030	0.025	-0.009	-0.001	0.007	0.012	0.027	0.018	0.008	0.014	0.021	0.039
AI-1013	*Abs.*	0.038	0.028	0.005	-0.001	-0.003	-0.025	0.004	-0.007	0.014	0.010	0.023	0.047
	Conc.	0.036	0.025	0.011	0.020	0.017	-0.005	0.008	-0.001	0.026	0.028	0.011	0.041
	Intro.	0.043	0.029	-0.008	0.012	-0.009	-0.021	0.029	0.024	0.008	0.011	0.019	0.044
AI-1316	*Abs.*	0.033	0.008	0.005	0.015	0.003	-0.014	0.018	0.019	0.028	0.016	0.033	0.044
	Conc.	0.040	0.019	0.004	0.011	0.012	0.004	0.022	0.011	0.016	0.010	0.030	0.033
	Intro.	0.049	0.030	-0.016	0.005	-0.020	-0.014	0.022	0.018	0.014	0.016	0.036	0.044
AI-1619	*Abs.*	0.039	0.018	0.005	0.014	0.027	0.010	0.014	0.010	0.009	0.024	0.024	0.035
	Conc.	0.046	0.024	0.016	0.010	0.010	0.010	0.023	0.024	0.016	0.025	0.023	0.043
	Intro.	0.045	0.035	-0.011	0.011	-0.002	-0.002	0.032	0.030	0.006	0.032	0.019	0.043
AI-1925	*Abs.*	0.021	-0.033	0.008	0.018	-0.024	-0.008	0.013	0.021	0.025	0.027	0.030	0.040
	Conc.	0.039	-0.007	-0.003	0.011	-0.018	0.008	0.033	0.032	0.025	0.021	0.029	0.037
	Intro.	0.040	0.020	-0.027	-0.008	-0.047	-0.013	0.035	0.036	0.015	0.022	0.035	0.050
AI-more25	*Abs.*	0.050	0.016	-0.004	0.008	-0.040	-0.062	0.039	0.031	0.024	0.028	0.037	0.038

	Conc.	0.048	0.009	-0.025	-0.008	-0.066	-0.081	0.051	0.041	0.006	0.016	0.036	0.041
	Intro.	0.052	0.025	-0.052	-0.034	-0.085	-0.117	0.063	0.060	0.014	0.020	0.039	0.037

Table C-6. The *Increase* metric on each sub-dataset when using different sim-SLNs and different standard summary.

Increase metric		JCD-type sim-SLNs		AVG-type sim-SLNs		SIF-type sim-SLNs		GSE-type sim-SLNs		LIN-type sim-SLNs		WUP-type sim-SLNs	
		$R_g\|\|R_s$	$R_p\|\|R_s$	$R_g\|\|R_s$	$R_p\|\|R_s$	$R_g\|\|R_s$	$R_p\|\|R_s$	$R_g\|\|R_s$	$R_p\|\|R_s$	$R_g\|\|R_s$	$R_p\|\|R_s$	$R_g\|\|R_s$	$R_p\|\|R_s$
ACL-short	*Abs.*	0.002	0.003	0.008	-0.002	0.009	0.014	0.003	0.005	-0.012	-0.003	0.004	0.011
	Conc.	-0.003	-0.005	0.003	-0.003	0.007	0.016	0.004	0.001	-0.013	-0.001	-0.003	0.016
	Intro.	0.001	0.001	0.001	-0.001	0.013	0.016	0.002	0.001	-0.001	-0.001	0.004	0.005
ACL-long	*Abs.*	0.004	-0.004	-0.002	-0.013	0.032	0.033	-0.003	-0.011	-0.002	0.002	0.004	0.003
	Conc.	0.007	0.001	0.004	-0.002	0.031	0.032	0.003	0.005	0.012	0.011	0.021	0.013
	Intro.	0.006	0.005	-0.006	-0.008	0.018	0.028	-0.007	-0.007	0.004	0.003	0.009	0.015
AI-less10	*Abs.*	0.013	0.003	-0.007	0.006	0.007	-0.001	0.002	0.006	0.006	0.007	0.004	0.006
	Conc.	0.011	0.000	0.007	0.008	0.009	0.003	0.005	0.002	0.005	0.008	0.013	0.018
	Intro.	0.011	0.009	-0.005	-0.003	0.005	0.005	0.010	0.008	0.001	0.003	0.008	0.015
AI-1013	*Abs.*	0.017	0.013	0.004	-0.003	0.002	-0.011	0.001	-0.002	0.004	0.004	0.007	0.014
	Conc.	0.013	0.007	0.004	0.005	0.008	-0.003	0.001	-0.002	0.009	0.011	0.003	0.012
	Intro.	0.015	0.010	-0.002	0.005	0.000	-0.009	0.010	0.007	0.001	0.003	0.006	0.015
AI-1316	*Abs.*	0.012	0.002	0.003	0.003	-0.002	-0.010	0.006	0.007	0.009	0.004	0.012	0.015
	Conc.	0.016	0.008	0.000	0.002	0.005	-0.001	0.008	0.004	0.005	0.000	0.009	0.010
	Intro.	0.018	0.011	-0.007	0.000	-0.006	-0.007	0.008	0.006	0.004	0.005	0.012	0.014
AI-1619	*Abs.*	0.013	0.007	0.002	0.004	0.010	0.001	0.005	0.003	0.004	0.008	0.008	0.011
	Conc.	0.016	0.009	0.006	0.001	0.004	0.002	0.008	0.008	0.005	0.007	0.006	0.013
	Intro.	0.016	0.011	-0.005	0.002	0.001	-0.002	0.012	0.011	0.002	0.010	0.005	0.014
AI-1925	*Abs.*	0.007	-0.011	0.002	0.007	-0.011	-0.004	0.006	0.008	0.007	0.008	0.009	0.013
	Conc.	0.013	-0.003	-0.002	0.002	-0.008	0.000	0.012	0.011	0.008	0.006	0.009	0.011
	Intro.	0.014	0.006	-0.011	-0.004	-0.017	-0.006	0.013	0.012	0.004	0.006	0.012	0.016
AI-more25	*Abs.*	0.017	0.006	-0.001	0.002	-0.015	-0.024	0.013	0.011	0.007	0.008	0.011	0.011
	Conc.	0.018	0.003	-0.008	-0.002	-0.024	-0.030	0.017	0.014	0.001	0.004	0.011	0.012
	Intro.	0.019	0.009	-0.020	-0.013	-0.031	-0.044	0.023	0.022	0.004	0.005	0.012	0.012

Table C-7. The *Increase%* metric on each sub-dataset when using different sim-SLNs and different standard summary.

Increase% metric		JCD-type sim-SLNs		AVG-type sim-SLNs		SIF-type sim-SLNs		GSE-type sim-SLNs		LIN-type sim-SLNs		WUP-type sim-SLNs	
		$R_g\|\|R_s$	$R_p\|\|R_s$	$R_g\|\|R_s$	$R_p\|\|R_s$	$R_g\|\|R_s$	$R_p\|\|R_s$	$R_g\|\|R_s$	$R_p\|\|R_s$	$R_g\|\|R_s$	$R_p\|\|R_s$	$R_g\|\|R_s$	$R_p\|\|R_s$
ACL-short	*Abs.*	0.000	0.063	0.102	-0.035	0.196	0.265	0.091	0.164	-0.144	0.003	0.179	0.257
	Conc.	-0.032	-0.054	0.017	-0.087	0.068	0.146	0.106	0.066	-0.155	0.028	-0.072	0.246
	Intro.	0.009	0.004	0.053	0.025	0.111	0.141	0.067	0.046	-0.010	0.004	0.046	0.050
ACL-long	*Abs.*	0.090	0.048	-0.031	-0.139	0.792	0.761	0.024	-0.120	0.040	0.205	0.079	0.084
	Conc.	0.134	0.086	0.146	0.118	0.492	0.444	0.157	0.182	0.238	0.265	0.270	0.170
	Intro.	0.053	0.064	-0.026	-0.048	0.150	0.241	-0.061	-0.045	0.061	0.078	0.133	0.203
AI-less10	*Abs.*	0.140	0.115	-0.140	0.185	0.084	-0.042	0.013	0.149	0.109	0.078	0.015	0.101
	Conc.	0.072	0.011	0.174	0.158	0.104	-0.004	0.040	0.004	0.055	0.201	0.159	0.204
	Intro.	0.063	0.059	-0.053	-0.071	0.091	0.049	0.071	0.088	-0.052	-0.011	0.086	0.156

AI-1013	*Abs.*	0.248	0.203	0.140	-0.125	0.171	-0.163	0.017	0.064	-0.021	0.022	-0.021	0.055
	Conc.	0.088	0.020	0.027	-0.025	0.085	-0.041	-0.039	-0.034	0.113	0.191	-0.021	0.010
	Intro.	0.073	0.034	0.024	0.044	0.066	-0.050	0.031	0.008	-0.026	0.004	0.018	0.082
AI-1316	*Abs.*	0.132	0.007	0.140	-0.055	-0.098	-0.217	0.037	0.161	0.116	-0.122	0.369	0.257
	Conc.	0.123	0.072	-0.028	-0.019	0.040	-0.037	0.047	0.051	0.005	-0.103	0.045	0.029
	Intro.	0.110	0.064	-0.063	-0.043	-0.005	-0.070	0.032	0.028	0.009	-0.005	0.050	0.057
AI-1619	*Abs.*	0.106	0.084	0.029	-0.006	0.214	-0.129	0.062	0.066	0.222	0.185	0.058	0.022
	Conc.	0.115	0.100	0.057	-0.050	0.058	-0.071	0.058	0.042	0.037	0.024	-0.010	0.036
	Intro.	0.089	0.034	-0.057	-0.028	0.039	-0.034	0.084	0.072	-0.006	0.048	-0.008	0.044
AI-1925	*Abs.*	0.075	-0.131	0.091	0.262	-0.264	-0.089	0.270	0.389	0.031	-0.045	0.096	0.437
	Conc.	0.088	-0.055	-0.053	-0.023	-0.094	-0.087	0.123	0.074	0.076	0.019	0.024	0.061
	Intro.	0.081	0.011	-0.084	-0.055	-0.084	-0.062	0.087	0.068	-0.001	0.002	0.083	0.089
AI-more25	*Abs.*	0.380	0.246	0.006	0.040	-0.178	-0.324	0.312	0.272	-0.133	0.000	0.294	0.180
	Conc.	0.179	0.034	-0.044	0.002	-0.134	-0.204	0.106	-0.368	-0.031	-0.016	0.038	0.030
	Intro.	0.140	0.052	-0.108	-0.087	-0.139	-0.204	0.141	0.137	-0.008	-0.020	0.044	0.048

Table C-8. The *Diverge* metric on each sub-dataset when using different sim-SLNs and different standard summary.

Diverge metric		JCD-type sim-SLNs		AVG-type sim-SLNs		SIF-type sim-SLNs		GSE-type sim-SLNs		LIN-type sim-SLNs		WUP-type sim-SLNs	
		$R_g\|\|R_s$	$R_p\|\|R_s$	$R_g\|\|R_s$	$R_p\|\|R_s$	$R_g\|\|R_s$	$R_p\|\|R_s$	$R_g\|\|R_s$	$R_p\|\|R_s$	$R_g\|\|R_s$	$R_p\|\|R_s$	$R_g\|\|R_s$	$R_p\|\|R_s$
ACL-short	*Abs.*	0.014	0.023	0.060	-0.013	0.066	0.106	0.024	0.038	-0.084	-0.022	0.034	0.089
	Conc.	-0.021	-0.035	0.020	-0.022	0.049	0.125	0.027	0.011	-0.089	-0.010	-0.019	0.128
	Intro.	0.004	0.002	0.007	-0.008	0.094	0.120	0.012	-0.003	-0.009	-0.006	0.032	0.037
ACL-long	*Abs.*	0.030	-0.028	-0.015	-0.086	0.278	0.281	-0.024	-0.077	-0.012	0.023	0.028	0.022
	Conc.	0.052	0.012	0.029	-0.014	0.254	0.260	0.028	0.037	0.096	0.090	0.164	0.100
	Intro.	0.044	0.036	-0.039	-0.056	0.139	0.220	-0.053	-0.047	0.028	0.025	0.069	0.114
AI-less10	*Abs.*	0.097	0.028	-0.045	0.047	0.051	-0.006	0.016	0.043	0.043	0.051	0.031	0.048
	Conc.	0.085	0.002	0.051	0.059	0.064	0.024	0.035	0.012	0.040	0.061	0.098	0.138
	Intro.	0.080	0.066	-0.032	-0.020	0.037	0.036	0.076	0.059	0.006	0.024	0.062	0.115
AI-1013	*Abs.*	0.135	0.096	0.031	-0.020	0.020	-0.076	0.007	-0.012	0.029	0.026	0.050	0.112
	Conc.	0.097	0.055	0.031	0.037	0.056	-0.019	0.010	-0.012	0.068	0.084	0.022	0.093
	Intro.	0.114	0.070	-0.012	0.036	0.001	-0.061	0.071	0.053	0.010	0.025	0.044	0.115
AI-1316	*Abs.*	0.093	0.018	0.022	0.022	-0.011	-0.064	0.042	0.054	0.067	0.028	0.091	0.118
	Conc.	0.117	0.057	-0.001	0.016	0.034	-0.004	0.057	0.032	0.035	0.005	0.071	0.075
	Intro.	0.136	0.080	-0.052	-0.002	-0.039	-0.049	0.056	0.046	0.032	0.033	0.087	0.106
AI-1619	*Abs.*	0.097	0.049	0.012	0.027	0.079	0.006	0.035	0.023	0.028	0.062	0.058	0.081
	Conc.	0.123	0.070	0.043	0.009	0.032	0.012	0.058	0.055	0.036	0.054	0.047	0.096
	Intro.	0.119	0.083	-0.038	0.017	0.009	-0.012	0.088	0.078	0.012	0.077	0.038	0.102
AI-1925	*Abs.*	0.136	0.048	-0.008	0.016	-0.099	-0.155	0.104	0.084	0.051	0.063	0.090	0.090
	Conc.	0.137	0.024	-0.059	-0.015	-0.160	-0.200	0.133	0.102	0.008	0.032	0.082	0.091
	Intro.	0.147	0.066	-0.135	-0.094	-0.212	-0.287	0.177	0.169	0.027	0.036	0.092	0.089
AI-more25	*Abs.*	0.051	-0.076	0.018	0.053	-0.073	-0.025	0.043	0.063	0.053	0.057	0.069	0.099
	Conc.	0.100	-0.023	-0.017	0.017	-0.053	0.001	0.091	0.079	0.062	0.048	0.064	0.086
	Intro.	0.105	0.045	-0.077	-0.031	-0.117	-0.043	0.096	0.091	0.030	0.047	0.090	0.123

C.3 The Role of the Types of Semantic Links in Generating Groups

Figure C-2, Figure C-3, Figure C-4 and Figure C-5 show more results of the *Diverge* metric got by using different groups that are generated by different clustering algorithms. The results shown in these figures support the five strategies we proposed about using the *is-part-of*, *sequential*, *similar-to* and *cause-effect* links to generate groups.

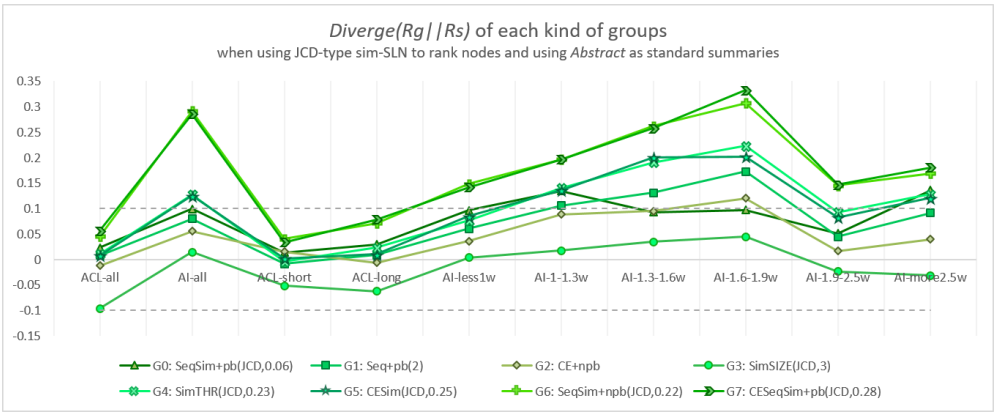
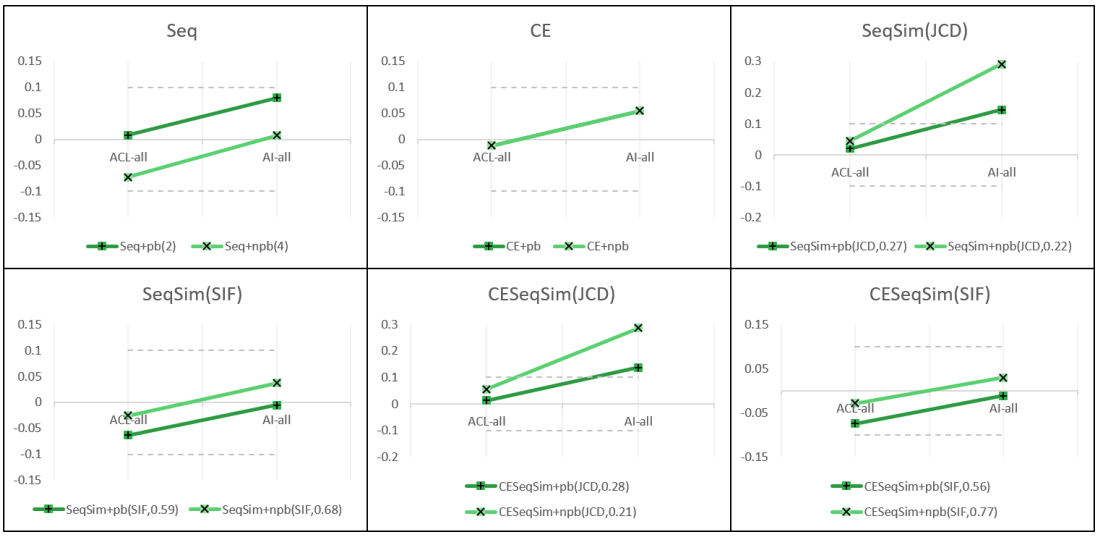
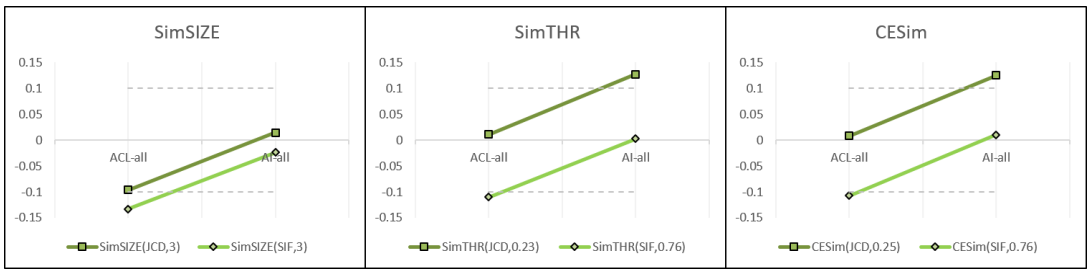


Figure C-2. The *Diverge* metric results of different kinds of group on all datasets when using JCD-type sim-SLN to rank nodes and using *Abstract* as standard summaries.

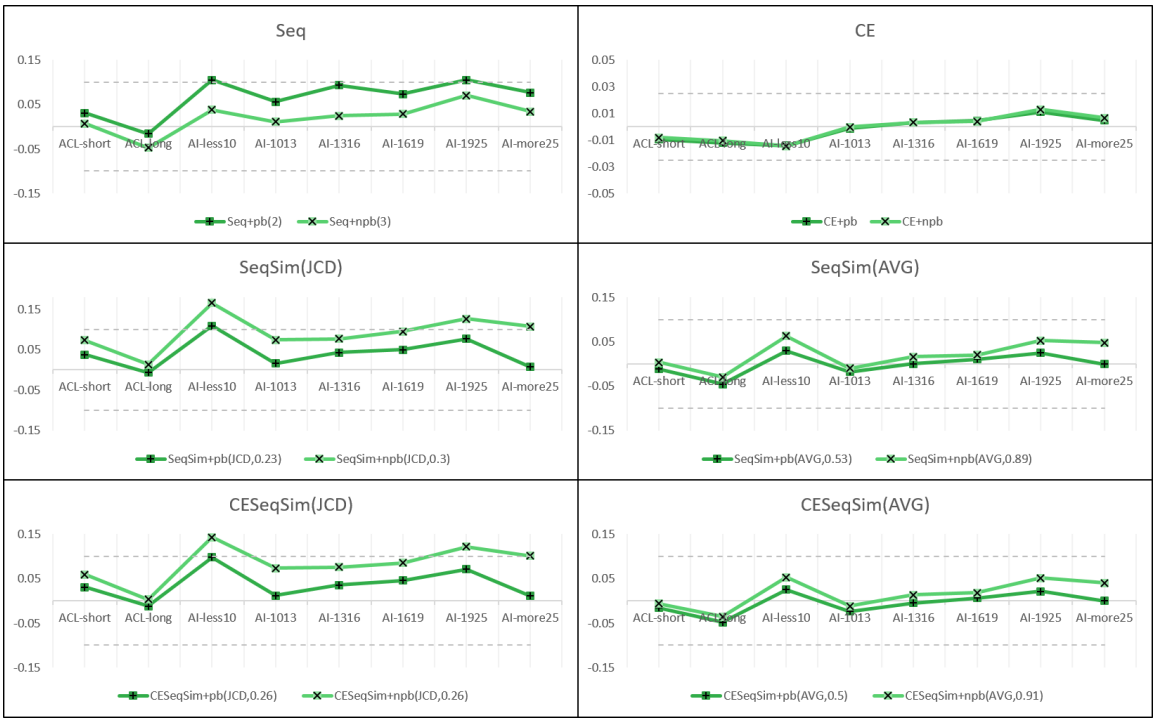


(a) Using '*+pb*' or '*+npb*' mode in each clustering algorithm.



(b) Using JCD-type or SIF-type *similar-to* link in each clustering algorithm.

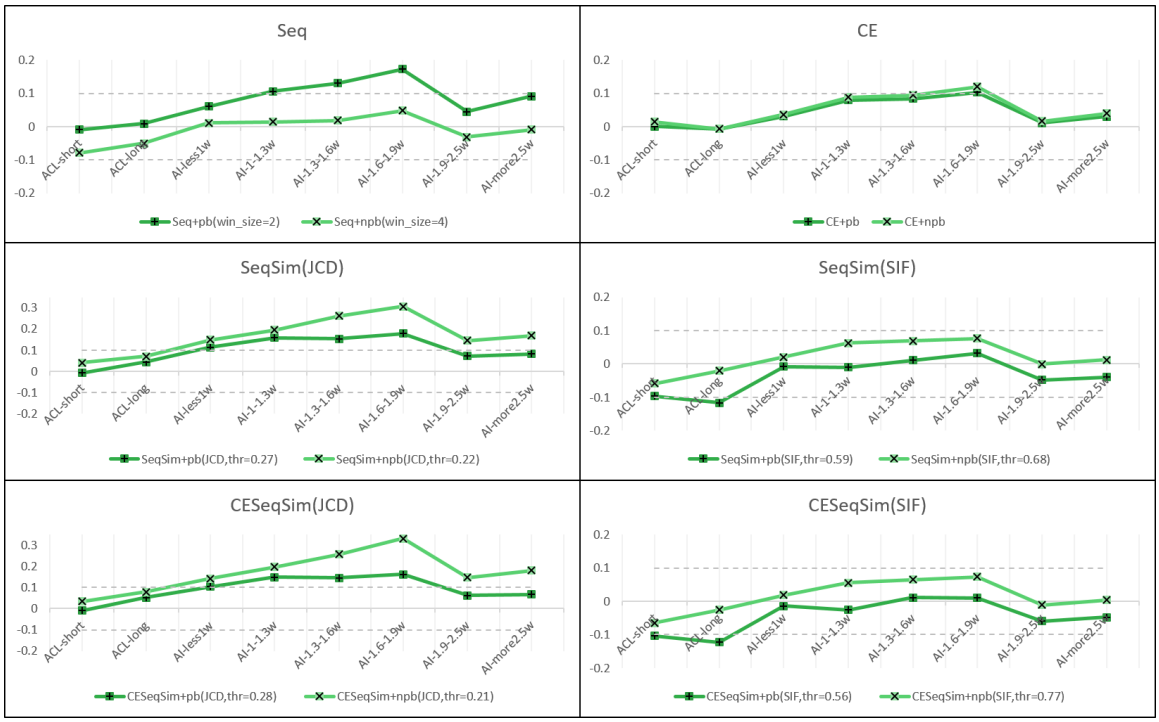
Figure C-3. The *Diverge* metric of each kind of group on the *ACL-all* and *AI-all* datasets when using JCD-type sim-SLN to rank nodes and using *Abstract* as standard summaries.



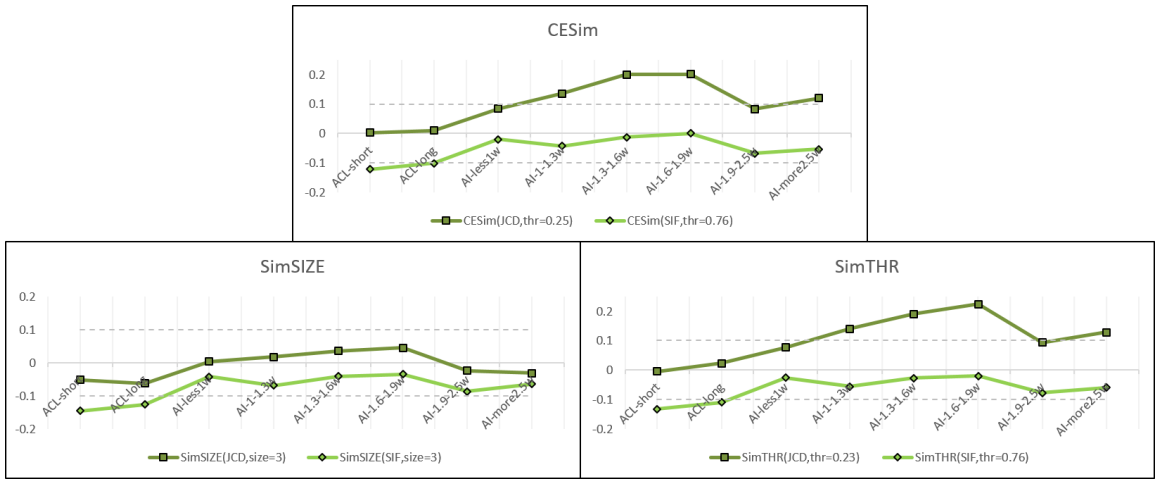
(a) Using '+*pb*' or '+*npb*' mode in each clustering algorithm.

(b) Using JCD-type or AVG-type *similar-to* link in each clustering algorithm.

Figure C-4. The *Diverge* metric results of each kind of group on all sub-datasets when using AVG-type sim-SLN to rank and using *Abstract* as standard summaries



(a) Using '+*pb*' or '+*npb*' mode in each clustering algorithm.



(b) Using JCD-type or SIF-type *similar-to* link in each clustering algorithm.

Figure C-5. The *Diverge* values of each kind of group on all sub-datasets when using JCD-type sim-SLN to rank nodes and using *Abstract* as standard summaries.

Appendix D. The Readability Evaluation

A summary generated by extracting sentences from the source text in isolation usually has low readability, because the context and the original semantic relations of the extracted sentences are lost. Since sentences in the source text are coherent, we propose the following assumption:

Assumption: *Extracting larger language units that retain the semantics between sentences in the source text to generate a summary naturally improves the readability of the generated summary*.

To verify the above assumption, we randomly select 18 long papers and 17 short papers from the *ACL-all* dataset, and then collect the *sentence*-based, *group*-based and *paragraph*-based summaries for each paper to form a dataset named as *Samples*. The *Samples* dataset is further divided into the *Samples-long* and *Samples-short* sub-datasets to study the readability of summaries of different lengths of papers.

We consider that the readability of a summary includes four aspects:

- *Sentence integrity*, referring each sentence in a summary should be a grammatically complete sentence.
- *Referential clarity*, referring the anaphors (including pronouns, noun phrases started with a determiner, unfamiliar named entities and unfamiliar abbreviations) within a summary can be resolved correctly.
- *Local coherence*, referring to any two adjacent sentences should be coherent. The *local coherence* can be assessed from three aspects:
  - *Thematic coherence*, meaning that two adjacent sentences should talk about the same theme. The theme of a sentence can be reflected by the noun/verb phrases within the sentence.
  - *Conciseness,* meaning that any two adjacent sentences should not express the same meaning. We view the *conciseness* as an aspect of *local coherence* instead of emphasizing it on the whole summary because it is reasonable to have some redundancy in a summary to highlight some themes.
  - *Semantic coherence,* meaning that the two adjacent sentences should correctly express a semantic relation when the corresponding cue word/phrase/sentence-pattern occurs. For example, if s_1 and s_2 are two adjacent sentences in a summary and s_2 starts with the word 'However', then there should be a transitional relation between s_1 and s_2.
- *Structure*, referring that the generated summaries should follow the structure that people follow when they write a summary. For example, the structure of summaries for scientific papers should be *the purpose → the backgroud → the methods → the results or conclusions.*

To date, there is no appropriate metric for evaluating the readability of summaries according to the aspects listed above. Researchers in the field of education and linguistics first put forward some metrics to measure the readability of texts, such as SMOG index[56], automated readability index, Gunning fog index and Flesch reading easy formula [57]. However, these metrics are only applicable in evaluating the manually written articles, because they only depend on the percentage of complex words and the average amount of words in each sentence without considering any aspect of readability listed above. The models introduced in [58, 59] aim to evaluate the *local coherence*. But they just estimate the probability of dependence between any two content words (including verbs, nouns, named entity tags and connectives) to form the score of coherence, without considering the semantic coherence between adjacent sentences, that is, whether adjacent sentences convey a semantic relation such as cause-effect, question-answer, transition, progressive relation, and so on. The model in [60] aims to automatically evaluate the readability of summaries from aspects of grammaticality, coherence and focus. However, the grammaticality evaluated in this work just refers to the frequency of POS-tag or chunk-tag tri-gram sequence, and the coherence between two adjacent sentences is still evaluated by estimating the relatedness among noun phrases within these two sentences.

We invite two volunteers A and B to manually evaluate the readability of summaries. The *referential clarity* is evaluated according to the number of dangling anaphors within a summary. Dangling anaphors are those anaphors whose referents cannot be found in the text, and a summary has higher *referential clarity* if it contains less dangling anaphors. The *sentence integrity*, the *local coherence* and the *structure* of a summary are evaluated together according to the readability rating of a summary, which is an integer from 0 to 5 that is obtained by sum up the scores in Table D-1.

Figure D-2 shows the dangling anaphors and the readability ratings annotated for the generated summaries of a scientific paper. The phrase "the baseline" in the *sentence*-based summary is labelled as a dangling anaphor since its referent cannot be found in this summary. No dangling anaphor is found in both *group*-based and *paragraph*-based summary. The readability ratings of the *group*-based summary given by A and B are both 5, which are higher than the readability ratings of *sentence*-based and *paragraph*-based summary.

All the summaries in the *Samples* dataset and the manual annotations for evaluating the readability of these summaries can be downloaded from the GitHub[1].

Table D-1. The aspects and the corresponding score values for calculating the readability rating of a summary.

Aspect		Condition	Score
Sentence Integrity		All sentences are complete sentences. (Wrong segmentation of sentence and formulas cause incomplete sentences.)	1
Local Coherence	*Thematic coherence*	More than 66% pairs of adjacent sentences sharing at least one noun/verb phrase (The parameter 66% is selected according to the observation that more than 96% summaries in the *Sample* dataset have at least 3 pairs of adjacent sentences.)	1
	Conciseness	Any two adjacent sentences should not express the same meaning.	1
	Semantic coherence	Any two adjacent sentences express the correct semantic relations if they contain the corresponding cue word/phrase/sentence-pattern. (We require that the question sentence in the summary must be followed by an answer sentence.)	1
Structure		The sentences within a summary expresses two or more of *the purpose*, *the background, the methods*, *the results or conclusions*.	1

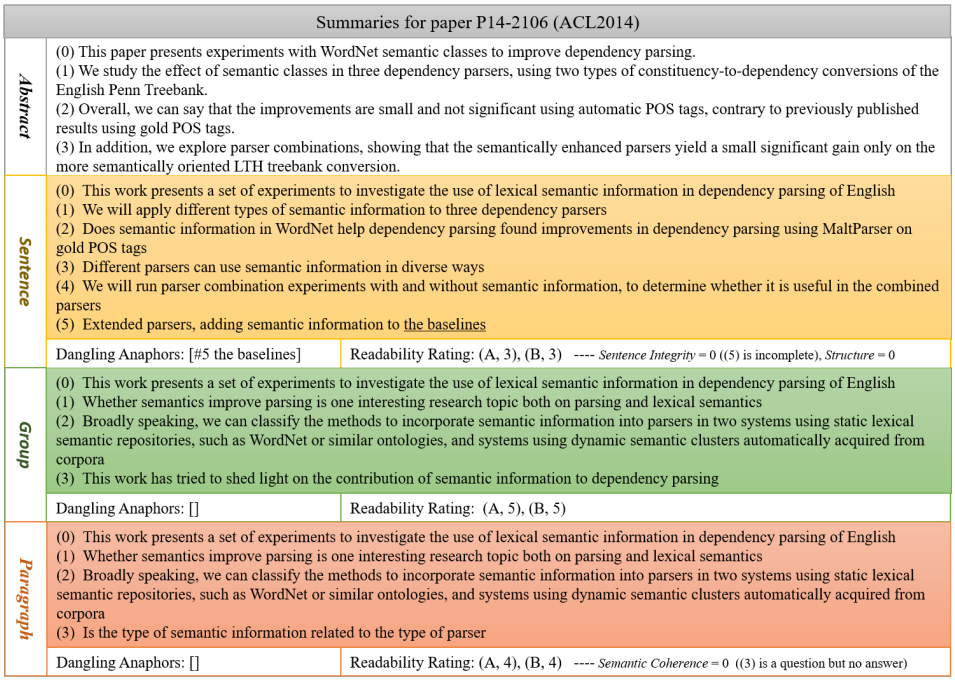


Figure D-2. Dangling anaphors and readability ratings in the *sentence*-based, *group*-based and *paragraph*-based summaries of a paper.

Table D-2 lists the weighted kappa coefficients between the two volunteers for the readability ratings, indicating that the readability ratings given by the two annotators are strongly consistent. Table D-3 shows the average amount and the standard deviation of dangling anaphors in the summaries, where the minimum on each dataset is highlighted in bold. Table D-4 shows the average readability ratings and the standard deviation of the summaries, where the maximum on each dataset is highlighted in bold.

From Table D-3 and Table D-4, we can see that both the *group*-based summary and the *paragraph*-based summary contain less dangling anaphors and get higher readability ratings than the *sentence*-based summary, and the *group*-based summary contains a similar amount of dangling anaphors and gets similar readability ratings as the *paragraph*-based summary. This suggests that summaries generated by extracting larger language units have better readability than summaries generated by extracting sentences.

[1] https://github.com/Angela7126/Group_For_Summarization

Comparing the dangling anaphors statistics on the *Samples-long* and *Samples-short* sub-datasets in Table D-3, we find than the number of dangling anaphors contained in the *group-based* or *paragraph*-based summary decreases as the source text becomes longer, while the amount of dangling anaphors contained in the *sentence*-based summary remains the same.

Comparing the average of readability ratings on the *Samples-long* and *Samples-short* sub-datasets in Table D-4, we find that the average ratings of the *sentence*-based, *group*-based and *paragraph*-based summary all decline as the source texts become longer. However, the average readability ratings of the *group*-based and *paragraph*-based summary decrease less than the average readability rating of *sentence*-based summary, showing the performance of larger language units in maintaining the readability of the generated summaries when the source text becomes longer.

Based on the above analysis, we can say that the readability of summaries composed by larger language units is better than the readability of the *sentence*-based summary especially when the source texts become longer, verifying the assumption we proposed in the beginning.

Table D-2. The weighted kappa coefficients between the readability ratings given by the two volunteers.

	sentence-based	*group*-based	*paragraph*-based
Weighted Kappa	0.792	0.749	0.747

Table D-3. The average amount of dangling anaphors in the summaries composed by *sentence*, *group* or *paragraph*.

Dataset	*sentence*-based	*group*-based	*paragraph*-based
Samples	4.4 ± 2.23	3.06 ± 1.88	$\mathbf{3.03} \pm 2.54$
Samples-long	4.44 ± 2.14	2.94 ± 1.68	$\mathbf{2.67} \pm 1.63$
Samples-short	4.35 ± 2.32	$\mathbf{3.18} \pm 2.06$	3.41 ± 3.18

Table D-4. The average readability ratings of the summaries composed by *sentence*, *group* or *paragraph*.

Dataset	*sentence*-based	*group*-based	*paragraph*-based
Samples	2.0 ± 1.23	$\mathbf{3.29} \pm 1.24$	3.23 ± 1.26
Samples-long	1.78 ± 1.27	3.08 ± 1.23	3.08 ± 1.23
Samples-short	2.24 ± 1.14	$\mathbf{3.5} \pm 1.22$	3.38 ± 1.26

Supplementary References

[56] Mc Laughlin, G. Harry, SMOG grading-a new readability formula, Journal of Reading, 12 (8) (1969) 639-646.

[57] J.P. Kincaid, R.P. Fishburne, R.L. Rogers, B.S. Chissom, Derivation of new readability formulas (automated readability index, fog count, and flesch reading ease formula) for navy enlisted personnel, Naval Air Station Memphis Research Branch Report, (1975).

[58] R. Barzilay, M. Lapata, Modeling local coherence: An entity-based approach, Computational Linguistics, 34 (1) (2008) 1-34.

[59] H. Nishikawa, T. Hasegawa, Y. Matsuo, G. Kikui, Optimizing informativeness and readability for sentiment summarization, in: Proceedings of the ACL 2010 conference short papers, 2010, pp. 325-330.

[60] R. Vadlapudi, R. Katragadda, On automated evaluation of readability of summaries: Capturing grammaticality, focus, structure and coherence, in: Proceedings of the NAACL HLT 2010 student research workshop, 2010, pp. 7-12.

Appendix E. The Extensions of Our Previous Work

This paper extends our previous work[1] on the experimental datasets, the semantic links for constructing SLNs, the clustering algorithms for generating groups, and the conclusions of the research. Table E-1 lists all the extensions.

Table E-1. The extensions of our previous work.

	Previous Work	Extensions in this paper
Experimental dataset	The *ACL-all* dataset: it contains 173 conference papers (88 long papers and 85 short papers) collected from the proceedings of ACL 2014. The *ACL-short* and *ACL-long* datasets: these are the subsets of the *ACL-all* dataset to discuss the performance of *group* in extractive summarization on different length of source text. The *Sample* dataset: it contains 105 summaries of 35 papers for comparing the readability of the summaries composed by different language units.	The *AI-all* dataset: it contains 372 journal papers collected from the *Artificial Intelligence*. The length of papers ranges from 2542 words to 52833 words. The *AI-less10*, *AI-1013*, *AI-1316*, *AI-1619*, *AI-1925* and *AI-more25*: these are the subsets of the *AI-all* dataset to discuss the performance of *group* in extractive summarization on different length of source text.
Semantic links for constructing SLN	The *sequential* link: the appearance order of sentences in the source text. The *is-part-of* link: the subordination between each sentence to its paragraph. The *similar-to* link: Jaccard distance (JCD) is used to calculate the similarity between two language units.	The *similar-to* link: Jaccard distance (JCD), average word embedding (AVG), smooth inverse frequency (SIF), Google sentence encoder (GSE), wup-similarity on WordNet (WUP), and lin-similarity on WordNet (LIN) are used to calculate the similarity between two language units. The *cause-effect* link: a pattern-based algorithm[2] is used to extract the explicit cause-effect links between sentences.
Clustering algorithms for generating groups	• *SeqSim+pb* $(JCD, thr \in R^+)$	• *Seq+pb/npb (win_size* $\in N^+$*)* • *SimSIZE (sim_type,* $size \in N^+$*)* • *SimTHR (sim_type,* $thr \in R^+$*)* • *CE+pb/npb* • *CESim (sim_type,* $thr \in R^+$*)* • *SeqSim+pb/npb (sim_type,* $thr \in R^+$*)* • *CESeqSim+pb/npb (sim_type,* $thr \in R^+$*)*
Conclusions	Conclusion 1: Comparing with the summaries composed by sentences, the summaries composed by larger language units have similar ROUGE scores but have better readability Conclusion 2: Using a group of sentences is more effective than using sentence and paragraph. Conclusion 3: The quality of summaries composed by group becomes better when the average length of the source texts increases.	Conclusion 1: summaries composed by *group* or *paragraph* tend to contain more key words or phrases than summaries composed by *sentence*. Conclusion 2: summaries composed by *group* contain more key words or phrases than those based on *paragraph*, especially when the average length of source texts is between 7,000 and 17,000 words. Strategy 1: The *is-part-of* link is a suitable clustering constraint when only using the *sequential* link to generate groups, while in other cases of using the is-part-of link as a clustering constraint has no effect or even has an adverse effect for generating groups. Strategy 2: The lexical-based *similar-to* link is more suitable for generating groups, while the embedding-based *similar-to* link is more suitable for ranking groups. Strategy 3: When only using the *similar-to* link to generate groups, it is better not to restrict the number of sentences within each group but to restrict the degree of *similar-to* between sentences within each group. Strategy 4: The *sequential* link is well suitable to be used together with the *similar-to* link for generating groups. Strategy 5: The *cause-effect* link improves the readability of the *group*-based summary without reducing the amount of key words/phrases contained in the summaries.

[1] M. Cao, H. Zhuge, What size of language unit is more appropriate for text summarization?, in: Proceedings of the 14th International Conference on Semantics, Knowledge and Grids (SKG), 2018, pp. 196-202.

[2] M. Cao, X. Sun, H. Zhuge, The contribution of cause-effect link to representing the core of scientific paper -- The role of Semantic Link Network, PloS one, 13 (6) (2018) e0199303.