E-031

A Method for Corresponding Paragraphs with Sentences in Academic Paper's Abstract

李　元†　　内山　清子‡　　亀田　尭宙†　　相澤　彰子†‡
Yuan Li　　Kiyoko Uchiyama　　Akihiro Kameda　　Akiko Aizawa

1. Introduction

Abstracts of scientific papers play a crucial role for an efficient access to scientific literature. Their conciseness often helps researchers to quickly decide the relevance of the papers to their search objectives. However, due to the diversity of researchers' viewpoints, abstracts sometimes fail to provide sufficient information for such relevance judgment.

In this paper, we propose a method to identify links between each sentence in the abstract with its corresponding paragraphs in the body of the full-text paper. This enables the retrieval system to quickly navigate researchers to more detailed information when they find an interesting statement in the abstract. For this purpose, we first manually analyzed the correspondence between the abstracts and full-text content, and then, propose a method to automatically identify the correspondence.

2. Dataset

In our dataset construction, we selected 30 papers from ACL Anthology Reference Corpus (ACL ARC) (Steven et al. 2008). We first applied Omnipage XML parser (QasemiZadeh et al. 2010) to get clean XML files, and then, manually extracted the correspondence between the sentences in the abstracts and their corresponding paragraphs. In our annotation, we allow many-to-many correspondences; namely, sentences in abstract may have more than one corresponding paragraph, and vice versa. The result is summarized in Table 1.

Table 1 Statistics of papers in dataset. ("*" means "calculated without Abstract section", "Avg." is short of "Average")

Total number of papers:	30
Total number of Abstract sentences:	148
Total number of relations:	229
Avg. number of sentences in an abstract:	4.9
Avg. number of relations in an abstract sentence:	1.5
*Avg. number of sections in a paper:	12.8
*Avg. number of paragraphs in a paper:	43.5
*Total number of paragraphs (with links):	1305 (169)
*Total number of sections (with links):	384 (134)

If there are sub-sections exist in a section, we regard each sub-section and the part from the main section title to the first sub-section title as different "section", therefore, in our corpus, the average number of sections in a paper is 12.8, which is more than the impression of the section number of formal IMRAD papers.

In our dataset, there are 43.5 paragraphs in a paper in average, which means that readers have to check more than 40 independent text blocks in each paper to make sure they won't miss any information they found in the abstract that interested them. However, only 34.9% of sections and 13.0% of paragraphs are related to the abstract (contain links), which means that it's time-saving for readers to avoid those irrelevant full text and directly go to the part they are interested in.

3. Analysis

We classified the corresponding relation between an abstract sentence and a full-text paragraph into the following four types: full match, partial match, citation match and numerical match.

Full match means the abstract sentence and its corresponding paragraph contain more related information than any other paragraph, such as both of them are talking about the effect of the same feature, or the accuracy improvements of the model.

Partial match occurs when an abstract sentence can be clearly split into multiple parts, and each part has an independent topic. In this instance, each part of the abstract sentence corresponds to a different paragraph, and all those paragraphs have the partial match relation with the abstract sentence.

Citation match means that the abstract sentence and its corresponding paragraph contain the same paper reference. This relationship demonstrates that the abstract sentence and corresponding paragraph have the same or related topic.

Numerical match means that the abstract sentence and its corresponding paragraph contain the same numerical string. This kind of relation usually appears in experiment result part. In order to exclude meaningless numbers such as the version number of software, we removed all the numerical strings that appeared more than once in the abstract.

The statistics of these 4 kinds of relations is in Table 2.

Table 2 Statistics of annotated relations in dataset.

	total amount	sentences that contain this relation	papers that contain this relation
Full relation:	181	145	30
Partial relation:	36	16	15
Citation match:	6	5	2
Numerical match:	6	5	4

In our dataset, there are 148 abstract sentences in total, 145 of them contain at least one full match relation, which means that the coverage of full match relation is near 98%.

After analyzing the dataset, we found that there are always some summarized sentences appear in the full-text of paper. Authors tend to rephrase them and then reuse them in the abstract.

Moreover, the occurrence of citation match relation and numerical match relation is few, and most of those occurrences accompany a full match relation that links the same abstract sentence and full text paragraph of that reference of numerical relation.

† The University of Tokyo, UT

‡ National Institute of Informatics, NII

4. Method

For each abstract sentence, we rank paragraphs by its relevance score with the abstract sentence, and then select the most related paragraph as the corresponding paragraph of the abstract sentence. In the following part, we will introduce two methods to obtain the relevance score between an abstract sentence and a paragraph.

PR-ISR:

Chiang (2011) proposed “Paragraph Relevance – Inverse Sentence Relevance” (PR-ISR) method to calculate the relevance between the sentence in the abstract and the paragraph in the corresponding full-text. PR-ISR method uses Part-Of-Speech tag pattern (e.g.: “/NN+/NN(NNP, NNS)”) to extract keywords from the abstract, and then assigns weight to keyword by its occurrence in the abstract. The more sentences that contain this keyword, the lower the weight of this keyword will be. The PR score of an abstract sentence-full text paragraph pair can be obtained by how many keywords they shared, and the ISR score of a paragraph can be obtained by dividing the total number of paragraphs by the sum of the PR score between this paragraph and every abstract sentence, and then taking the logarithm of the quotient. The PR-ISR score is the product of PR and ISR score. We use PR-ISR method as our baseline method.

LCS-based similarity:

We proposed a new method, which is based on the longest common subsequence (LCS) algorithm, to calculate the similarity between each abstract sentence-full text sentence, so as to find the original sentence of each abstract sentence. The paragraph including original sentence will be regard as the corresponding paragraph of the abstract sentence. The definition of similarity is as follows:

$$w_t = \frac{1}{\log(f_t)+1} \qquad (1)$$

$$\text{Similarity (AS, FS)} = \sum_{i=1}^{|t|} t_i \cdot w_{t_i} \times \frac{|CS|}{|AS|} \cdot \frac{|CS|}{|FS|} \qquad (2)$$

$$(\, t \in AS \wedge t \in FS \,)$$

In this definition, t indicates keyword, w_t is the weight of t, f_t is the frequency of t in dataset, AS indicates abstract sentence, FS indicates full text sentence, CS indicates the longest common subsequence.

Characteristic mentions:

We found that specific types of tokes serves as a strong clue by which we can find a paragraph that contains the identical numerical sequence in the full text. Particularly, when an abstract contains citation strings or some numerical sequences, those numerical sequences describe the experiment results in most case, and this paragraph is likely to be a relative paragraph of the abstract sentence. Therefore, as an exception, when a numerical sequence can be found in both abstract and full text, we directly choose the first full text paragraph that contains that numerical sequence as the corresponding paragraph of the abstract sentence that contains the numerical sequence.

5. Experiment

In our experiment, we used Stanford parser to split paragraph into sentences, then used Stanford Parser to annotate the Part-Of-Speech tag for each sentence. Since the coverage of the full match relation is near 98% in our dataset, we only used full match relations in the evaluation of our experiment.

In the evaluation, we consider if one abstract sentence has more than one full match relations, then all the paragraphs in those relations are correct result. If the program found any of those correct paragraphs, then we consider the program got the correct corresponding paragraph of the abstract sentence. If an abstract sentence has no full match relation, all the matched paragraphs by the program are wrong paragraphs for them. The result is shown in Table 3.

Table 3 Experiment result.

	Baseline method	LCS based method
Correct rate	11.5%	43.2%
Correct rate in paragraphs that were predicted in Introduction	13.3%	59.6%
Correct rate in paragraphs that were predicted out of Introduction	8.6%	33.0%

From the experiment result, we found that our method achieved a better correct rate than the baseline method. And our method works better when predicting a paragraph in Introduction than in other sections.

6. Conclusion And Future Work

In this paper, we proposed an LCS based method to correspond the paragraphs in full text with each abstract sentence. Compared with the baseline method, our method achieved a better correct rate. Since authors tend to reuse sentences in the introduction, our LCS based method works better when predicting a related paragraph in Introduction section than in other sections.

In our following research, we plan to apply lexico-syntactic pattern to improve the correct rate in semantic way. Moreover, a proper way would be introduced to identify and split one sentence into multiple sub-sentences.

References

S. Bird, R. Dale, B. J. Dorr, B. Gibson, M. T. Joseph, M.-Y. Kan, D. Lee, B. Powley, D. R. Radev, and Y. F. Tan. 2008. The ACL anthology reference corpus: A reference dataset for bibliographic research. Proceedings of the Language Resources and Evaluation Conference (LREC 2008)

QasemiZadeh, B., Buitelaar, P. and Monaghan, F. 2010. Developing a dataset for technology structure mining. IEEE Fourth International Conference on Semantic Computing.

Jung-Hsien Chiang, Heng-Hui Liu and Yi-Ting Huang. 2011. Condensing biomedical journal texts through paragraph ranking. Bioinformatics 27, 1143-1149.