


Functional structure identification of scientific documents in computer science

Wei Lu¹ · Yong Huang^{1,2} · Yi Bu² · Qikai Cheng¹ 

Received: 14 September 2017 / Published online: 2 February 2018
© Akadémiai Kiadó, Budapest, Hungary 2018

Abstract The increasing number of open-access full-text scientific documents promotes the transformation from metadata- to content-based studies, which is more detailed and semantic. Along with the benefits of ample data, the confused internal structure introduces great difficulties to data organization and analysis. Each unit in scientific documents has its own function in expressing authors' research ideas, such as introducing motivations, describing methods, stating related work, and drawing conclusions; these could be used to identify functional structure of scientific documents. This paper firstly proposes a clustering method to generate domain-specific structures based on high-frequency section headers in scientific documents of a domain. To automatically identify the structure of scientific documents, we categorize scientific documents into three types: (1) strong-structure documents; (2) weak-structure documents; and (3) no-structure documents. We further divide the identification into three levels—section header-based identification, section content-based identification, and paragraph-based identification—corresponding to the three types of documents. Our experiments on documents in the field of computer science show that: (1) section header-based identification is the most direct and simplest method, but its accuracy is limited by unknown words in section headers; (2) section content-based identification is more stable and obtains good performance; and (3) paragraph-based identification is promising in identifying functions of no-structure documents. Additionally, we apply our methods to two tasks: academic search and keyword extraction. Both tasks demonstrate the effectiveness of functional structure.

Keywords Functional structure · Text categorization · Academic retrieval · Keyword extraction

✉ Qikai Cheng
chengqikai@whu.edu.cn

¹ Information Retrieval and Knowledge Mining Laboratory, School of Information Management, Wuhan University, Wuhan, Hubei, China

² School of Informatics, Computing, and Engineering, Indiana University, Bloomington, IN, USA

Introduction

As the accessibility of machine-readable scientific documents (in XML or HTML format, not PDF) has increased, full text-based fine-grained scientific documents analysis has dominated in recent years. Along with the benefits of ample data, the confused structure of scientific documents introduces great difficulties to data organization and analysis. Therefore, it is of importance to identify structures in scientific publications. Fortunately, the internal units of scientific papers, such as sections and paragraphs, are arranged for the expression of the authors' ideas in accordance with a certain logic. Each unit in scientific documents has been found to have its own latent function, such as introducing motivations, providing background, describing methods, presenting results, and drawing conclusions (Zhang 2012). In this paper, we define these kinds of latent functions as structural functions. These functions define the roles of sections or paragraphs in transmitting the main ideas of authors and constitute papers' generic structure, defined as functional structure.

To automatically identify the functional structure of scientific documents, we first categorize scientific documents into three types: (1) strong-structure documents; (2) weak-structure documents; and (3) no-structure documents, and then divide functional structure identification into three levels corresponding to three types of scientific documents.

Section header-based identification This is the most commonly applied method in current research and is applicable for strong-structure documents. Since section headers are structural function tags labeled by authors, this is the most direct and simplest method.

Section content-based identification The efficiency of section header-based identification will dramatically decrease with weak-structure documents, of which section headers contain unknown words and are not arranged in an orderly manner. In this circumstance, the section content will provide more information. In this paper, we treat section content-based identification as a text categorization problem and propose three kinds of features (i.e., lexical, clustering, and pattern features).

Paragraph-based identification There is no section structure in no-structure documents, such as scientific reports. Under such circumstances, only paragraphs could be used as a basis of identification. In this paper, paragraph-based identification is also regarded as a text categorization problem.

We implement all of the aforementioned methods on scientific documents in the field of computer science. Finally, we apply our method to two tasks: academic search and keyword extraction, both of which show that functional structure we identify is useful and promising. The main contributions of this paper are as follows:

- We propose an algorithm used to generate a domain-specific functional structure. This enables our schemas to be more suitable to data used in the experiments.
- Our three-level identification methods could identify all types of scientific documents. This makes our research more comprehensive.
- Two tasks of academic search and keyword extraction demonstrate its potential in applications of our algorithm.

The rest of the paper is organized as follows. “[Related work](#)” section discusses related work regarding functional structure schemas and identification methods. “[Methodology](#)” section elaborates the definition of structural function, the algorithm used to generate domain-specific schema, and the identification levels. “[Experiments and results](#)” section presents the experiments of the three levels of identification. “[Result](#)” section showcases

two potential applications of our proposed algorithm. “[Applications: two examples](#)” section concludes the whole paper and identifies the directions for future work.

Related work

Structure schemas

The classic schema “IMRD”, which comprises “introduction”, “method”, “result”, and “discussion”, was derived from linguistics (Day 1989; Sollaci and Pereira 2004). The schema is used to teach new students to write scientific papers, and has been adopted to analyze citation distribution (Hu et al. 2013) and information use (Zhang 2012). In addition to “IMRD”, there are some schemas proposed by other researchers, among whom Nguyen and Kan (2007) utilized 14 generic headers, including “abstract”, “categories and subject descriptor”, “general term”, “introduction”, “background”, “method”, “conclusion”, “reference”, “evaluation”, “related work”, “acknowledgment”, “applications”, “motivation”, and “implementation” to represent generic structures, and applied them in keyword extraction. Ding et al. (2013) defined a schema, including “abstract”, “introduction”, “literature review”, “method”, “result”, and “conclusion”, and studied the difference of citation behavior between different functions. From the existing structure schemas, one can find that there is no uniformed structure, and the existing schemas have been proposed based on authors’ experience or have simply taken from the field of linguistics. All of these, however, might be unsuitable for the actual data from a given field. In this paper, therefore, we propose a clustering algorithm to generate a domain-specific functional structure based on high-frequency section headers which will be discussed in “[Methodology](#)” section.

Identification methods

The identification of previous work is a classification problem which comprises mapping sections of an article to functions defined in their schemas. Section headers de facto constitute the main evidence for classification. Since the dictionary is built manually, there could exist some unclassified section headers (Ding et al. 2013; Hu et al. 2013). Moreover, Nguyen and Kan (2007) applied support vector machine (SVM) and Maxent to classify section headers to their defined functions. Meanwhile, in Councill et al. (2008)’s work, they used Parscit and treated identification as a sequence-labeling problem by employing conditional random fields (CRFs) to recognize the structure of research articles with three kinds of features, position feature, word feature and whole header; this research is an important foundation of our section header-based identification approach. Regardless of which method is used, the features useful in identification are frequently occurring words such as “introduction”, “method”, and “result”. The accuracy of identification would be greatly reduced if these high-frequency words are absent in section headers. For instance, in many social science publications, regardless of qualitative or quantitative articles, the word “finding” has been employed as a header instead of “result”, which reflects the drawbacks of the algorithms in which rigid structure of section titles is employed.

Unlike section header-based identification, section content is more general. The word distribution of section content determines the functions of this section, and it is a text

categorization problem. The most commonly used classification model for text categorization problems is SVM (Cortes and Vapnik 1995), which is also applied in our research. Yang and Pedersen (1997) demonstrated that information gain is the most stable feature selection method when the classifier is SVM.

On the other side, from the perspective of paragraph-based identification, its main purpose is to identify the functional structure of no-structure documents. In linguistics, a large body of extant research (Li and Ge 2009; Martin 2003; Nwogu 1997) exists under the topic of genre analysis (Swales 1990). Nevertheless, in this study we do not explore paragraphs’ function in each section with different functions. Instead, we assume that different types of scientific documents in the same field share the same functional structure. Paragraph-based identification is also regarded as a text categorization problem. The details of the clustering algorithm and identification will be explained in the following.

Methodology

Definitions

Scientific documents have different structure levels, as shown in the left side of Fig. 1. Sentences, for instance, are the minimal semantic unit (Leydesdorff 2001). From bottom to top, sentences constitute paragraphs and sections, sections constitute documents, and documents constitute journals as well as domains, the structure levels become increasingly general. Moreover, the word distributions of each layer convey different kinds of knowledge. At the journal and domain levels, we often study theories, trends, and popular research topics by using highly-frequent words, matrices of co-occurred words (Ding et al. 2001), or cross-section analysis (De Sordi et al. 2017). At the document level, word distribution reveals authors’ research ideas and research topics. At the sentence level, relations between entities, syntax trees, etc., could be discerned. At the internal level, sections and paragraphs are arranged to clearly express papers’ topics. Moreover, each internal unit has its own special function. We provide an example of this in the right side of Fig. 1, in which the first section introduces the paper’s motivation, the literature review (related work) is presented in the second section, the third section describes the implemented methods, the fourth section presents the experiments and shows the results, and the last section concludes the

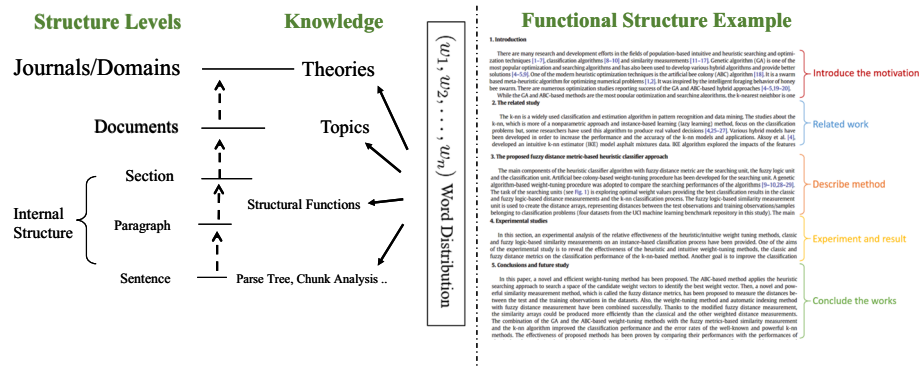


Fig. 1 Illustration of structure levels of scientific documents (left side) and structural functions that constitute functional structure (right side)

article. We define such kinds of functions, like “introduce motivation”, “related work”, “method”, “experiment and result”, and “conclude the work” as *structural functions*. These functions define the roles of sections or paragraphs in transmitting the main ideas of authors. These structural functions constitute papers’ generic structure, defined as *functional structure*. In this paper, we simply focus on the main content of scientific documents, i.e., some other parts, such as “abstract”, “acknowledgments”, and “references” will not be included in our functional structure.

Domain-specific functional structure schema

As mentioned before, the schemas proposed in previous researches are either from linguistics or generated based on the author’s research experience. It is therefore difficult to determine whether these schemas are suitable for particular experimental data without the support of actual data. Moreover, every domain has its unique logic of problem-solving, which inevitably leads to different conventions of paper-writing. Since functional structure schemas are not compatible between domains, an algorithm that can generate a functional structure schema based on used data is necessary. Keywords of scientific papers are often treated as research topics, and high-frequency keywords in a domain are used as main research topics of that domain (Wang et al. 2014). Similarly, section headers of scientific papers could be regarded as function tags labeled by authors, and high-frequency section headers in a domain could represent that domain’s main structural functions. Based on this idea, a clustering method is proposed. The procedures of this algorithm are shown in “Algorithm 1”. As shown in “Algorithm 1”, our input are papers collected from a specific domain, e.g., computer science, and the output is the functional structure of that domain. First, section headers of the input papers are extracted and ranked based on their frequency in descending order. Then, the top N high-frequency section headers are selected to construct a section header collection $H(h_1, \dots, h_i, \dots, h_N)$. Note that the value of N affects the result of this algorithm. The section header collection would include too many trivial section headers if N is too large; the coverage and accuracy of headers could be negatively affected if N is too small. In this paper, we set N to 50. The following steps, from steps 3 to 10, describe the details of clustering section headers to some specific function clusters. The judgment in step 6 is the key of this algorithm, in which we first obtain the function F_i of header h_i through equation $F_i = f(h_i)$, and then search it in the functional structure S . In this paper, the procedures of clustering are performed manually. The reasons lie in threefold: First of all, only 50 headers need to be judged, and this takes just a few hours to do manually. Secondly, this algorithm will not only consider the specific conditions of the data that we use, but also combines writing conventions of that domain by introducing manual effects. Finally, this algorithm combines authors’ intuition and information from real data to generate a domain-specific functional schema S .

Algorithm 1 Algorithm to generate domain specific functional structure

Input: Papers from a specific domain

Output: Functional structure S

- 1: Extract section headers from input papers and rank them based on their frequency in descending order.
 - 2: Select the top N high frequency section headers to construct a section header collection $H(h_1, \dots, h_i, \dots, h_N)$ and create a functional structure S , a empty function collection.
 - 3: $i = 0$
 - 4: **while** $i < N$ **do**
 - 5: $i = i + 1, LEN = len(S)$
 - 6: Judge whether the function of $h_i, F_i = f(h_i)$, could be found in S :
 - 7: **if** F_i is not in S **then**
 - 8: Create a new function cluster s_{LEN+1} , set its label to F_i , add h_i into it. And then insert this new cluster to S .
 - 9: **else**
 - 10: Add h_i to that function cluster.
 - 11: **return** S
-

We implement our algorithm on articles from the field of computer science. The research articles from computer science are full-text access articles collected from ScienceDirect¹ within computer science and used to create dataset CS. The number of articles in CS is approximately 130,000. All experiments in the following are performed on this dataset.

We apply this algorithm to all research articles from CS, and obtain a functional structure schema (as shown in Table 1). There are five functions in computer science's functional structure.

Introduction This is usually the first section in the main content, and is used to illustrate the motivation and problems, and present an overall picture of the whole paper.

Literature review Sections with this function are used to discuss related work and the background of the current research.

Method This is the core section of the paper, and often describes authors' ideas, models, and algorithms.

Experiment and result This function is used to present the details of experiments, evaluations, experiment results, performance analysis, etc.

Discussion and conclusion This section is used to conclude the paper, identify limitations, and present directions for future work.

This functional structure of scientific documents we propose here is general from the field of computer science, and describes the internal sections' structural function. The functional structure is useful because it constitutes the navigation of different kinds of knowledge. The importance of terms and citations in specific functions differs. For example, in CS, papers cited in "method" may be more important than papers cited in "literature review", as they provide more methodological trajectories. Moreover, terms

¹ <http://www.sciencedirect.com/science/journals/sub/computerscience/all/full-text-access>.

Table 1 The functional structure schema generated from computer science using the clustering algorithm

Section headers	Functions
Introduction, preliminaries	Introduction
Related work, background, related works, literature review	Literature review
Methods, methodology, materials and methods, method, problem formulation, problem statement	Method
Experiments, performance evaluation, evaluation, implementation, examples, numerical examples, experimental evaluation, numerical experiments, experimental, experiment, results, experimental results, results and discussion, simulation results, numerical results, experiments and results, results and discussions, main results, performance analysis	Experiment and result
Discussion, discussions, conclusions, conclusion, conclusions and future work, concluding remarks, conclusion and future work, summary, discussion and conclusions, summary and conclusions, discussion and conclusion, future work	Discussion and conclusion

There are five kinds of functions in this schema. The left column is the section header set and the right column is the corresponding function name, which are labelled manually based on the headers in the left set

that occur in different locations can also have different semantics. So, it is meaningful to automatically identify the functional structure.

Automatic identification

The second problem to be addressed is the automatic identification of functional structure. Since there are various types of scientific documents such as research articles, surveys, technical reports, etc., we use four questions below to categorize all scientific documents into three categories, which are strong-structure, weak-structure, and no-structure documents:

- *Section* Are there section units in scientific documents?
- *Paragraphs* Are there paragraph units in scientific documents?
- *Orderly arranged* Are the sections arranged in an orderly manner?
- *Within collection* Are most section headers in the high-frequency section header collection (i.e., the left column of Table 1)?

As shown in Table 2, sections and paragraphs are orderly arranged, and most section headers are included in the high-frequency header collection in strong-structure documents, such as research articles. Weak-structure documents, such as surveys, have clear

Table 2 The four questions listed are used to assess the difficulty and differences between three types of documents

Types/measures	Sections	Paragraphs	Orderly arranged	Within collection
Strong structure	Y	Y	Y	Y
Weak structure	Y	Y	N	N
No structure	N	Y	N	N

It is given a “Y” if the answer is yes and an “N” if the answer is no

sections and paragraphs, but section headers are not always in the high-frequency section header collection or arranged in an orderly manner. No-structure documents are often short scientific papers, such as scientific reports, which have limited section structure, but only paragraphs, and still have strict internal logic from introducing motivations to providing conclusions.

To automatically identify all three categories of documents, we divide functional structure identification into three levels: (1) section header-based identification; (2) section content-based identification; and (3) paragraph-based identification. The solution for the three identification levels are as follows:

Section header-based identification Section header-based identification is mainly used for functional structure identification of strong-structure documents. Section headers are the observations of latent structural functions and, respectively, denoted as $X(x_1, \dots, x_i, \dots, x_m)$ which might range over all possible headers and $Y(y_1, \dots, y_i, \dots, y_n)$ which might range over the functions in functional structure S . In strong-structure documents, such as research articles, section headers are orderly arranged, and can be thought of as an observation sequence, and the functions are the latent variable sequence. For example, in research articles in the field of computer science, sections with the “related work” function usually follow sections with the “introduction” function, and sections with the “experiment” function follow sections with the “method” function. Consequently, section header-based identification could be treated as a sequence-labeling problem, such as part of speech (POS) and named entity recognition (NER). Given a sequence of section headers, the objective of this problem is to find the most probable latent function sequence. In other words, our aim is to compute a function sequence Y^* to maximize the conditional probability $P(Y|X)$. So, a linear-chain CRF (Sutton et al. 2012) takes the form

$$P(Y|X) = \frac{1}{Z(X)} \prod_{t=1}^T \exp \left\{ \sum_{k=1}^K \theta_k f_k(y_t, y_{t-1}, x_t) \right\}$$

where $Z(X)$ is an instance-specific normalization function

$$Z(X) = \sum_y \prod_{t=1}^T \exp \left\{ \sum_{k=1}^K \theta_k f_k(y_t, y_{t-1}, x_t) \right\}$$

could be used in this problem. Here, $f_k(y_t, y_{t-1}, x_t)$ is the feature function; T is the length of input section headers; K is the number of feature function; and θ is the parameters of this model. This linear-chain CRF (Lafferty et al. 2001) assumes that the latent function y_t at position t only depends on the observation header x_t and the former latent function y_{t-1} . In this paper, a linear-chain CRF tool CRF++² is used in the following experiments. Similar to Luong, Nguyen and Kan (2012), the features used in section header-based identification are listed as follows:

The absolute position and relative position of section header The absolute position is the section index in the document. For instance, in the paper you are reading, “Methodology” is the third section. The relative position comprises dividing the absolute position into 10 parts. For instance, in this paper, “Methodology” features a

² <https://taku910.github.io/crfpp/>.

relative position of $3/6 \times 10 = 5$ where three is the absolute position and six is the total number of sections.

The first two words and the last two words of section header Keywords, such as “introduction”, “method”, and “experiment” are the most useful features. The four words would be set to the first word if the section header contains only one word. For example, given the section header “introduction”, the four words will all be set to “introduc”. PorterStemmer is used here (Porter 1980).³

The whole section header Let the model remember headers that it has already encountered. The words in a header will be stemmed and joined with “_”. For example, the section header “related work” will be processed as “relat_work”.

Section content-based identification Section content offers more information than section headers. Section content-based identification means to map word distributions to their latent functions. It constitutes a text categorization problem that can be represented as the following equation:

$$y_i = g(f(X_i))$$

The input represented as X_i is the content of a section, and the function $f(X_i)$ is used to extract features from input. Then, a classifier $g(z)$ outputs the latent structural function y_i , which is a range over structural functions in functional structure S . The classifier $g(z)$ used in this paper is the support vector machine (SVM), which is a very stable classifier and commonly used in text categorization problems (Joachims 1998). Since the speed of SVM with kernels will be very slowly trained on large-scale data, we use the linear SVM tool LIBLINEAR (Fan et al. 2008) as a classifier instead. LIBLINEAR is a tool developed especially for large-scale text categorization problems. It will be much faster than SVM with kernels, but with similar performance. Three kinds of features (lexical feature, clustering feature, and pattern feature) are proposed here, and are concatenated and put into the classifier. The details of data construction, feature selection, and experiments will be elaborated in the next section.

Paragraph-based identification The purpose of paragraph-based identification is to identify the structure of no-structure documents. For a no-structure document, there are only paragraphs in its main content. The identification of its functional structure is the process of labeling functions of paragraphs. It is also a text categorization problem represented as section content-based identification. The input X_i here is the content of a paragraph, and the output y_i is the structural function of that paragraph. The features $f(X_i)$ and classifier $g(z)$ could be the same as section content-based identification. Compared to section content-based identification, the average length of the content of paragraphs will be much smaller, which will inevitably lead to a reduction of accuracy of identification, even with the same method and the same features.

In this section, we have divided the identification into three levels: (1) features; (2) data; and (3) procedures of the experiment. The details of the results of these three levels will be presented in the next section.

³ <https://tartarus.org/martin/PorterStemmer/>.

Experiments and results

Data

We randomly select 300 research articles from dataset CS, and articles in which the number of sections is greater than ten or smaller than three are omitted. Two students majoring in information science are invited to label the selected articles. The Kappa value of the labeling result is 0.81. For samples with different labeling results, the author selects one as the final label.

Section content-based identification is a text categorization problem, of which the features have thousands of dimensions, and the classifier cannot be fully trained based only on 300 samples. It is almost impossible to manually build a sufficiently large dataset for it. Therefore, it is necessary to automatically generate large amounts of data, rather than relying on manual labeling. As discussed in the previous sections, section headers could be treated as structural function tags labeled by its author. In other words, since the structural functions of sections with these specific section headers (section headers in the left column in Table 1) are already known, the dataset of section content-based identification could be constructed based on Table 1. Section content with headers in the left column of Table 1 will be extracted and labeled with corresponding structural functions in the right column. In our CS dataset, there are more than 320,000 samples, including 120,000 “introduction” sections, 20,000 “related work” sections, 15,000 “method” sections, 50,000 “experiments” sections, and 120,000 “conclusion” sections are extracted from CS. We randomly select 5,000 samples for each label in computer science’s functional structure to achieve a balance and sufficiently large dataset, which is named CS_SEC.

In terms of the paragraph-based identification, traditionally, to construct a dataset for that, it is necessary to randomly select several no-structure scientific documents, and label the functions of paragraphs in these papers manually. Obviously, manually-labeled data cannot make our model be fully trained. From the section content-based identification experiments, it is determined that the distribution of non-topic words affects functions. Therefore, we assume that paragraphs with the same function in different categories of scientific documents share similar non-topic word distribution. In this way, paragraphs of known function in strong and weak documents could be used as training data. The same as the data construction method of section content-based identification, datasets of paragraph-based identification could also be constructed by using Table 1, in which titles have been tagged with their functions. We first select all sections with the titles in the left column of the two tables, and then label the paragraphs in every section with the function in the right column. We extract samples from articles of CS based on this idea. More than 1,990,000 samples are extracted, and the number of samples of each function is greater than 100,000. To construct balanced datasets, for two domains, we randomly select 10,000 samples for each function from the samples. Finally, a dataset for computer science, named CS_PARA and including 50,000 samples, is constructed.

Section header-based identification

In our experiment, we only use the unigram feature in the CRF++ template. Five-folder cross validation experiments are implemented. We used three indicators that are precision abbreviated as PRE, recall abbreviated as REC, and F1-measure abbreviated as F1 to measure the results of our experiments.

For computer science, as shown in Table 3, the F1-values of “introduction” and “discussion and conclusion” are greater than 90%. In most cases, section headers of these two functions are relatively stable. The F1-value of the other two functions are greater than 80%. For “related work” and “method” sections, the authors may use the specific name of their research topics. For example, a paper related to language modeling in information retrieval names their “related work” with “language modeling”, and names their “method” with “modification of language model”.

From the results of section header-based identification, it can be found that the performance of functions with stable section headers, such as “introduction” and “conclusion”, are better than others. If words in section all occur frequently contained in the section header collection (the left column in Table 1), irrespective of whether the dictionary-based or CRFs-based method is used, we will obtain good performance. In other words, once too many unknown words appear in section headers, performance will be substantially degraded. This makes sense to some extent, because the efficiency of section header-based identification depends on the percentage of high frequency words in section headers. Essentially, section header-based identification is particularly useful when the section headers are in a control set.

Section content-based identification

Feature selection

Three kinds of features in identifying section base are proposed in this paper, the details of which are listed as follows.

Lexical feature

Lexical feature is the most frequently used feature in text categorization problems. For structural function identification, the word distributions of sections with different functions are dissimilar. In addition, the vocabulary contains thousands of words, and it will be inefficient if we use all of the words in the vocabulary. The main task here is to select the most effective words. The steps are listed as follows:

- Preprocessing: Lowercase all the words, remove all punctuations and numbers, and stem words with PorterStemmer.
- Feature selection: Information gain (IG) is used as the feature selection method. Yang and Pedersen (1997) demonstrated that IG is the most stable method in text categorization when the classifier is SVM. The feature value used in feature selection is one when a term appears in the section content or zero otherwise.

Table 3 Result of section header identification of CS

Label	PRE	REC	F1
Introduction	0.9968	0.9777	0.9871
Related work	0.7159	0.5294	0.6087
Method	0.6726	0.8267	0.7417
Experiment and result	0.7849	0.677	0.7269
Discussion and conclusion	0.9165	0.9005	0.9084
AVG	0.8173	0.7823	0.7946

- Feature extraction: The feature value used in this step is term frequency (TF), which is the number of term occurrence normalized with the length of the section content. We do not use the TFIDF here because the feature value would be scaled prior to being put into the SVM classifier. The scale method is (feature value-minimum)/(maximum-minimum). Since the minimum of all terms is zero, the real scale formula is (feature value/maximum). The IDF is the same for one specific term, and thus the scaled TFIDF value is equal to the scaled TF value.
- Optimal number of words: Words are ranked based on their IG score in descending order. We use the top 10,000 words as the initial feature. We choose the most appropriate word number in the range of [1000, 10,000] with an interval 1000. In this paper, LIBLINEAR is used as the classifier, and the parameter c in LIBLINEAR is selected from the range of [0, 1] with an interval 0.1. Five-fold cross validation experiments are conducted with a varying set of parameters. Figure 2a presents a line chart based on the results, which displays the accuracy trend over the number of words. 3000 words are selected as the final lexical feature. Although, accuracy will be greater when the number of words is greater than 3000, there is no significant improvement.

The lexical feature is effective here and obtains 82% accuracy. The lexical feature is always effective in text categorization problems. The words selected in different text categorization tasks are supposed to be different. To identify words that are effective in our experiment, we collect the top 50 words in the ranked IG list, as shown in Table 4.

As shown in Table 4, most of these words are non-topic words, such as “figure”, “table”, “paper”, “show”, “compare”, etc. This is totally different from topic-oriented text categorization, such as topic classification, in which words used as features should be topic-related. Section content-based identification is not a topic-oriented text categorization problem, but rather a structure-oriented text categorization.

Clustering feature

Leydesdorff (2001) asserted that there are some latent factors, such as explanatory, methodological, and conclusive factors, that affect word distribution and make sections to show specific functions. For “explanatory”, words are usually used to explain some theories and frequently occur in “introduction”. In addition, “method” words are used to describe the method and “conclusion” words are used to conclude the papers (Zhang 2012). The latent factor of a word depends on its context. Thus, we could use the clustering algorithm to cluster words with similar functions based on their context. However, there will be a sparsity problem if we cluster words based on their context word by a one-hot feature. Word embedding is a new technology that uses a vector composed of multidimensional float values to represent the semantic of a word. Word embedding is usually pre-

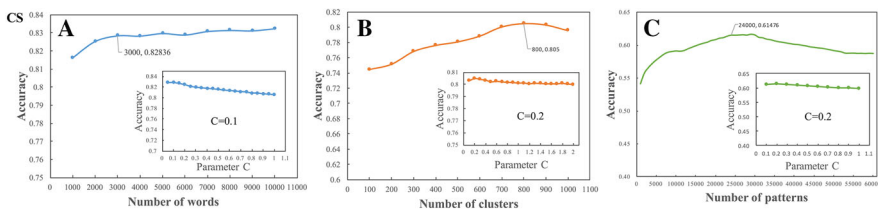


Fig. 2 Parameter estimation result of section content-based identification on CS_SEC

Table 4 The 50 words in the ranked IG list

 Top IG words

fig, table, each, recent, et, paper, al, follow, all, were, section, shown, average, organ, show, respect, value, total, they, been, three, after, then, at, from, test, research, calculus, as, plot, two, four, same, their, many, or, propose, such, however, between, was, define, sample, figure, by, eq, repress, are, had, five

trained on a sufficiently large dataset, and then the vectors of words are used directly in specific experiments. We cluster the words in sections based on word embedding pre-trained with Word2vec (Mikolov et al. 2013) on the scientific documents from CS. The window size of Word2vec is five, and the dimension of word embedding is 200. K-means is used to cluster all words into several clusters based on their word embedding. Moreover, the number of clusters that should be used is an important variable. The steps to select the most effective number of clusters are listed as follows:

- Train word embedding with Word2vec on the main content of the papers collected from computer science and medicine, respectively.
- Clustering the words based on word embedding with k-means and obtain N clusters. To select the best N, we try numbers in range of [100, 1000] with 100 as an interval.
- Extract feature value: The value metric used here is cluster frequency (CF), which is the occurrence number of words in one cluster normalized by the length of the content.
- Draw a line chart to display the accuracy trend over the number of clusters shown in Fig. 2b.

As shown in Fig. 2b, accuracy reaches its highest point when the number of clusters is 800. The accuracy of the clustering-feature based experiment on CS_SEC is lower than the lexical-feature based experiments.

Pattern feature

The sentences in papers follow some patterns that differ among sections with different structural functions. Sentence pattern triples are composed of a subject word, a verb word, and an object word. In this paper, we use ReVerb (Fader et al. 2011), an open information extraction tool, to extract triples.⁴ ReVerb is a program that automatically identifies and extracts binary relationships from English sentences. For example, in the following sentence:

In this paper, we have introduced fuzzy job shop scheduling problems by incorporating the fuzzy processing time and fuzzy due date.

A triple “(we, introduce, fuzzy job shop scheduling problems)” is extracted. The pattern would be very sparse if we used the triples extracted directly. The object or the subject words in triples are usually topic-related, the same as the “fuzzy job shop scheduling problems” in the above example. We transform one triple into two bi-grams, i.e., the triple (subject, verb, object) will be transformed into two bi-grams (subject, verb) (verb, object).

⁴ <http://reverb.cs.washington.edu/>.

Patterns extracted from this example are “(we, introduce)” and “(introduce, fuzzy job shop scheduling problems)”. The first one is the general pattern, which would be more useful in our identification experiment, and the second is the topic-related pattern.

As shown in Table 5, some patterns will appear in sections with specific functions. For example, the patterns “(we, propose)”, “(we, present)”, “(we, use)”, and “(we, consider)” appear in “introduction”, and are used to generally introduce methods, ideas, and algorithms utilized in the paper. In “related work”, patterns “(they, use)” and “(they, propose)” will be used to explain related work. Some other typical patterns used in specific structural functions are identified in bold in Table 5. Since the patterns are different between functions, it would be useful in our identification.

Like the lexical feature, we use information gain to select the most effective K patterns in this task. The accuracy trends over the number of patterns K are presented in Fig. 2c. The value of K is selected from a range of [1000, 60,000] with an interval 1000.

As shown in Fig. 2c, we select the top 24,000 patterns in the ranked pattern list for computer science. The value of accuracy is relatively lower than the other two features.

Result

As mentioned before, three kinds of features are proposed: (1) optimal number of lexical features; (2) optimal number of clusters; and (3) optimal number of patterns. In this section, we set up three groups of experiments to test the effectiveness of various combinations of features. The combinations of features are listed in Table 6. The first group, experiment I, contains three experiments of which each only used one kind of feature. Three experiments combining two kinds of features are included in the second group, experiment II. For experiment III, all three types of features are utilized. In these experiments, the optimal features and parameters selected in the last section are used, and LIBLINEAR is the classifier. We also implement these three groups of experiments on CS_SEC.

From the result of Fig. 3, it can be seen that the more features we used, the better the effect. The performance of the lexical feature is better than the clustering feature, and the clustering feature is better than the pattern feature. Moreover, combinations of two kinds of features perform better than one feature. Combinations of three kinds of features have the highest score compared to others. These results demonstrate that all three kinds of features that we propose could be useful in section content-based identification.

To describe the result more concisely, we display the result of IIIA on two datasets in Table 7. Section content-based identification has better performance on “experiment and result” than on “introduction” and “discussion and conclusion”. The score of “experiment and result” is approximately 89%, which is the highest score. The score of “introduction” and “discussion and conclusion” is relative lower. This is just the opposite to section header-based identification. The average F1-measure is 86%.

From the details of the results, one can find that various functions obtain a relatively balanced performance, and the lowest F1 score of computer science is greater than 84%. It is demonstrated that section content-based identification is more general and stable for strong structure and weak structure.

Table 5 Ten most frequent patterns in each structural function of computer science

Functions	High-frequency patterns
Introduction	(we, propose) (we, present) (we, use) (we, consider) (it, be) (section #, describe) (section #, present) (we, introduce) (present in, section #) (we, describe)
Related Work	(it, be) (we, use) (they, use) (they, propose) (the authors, propose) (this, be) (take into, account) (we, consider) (it, use) (we, propose)
Method	(we, use) (be, #) (we, consider) (it, be) (we, define) (show in, fig.) (this, be) (. #, show) (show in, fig. #) (we, propose)
Experiment and Result	(we, use) (be, #) (. #, show) (show in, fig. #) (show in, fig.) (we, compare) (be, # %) (we, consider) (fig. #) (table #, show)
Discussion and Conclusion	(we, propose) (we, present) (we, use) (this, be) (it, be) (we, consider) (we, develop) (make, it) (be, #) (take into, account)

The typical patterns have been marked in bold

Table 6 Three groups of experiments are listed

Groups	Index	Feature combinations
Experiment I	IA	Lexical feature
	IB	Clustering feature
	IC	Clustering feature
Experiment II	IIA	Lexical feature + clustering feature
	IIB	Lexical feature + pattern feature
	IIC	Clustering feature + pattern feature
Experiment III	IIIA	Lexical feature + clustering feature + pattern feature

For experiment I, there is only one feature proposed in the last section used in each experiment

For experiment II, three groups of combinations of two kinds of features are set. For experiment III, all three kinds of features are used. For all groups of experiments, LIBLINEAR is used as the classifier. For each kind of feature, the optimal feature and parameter selected in the last section are used

Paragraph-based identification

We use methods of section content-based identification directly here. The details of methods and procedures of feature selection, feature extraction, parameter optimization, and experimental settings will no longer be stated in this section, i.e., only the results of these procedures will be elaborated.

Feature selection comprises selecting optimal parameters of three kinds of features: (1) the optimal number of words used in lexical feature; (2) the optimal number of clusters used in clustering feature; and (3) the optimal number of patterns used in pattern feature. We implement the feature selection on CS_PARA, and the result is shown in Fig. 4. The three sub-figures display three parameter selection results, respectively, and the lower three sub-figures show the results of medicine. Figure 4a is the accuracy trend over the number of words in the ranked IG list. We select the top 8000 words as the final lexical feature. Figure 4b displays the accuracy trend over the number of clusters. Obviously, the classifier

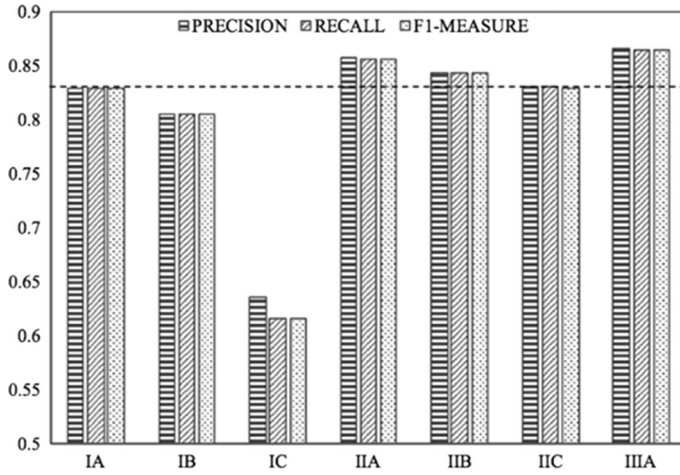


Fig. 3 Result of experiments on dataset CS_SEC. Three evaluation metrics (precision, recall, F1-measure) are used in these experiments and represented with different textures. As can be seen, the more features we used, the better result we obtained

Table 7 Result of section content-based identification on CS_SEC

Label	PRE	REC	F1
Introduction	0.8661	0.8250	0.8450
Related work	0.8428	0.8448	0.8438
Method	0.8429	0.8984	0.8698
Experiment and result	0.8937	0.8942	0.8939
Discussion and conclusion	0.8794	0.8606	0.8699
AVG	0.8650	0.8646	0.8645

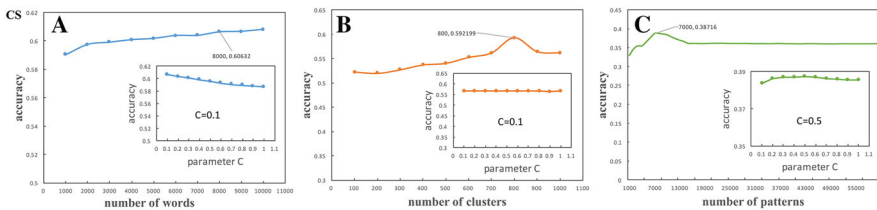


Fig. 4 Parameter estimation result of paragraph-based identification on CS_PARA dataset

has the best performance when the number of clusters is 800. Figure 4c shows the feature selection results of the pattern feature, and 7000 patterns are selected as the pattern feature.

We have selected the optimal parameters of three kinds of features on CS_PARA. Different combinations of three kinds of features, which are the same as the experimental settings in Table 6, are also implemented on these two datasets. The results are shown in Fig. 5, in which we can find that the lexical feature performs better than the clustering feature, and the clustering feature is better than the pattern feature. In addition, combinations of features perform better than only one kind of feature. Combinations of three kinds of features have the best performance. This result is consistent with that of section

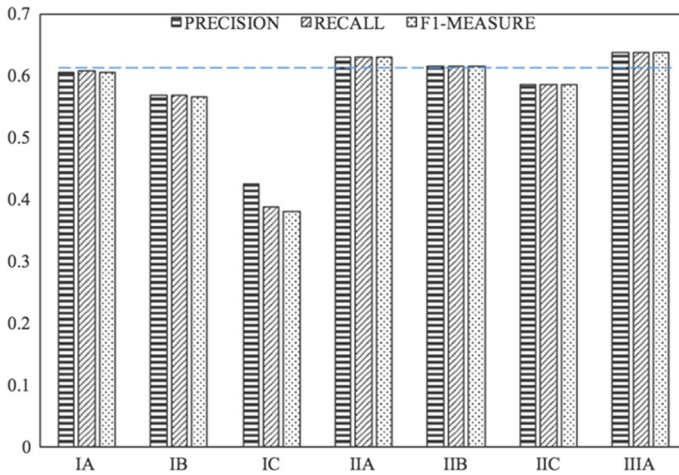


Fig. 5 Comparison result of different combinations of features

content-based identification, which is our expectation. The detailed results of combinations of three kinds of features are listed in Table 8, where the average F1-measure score is approximately 0.63, and the scores of each function are not bigger than 0.7.

From the result, it can be seen that accuracy is lower than section content-based identification. In addition, the average paragraph length is much smaller than a section, and the features extracted are sparser. Moreover, similar paragraphs could appear in sections with different functions. For example, there will be some summary paragraphs in “introduction” sections and “discussion and conclusion” sections, and there are also some paragraphs describing methods in “related work” and “method”. Consequently, paragraphs will be more likely to be misclassified as each other. Paragraph-based identification is utilized to identify the functional structure of no-structure scientific documents. This finding is very important and usually ignored in the extant literature.

Applications: two examples

Our proposed algorithm has several potential applications. Here we provide two of them, academic research and keyword extraction, in detail for future researchers who are interested in our algorithm.

Table 8 Result of paragraph-based identification on CS_PARA

Label	PRE	REC	F1
Introduction	0.6118	0.5476	0.5779
Related work	0.6364	0.6665	0.6511
Method	0.6579	0.6962	0.6765
Experiment and result	0.6793	0.6833	0.6813
Discussion and conclusion	0.5998	0.5960	0.5979
AVG	0.6370	0.6379	0.6369

Academic search

The structured full-text academic data provide more opportunities for the search engines to improve the search efficiency with the internal structure. Words appeared in sections with different functions may play distinct roles in expressing topics of the paper. Our proposed algorithm, therefore, unifies the internal structure of academic documents which could be treated as structured documents. The ranking results could be modified by weighing the sections with different functions.

Model

We assume that terms in sections with different functions should be weighted respectively. Once we identify the functional structure of scientific documents, the academic search could be regarded as a problem of structured document retrieval.

The notion used in the following section will be illustrated first. Assume that a query $Q = (q_1, q_2, \dots, q_m)$ composed of m words, q_i is the i th word in query Q . Functional structure $S(F_1, F_2, \dots, F_n)$ has n unique structural functions, and the corresponding weight of the structural functions is $W(w_1, w_2, \dots, w_n)$.

The field weighted language model (Kim et al. 2009) is most commonly used in structured document retrieval. The basic language model (Ponte and Croft 1998) uses probability $P(Q|D)$ to calculate the relevance between document D and query W as the following equation:

$$P(Q|D) = \prod_{i=1}^m P(q_i|D)$$

After the identification of functional structure, a scientific document D is a structured document composed of one or more structural function fields $D = (S_1, S_2, \dots, S_n)$. The probability mapping between structural function fields and query could be represented by the following equations:

$$P(S_j|q_i, C) = \frac{P(q_i|S_j, C)P(S_j|C)}{P(q_i|C)}$$

$$P(q_i|C) = \sum_{S_k \in S} P(q_i|S_k, C)P(S_k|C)$$

$$P(S_j|q_i, C) = \frac{P(q_i|S_j, C)P(S_j|C)}{\sum_{S_k \in S} P(q_i|S_k, C)P(S_k|C)}$$

The probability mapping $P(S_j|q_i, C)$ between query word q_i and structural function S_j is the ratio of the probability that the query word q_i occurs in structural function field S_j , and $P(q_i|C)$ is the sum of the probabilities that query word q_i occurs in all structural function fields. $P(S_k|C)$ is the prior probability of the structural function field S_k . The field weighted language model is as follows:

$$P(Q|D) = \sum_{i=1}^m \prod_{j=1}^n P(S_j|q_i, C)P(q_i|S_j, D)$$

$P(q_i|S_j, D)$ is the language model on each structural function field. The smoothing method used in this paper is Dirichlet (Zhai and Lafferty 2001). $P(S_j|q_i, C)$ is the probability mapping between query and the structural function fields, i.e., it is the field weight of the query words. In this paper, we assume that all query words share the same field weights. So, the language model could be transformed to:

$$P(Q|D) = \sum_{i=1}^m \prod_{j=1}^n w_j P(q_i|S_j, D)$$

The document score is the sum of the probabilities that every query word q_i in query Q occurs in all structural function fields. The sum of weight equals 1.

$$\sum_{j=1}^n w_j = 1 (w_j \in (0, 1))$$

The parameter in the language model could be trained by some Heuristic methods. We use GridSearch to estimate the parameter w_j : values range over [0, 1] with an interval 0.05, and ensure that the sum of the parameters is 1. Five-fold experiments are implemented on every set of parameters, and the optimal set of parameters is obtained.

Preliminary results

The data that we used are *inex-1.4*, which is from the XML Retrieval Conference in 2004. It contains 12,107 scientific documents from IEEE Computer Society publications ranging from 1995 to 2002. We only use the 40 CO (content only) query in our experiments. In addition, we only utilize the main content of science documents since our purpose is to test whether our functional structure is useful in academic search.

We set up two experiments for comparison: (1) the basic language model with the Dirichlet smoothing method (LM); and (2) the structural function fields weighted language model (SFFW).

Four metrics MAP, nDCG, P@5, and P@10 are used to compare the results shown in Table 9. It can be seen that all metrics are improved, among which P@5 exhibits a 13.93% relative improvement. This result demonstrates that the functional structure is useful in academic search.

Keyword extraction

Motivation and method

Keywords highly condense the content of scientific documents and could improve the efficiency of information retrieval, document classification, clustering, etc. It is thus a very important technique, especially in the current environment of scholarly big data. However,

Table 9 Results of academic search

	MAP	nDCG	P5	P10
LM	0.2838	0.5726	0.4389	0.4111
SFFW	0.2980	0.5966	0.5000	0.4345
Improvement (%)	5.02	4.21	13.93	5.68

only part of scientific documents has keywords, of which most are labeled by authors. At the same time, the results of manually annotated keywords are strongly subjective, because authors have different understandings of words and content. Consequently, it is almost impossible to manually perform keyword labeling under a unified standard. Therefore, it is critical to extract keywords automatically, which is a topic that has attracted a great amount of attention. For example, Witten et al. (1999) proposed the classic KEA model. Mihalcea and Tarau (2004) put forward the TextRank algorithm, while Beliga et al. (2016) proposed the network based keyword extraction algorithm SBKE. However, few studies have involved the position information of a word in the text. Witten asserts that keywords tend to appear more in the front position of scientific research papers, such as abstract and introduction, because content in the front of research papers constitutes a summary of the overall paper. In addition, there is no systematic study on the influence of functional structure in keyword extraction. In this part, we will explore the role of functional structure in keyword extraction.

The process of keyword extraction is mainly divided into the following steps: (1) candidate generation; (2) feature selection; and (3) model training.

Candidate generation Previous studies mainly focus on methods of generating candidate words to be more accurate. Certain methods, such as N-gram are used in these studies. Moreover, the keywords utilized by prior authors in a specific domain will improve the accuracy of keyword extraction ((Frank et al. 1999). In this paper, keywords used in prior papers with a document frequency larger than 1 will be treated as candidates.

Feature extraction The classic keyword extraction algorithm KEA proposed two features: (1) TFIDF; and (2) the first location at which the words occurred. Besides these two basic features, we propose two other kinds of features based on functional structure: (1) the TFIDF in each structural function field; and (2) the first structural function field in which the words occurred.

Model training There are two main ideas to train keyword extraction models. The first is to regard it as a binary classification problem of candidates (Turney 2000), and the second is to treat it as a ranking problem of the candidate list (Mihalcea and Tarau 2004). In this paper, LibSVM is used to classify whether a candidate is a keyword (Chang and Lin 2011). Meanwhile, the ranking algorithm LambdaMART in RankLib is utilized to rank candidates, and the top N of the ranked candidates list is the final keywords.

Preliminary results

To obtain the keyword candidate set, we first extract all author keywords in dataset CS, and preprocessing, such as word stemming and frequency statistics, are implemented. All keywords with a frequency larger than 1 are then added to the candidate set. The size of the candidate set is 74,723. Then, we randomly select 4000 papers as the experimental data. 3000 of them were randomly selected as the training document set, and the remaining 1000 papers were selected as the testing literature. The author keywords are used as the golden result. Precision is used in binary classification, and three metrics, MAP, P@5, and NDCG@5, are used to evaluate the result of learning to rank.

We set up two experiments for each algorithm based on the two feature sets: (1) the basic features used in KEA; and (2) our feature. The results are listed in Tables 10 and 11.

As shown in Table 10, the precision of our features, which include the functional structure feature, obtained a 10.75% relative improvement compared to the basic feature.

Table 10 Result of binary classification

Metrics	Basic	Ours	Improvement (%)
Precision	0.4867	0.5390	10.75

Table 11 Result of learning to rank

Metrics	Basic	Ours	Improvement (%)
MAP	0.3394	0.352	3.71
P5	0.1832	0.1925	5.08
NDCG@5	0.3539	0.3713	4.92

We also achieved an improvement on all three metrics, as can be seen in Table 11. These results demonstrate that the functional structure is useful in keyword extraction.

Discussions and conclusion

In this paper, we propose a novel clustering algorithm to generate a domain-specific functional structure, in which the identification of functional structure is detailed into three levels—section header, section content, and paragraph—based on the characteristics of three types of scientific documents. Section header-based identification is the most direct and simple method, and suitable for identifying the structural function of strong-structure documents. However, its accuracy would be greatly reduced when the sections in scientific documents are not orderly arranged and many unknown words appear in section headers. Section content-based identification is more general and its performance is more stable, and suitable for strong-structure documents and weak-structure documents. Paragraph-based identification is useful in the identification of no-structure scientific documents, such as scientific reports, without section structures. Applications in two tasks confirm that the identified structure obtains more relevant information and achieves better performance.

Our results indicate that it would be substantially advantageous for scientific documents of a specific domain to be transformed into a unified structure with three-level functional structure identification methods. Besides academic search and keyword extraction, this algorithm could be widely applied, for instance, citation recommendation (He et al. 2010), which aims to help researchers find the best suitable references for his/her paper. One of the main principles of citation recommendation is to calculate the similarity between the entities (e.g., abstract, title, or full text) in candidate recommended references (CRRs) and those in the given paper; the top N similar CRRs are chosen as the outputs of recommendation. If we apply our proposed algorithm into citation recommendations, the CRRs for different sections, such as “introduction” and “methodology”, of the target paper could be different. Distinct weight values for various sections will be considered.

By importing the structure function proposed in this article, future researchers are able to conduct analyses of which section(s) scientific papers are cited in their citing articles and of how the positions of citing sentences change over time. We believe that such analysis will promote the understanding of the purpose of papers being cited, as different types of papers, such as methodology-oriented papers (such as articles introducing a specific

approach), cased study papers, and survey papers, might have particular patterns over their citation processes.

Another potential application of our proposed algorithm lies in scientific article readings in mobiles. Smart phone and social media, as we know, have long been regarded important channels for scientists to share their research and projects (Sugimoto et al. 2017), and have attracted much attention on scientific article readings in mobiles with purposes of providing more pleasant reading experiences. Users might have different preferences when they are reading papers, perhaps due to distinct purposes—for an example, some users like to read introductions and results but skip the literature review section, but for some users they would like to view the related works in a paper since they are exploring the given discipline. Even for one specific user, he/she might have various reading preferences when read different types of papers or under distinct circumstances. By digging the reading intention of a researcher, we could develop a new scientific paper reading system by re-rendering the papers with a personalized section sequence with specific functions.

Scientometrically, location-based co-citation analysis could also be done based on our proposed algorithm. Traditional co-citation analysis (White and Griffith 1981; McCain 1991) and other related improvements (Zhao 2006; Bu et al. 2016; Jeong et al. 2014) simply involved information from bibliographic data or failed to consider the citation location information. The algorithm detailed in this paper provides the potential of involving citation location information into co-citation analysis so that the performance of knowledge domain mappings could be better and more accurate; specifically, this algorithm should be used into the step of “construction of co-citation matrix” in co-citation analysis. Other potential applications include detecting dynamics of citation position of given papers, improving the user experiences in systems of paper reading based on their habits on different sections of paper reading, as well as using location-based co-occurrence analysis to implement bibliometric analyses in future work.

Acknowledgements This study was supported by the Natural Science Funding in China (No. 71473183). The authors would like to thank Ying Ding and the anonymous reviewer for their insightful suggestions.

References

- Beliga, S., Meštrović, A., & Martinčić-Ipšić, S. (2016). Selectivity-based keyword extraction method. *International Journal on Semantic Web and Information Systems*, 12(3), 1–26.
- Bu, Y., Liu, T., & Huang, W.-B. (2016). MACA: A modified author co-citation analysis method combined with general metadata of citations. *Scientometrics*, 108(1), 143–166.
- Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3), 27.
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297.
- Councill, I. G., Giles, C. L., & Kan, M.-Y. (2008). ParsCit: An open-source CRF reference string parsing package. In *Proceedings of the international conference on language resources and evaluation* (pp. 661–667). May 28–30, 2008, Marrakech, Morocco.
- Day, R. A. (1989). The origins of the scientific paper: The IMRAD format. *Journal of the American Medical Writers Association*, 4(2), 16–18.
- De Sordi, J. O., de Paulo, W. L., Meireles, M. A., de Azevedo, M. C., & Pinochet, L. H. C. (2017). Proposal of indicators for the structural analysis of scientific articles. *Journal of Informetrics*, 11(2), 483–497.
- Ding, Y., Chowdhury, G. G., & Foo, S. (2001). Bibliometric cartography of information retrieval research by using co-word analysis. *Information Processing and Management*, 37(6), 817–842.
- Ding, Y., Liu, X., Guo, C., & Cronin, B. (2013). The distribution of references across texts: Some implications for citation analysis. *Journal of Informetrics*, 7(3), 583–592.

- Fader, A., Soderland, S., & Etzioni, O. (2011). Identifying relations for open information extraction. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 1535–1545). July 27–29, 2011, Edinburgh, UK.
- Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., & Lin, C.-J. (2008). LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9(8), 1871–1874.
- Frank, E., Paynter, G. W., Witten, I. H., Gutwin, C., & Nevill-Manning, C. G. (1999). Domain-specific keyphrase extraction. In *16th International joint conference on artificial intelligence (IJCAI 99)* (Vol. 2, pp. 668–673). San Francisco, CA: Morgan Kaufmann Publishers Inc.
- He, Q., Pei, J., Kifer, D., Mitra, P., & Giles, L. (2010). Context-aware citation recommendation. In *Proceedings of the 19th international conference on World Wide Web* (pp. 421–430). ACM.
- Hu, Z., Chen, C., & Liu, Z. (2013). Where are citations located in the body of scientific articles? A study of the distributions of citation locations. *Journal of Informetrics*, 7(4), 887–896.
- Jeong, Y.-K., Song, M., & Ding, Y. (2014). Content-based author co-citation analysis. *Journal of Informetrics*, 8(1), 197–211.
- Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *European conference on Machine Learning: ECML* (pp. 137–142). Berlin: Springer.
- Kim, J., Xue, X., & Croft, W. B. (2009). A probabilistic retrieval model for semistructured data. In *European conference on information retrieval* (pp. 228–239). Springer. Retrieved from http://link.springer.com/chapter/10.1007/978-3-642-00958-7_22.
- Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning* (Vol. 1, pp. 282–289). Retrieved from <http://www.jmlr.org/papers/volume15/doppa14a/source/biblio.bib.old>.
- Leydesdorff, L. (2001). *The challenge of scientometrics: The development, measurement, and self-organization of scientific communications*. Universal-Publishers. Retrieved from <https://books.google.com/books?hl=zh-CN&lr=&id=H7J6Q-IQ5GcC&oi=fnd&pg=PA1&dq=The+challenge+of+scientometrics:+The+development,+measurement,+and+self-organization+of+scientific+communication&ots=OQLb4jF3IH&sig=KJRGq6S2F7lwm9xgRYzcJUUE58>.
- Li, L.-J., & Ge, G.-C. (2009). Genre analysis: Structural and linguistic evolution of the English-medium medical research article (1985–2004). *English for Specific Purposes*, 28(2), 93–104.
- Luong, M.-T., Nguyen, T. D., & Kan, M.-Y. (2012). Logical structure recovery in scholarly articles with rich document features. *Multimedia Storage and Retrieval Innovations for Digital Library Systems*, 270.
- Martin, P. M. (2003). A genre analysis of English and Spanish research paper abstracts in experimental social sciences. *English for Specific Purposes*, 22(1), 25–43.
- McCain, K. W. (1991). Mapping economics through the journal literature: An experiment in journal co-citation analysis. *Journal of the American Society for Information Science*, 42(4), 290.
- Mihalcea, R., & Tarau, P. (2004). *TextRank: Bringing order into texts*. Association for Computational Linguistics.
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. [arXiv:1301.3781](https://arxiv.org/abs/1301.3781). Retrieved from <http://arxiv.org/abs/1301.3781>.
- Nguyen, T. D., & Kan, M.-Y. (2007). Keyphrase extraction in scientific publications. In *International conference on Asian digital libraries* (pp. 317–326). Springer. Retrieved from http://link.springer.com/10.1007%2F978-3-540-77094-7_41.
- Nwogu, K. N. (1997). The medical research paper: Structure and functions. *English for Specific Purposes*, 16(2), 119–138.
- Ponte, J. M., & Croft, W. B. (1998). A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on research and development in information retrieval* (pp. 275–281). ACM. Retrieved from <http://dl.acm.org/citation.cfm?id=291008>.
- Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3), 130–137.
- Sollaci, L. B., & Pereira, M. G. (2004). The introduction, methods, results, and discussion (IMRAD) structure: A 50-year survey. *Journal of the Medical Library Association*, 92(3), 364–371.
- Sugimoto, C. R., Work, S., Larivière, V., & Haustein, S. (2017). Scholarly use of social media and altmetrics: A review of the literature. *Journal of the Association for Information Science and Technology*, 68(9), 2037–2062.
- Sutton, C., McCallum, A., et al. (2012). An introduction to conditional random fields. *Foundations and Trends in Machine Learning*, 4(4), 267–373.
- Swales, J. (1990). *Genre analysis: English in academic and research settings*. Cambridge University Press. Retrieved from https://books.google.com/books?hl=zh-CN&lr=&id=shX_EV1r3-0C&oi=fnd&pg=

PR7&dq=Genre+analysis:+English+in+academic+and+research+setting.&ots=8FW0t-irxf&sig=U_dDsXBwVdpB1VIQMAx6UZZDX8U.

- Turney, P. D. (2000). Learning algorithms for key phrase extraction. *Information Retrieval*, 2(4), 303–336.
- Wang, X., Cheng, Q., & Lu, W. (2014). Analyzing evolution of research topics with NEViewer: A new method based on dynamic co-word networks. *Scientometrics*, 101(2), 1253–1271.
- White, H. D., & Griffith, B. C. (1981). Author cocitation: A literature measure of intellectual structure. *Journal of the Association for Information Science and Technology*, 32(3), 163–171.
- Witten, I. H., Paynter, G. W., Frank, E., Gutwin, C., & Nevill-Manning, C. G. (1999). KEA: Practical automatic keyphrase extraction. In *Proceedings of the fourth ACM conference on digital libraries* (pp. 254–255). ACM. Retrieved from <http://dl.acm.org/citation.cfm?id=313437>.
- Yang, Y., & Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In *Icml* (Vol. 97, pp. 412–420). Retrieved from <http://www.surdeanu.info/mihai/teaching/ista555-spring15/readings/yang97comparative.pdf>.
- Zhai, C., & Lafferty, J. (2001). A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 334–342). ACM. Retrieved from <http://dl.acm.org/citation.cfm?id=384019>.
- Zhang, L. (2012). Grasping the structure of journal articles: Utilizing the functions of information units. *Journal of the American Society for Information Science and Technology*, 63(3), 469–480.
- Zhao, D. (2006). Towards all-author co-citation analysis. *Information Processing and Management*, 42, 1578–1591.