Exploring Technical Phrase Frames from Research Paper Titles

Yuzana Win
Graduate School of Engineering
Nagasaki University
Nagasaki, Japan
bb52214201@cc.nagasaki-u.ac.jp

Tomonari Masada
Graduate School of Engineering
Nagasaki University
Nagasaki, Japan
masada@nagasaki-u.ac.jp

***Abstract*—This paper proposes a method for exploring technical phrase frames by extracting word *n*-grams that match our information needs and interests from research paper titles. Technical phrase frames, the outcome of our method, are phrases with wildcards that may be substituted for any technical term. Our method, first of all, extracts word trigrams from research paper titles and constructs a co-occurrence graph of the trigrams. Even by simply applying PageRank algorithm to the co-occurrence graph, we obtain the trigrams that can be regarded as technical keyphrases at the higher ranks in terms of PageRank score. In contrast, our method assigns weights to the edges of the co-occurrence graph based on Jaccard similarity between trigrams and then apply weighted PageRank algorithm. Consequently, we obtain widely different but more interesting results. While the top-ranked trigrams obtained by unweighted PageRank have just a self-contained meaning, those obtained by our method are technical phrase frames, i.e., a word sequence that forms a complete technical phrase only after putting a technical word (or words) before or/and after it. We claim that our method is a useful tool for discovering important phraseological patterns, which can expand query keywords for improving information retrieval performance and can also work as candidate phrasings in technical writing to make our research papers attractive.**

Keywords—phrase frames; word n-grams; Jaccard similarity; PageRank; keyphrase extraction

I. INTRODUCTION

Nowadays, it is a worthwhile task to extract valuable insights from a large scale dataset. However, the massive scale of the dataset prevents users from accessing the whole dataset efficiently and thus from obtaining information relevant to their need, because search results based on query words may not display the documents that do not contain the inputted query words. Therefore, users are required to have knowledge and skill in finding appropriate query words or phrases for search. Especially in research paper retrieval, which is the application we mainly consider here, it can be considerably difficult for users to find appropriate query words or phrases, because users may be unfamiliar with target research topics. Therefore, the important issue we need to address is to provide users with help in finding appropriate query words or phrases. In this paper, we propose a method for extracting important phrases as word *n*-grams from research papers so that the extracted phrases provide concise and precise summarizations of the target research topics and thus are useful in research information search. However, our method has an outstanding feature. That is, our method can provide *phrase frames* [1], not phrases in the ordinary sense, as its outcome.

Phrase frames are phrases with wildcards that may be substituted for any word. By substituting the wildcards of a phrase frame with words, we can obtain a complete phrase. Important phrase frames can be found based on their frequency from an arbitrary corpus [2]. However, our problem is to find *technically important* phrase frames. Therefore, we propose a method that addresses the problem as follows.

Firstly, we extract trigrams from a large set of research paper titles. We focus on trigrams, because longer *n*-grams make us suffer from data sparseness problem, and unigrams and bigrams are too short to compose useful technical phrase frames. The usage of paper titles can be justified by the fact that the paper title works as a good description of the paper. *Secondly*, we construct a co-occurrence graph of the extracted trigrams, where trigram pairs co-occurring in the same paper title are connected by edges. Here we may simply apply PageRank [4] algorithm to the co-occurrence graph. Then we can obtain the top-ranked trigrams in terms of PageRank score and regard them as technically important. However, our method introduces an additional step. That is, *thirdly*, we compute Jaccard similarity for every co-occurring pair of trigrams and use the similarity as the weight of the corresponding graph edge. And *fourthly*, we apply a weighted version of PageRank to the modified co-occurrence graph and obtain a list of top-ranked trigrams. Our method consists of the above four steps. Details of each step will be explained later.

The top-ranked trigrams obtained by weighted PageRank have a special feature when compared with those obtained by unweighted PageRank. The former trigrams are more *phraseological* than the latter ones. For example, when we use a set of the titles of the paper presented at NLP conferences, unweighted PageRank gives higher ranks to trigrams like “Word Sense Disambiguation” and “Statistical Machine Translation”. It can be said that unweighted PageRank tends to provide trigrams that have a self-contained meaning. In contrast, weighted PageRank gives higher ranks to trigrams like “of Linguistic and” and “for Statistical Machine”. We can expect that a widely different query expansion [3] will be achieved by using such

trigrams. For example, the trigram "for Statistical Machine" extends the query words like "Model" and "Adaptation" to the right and also extends the query words like "Learning" and "Translation" to the left. In short, our method explores trigrams that can provide more flexible expansions of queries. Further, our method may be used as a tool to recommend candidate phrasings for making our research papers attractive, because our methods tend to give phraseological trigrams. We will present more examples later.

The remainder of this paper is organized as follows. Section 2 describes the previous work. Section 3 explains the proposed method. Section 4 presents the details of the dataset used in our experiment. Section 5 contains the results of the experiment. Section 6 concludes the paper with discussion on future work.

II. Previous Work

PageRank algorithm is commonly used in social network analysis, information retrieval, keyphrase extraction, etc. We describe the previous work relating to keyphrase extraction, because it is relevant to our problem. There are two types of keyphrase extraction, i.e., supervised and unsupervised methods.

Hasan et al. [7] analyzed which type of unsupervised approach shows the best performance for keyphrase extraction. They used Tf-Idf, TextRank [9], SingleRank [11], ExpandRank [11] and Clustering-based approach [12] to gain a better performing keyphrase extraction. KEA [8] extracted keyphrases by using Tf-Idf as features of naïve Bayes classification. This method is a supervised one. According to the experimental results given in these two contributions, it is indicated that Tf-Idf is useful for keyphrase extraction. In contrast, our method focuses on co-occurrence frequency based similarity between phrases, not on weighting of each phrase.

Mihalcea et al. [9] first proposed TextRank, a graph-based ranking model for text processing including two innovative unsupervised methods of keyword and sentence extraction. They used a graph to represent the co-occurrence in a window of a fixed number of words, ranked the words in a weighted PageRank manner, and identified important connection between adjacent words in a text.

Mihalcea et al. [10] proposed a new approach for unsupervised knowledge-based word sense disambiguation to extract graph structures from documents. The researchers combined information drawn from a semantic network (WordNet) with graph-based ranking algorithms (PageRank) to implement summarization and word sense disambiguation. They built WordNet-based concepts graph to find the meaning of words and identified those synsets of highest importance.

Gollapalli et al. [13] proposed a method to extract keyphrases from research papers using citation networks. They used the CiteTextRank method of keyphrase extraction from research papers that are embedded in citation networks. They also used PageRank algorithm on word graphs constructed from target papers and their local neighborhood in a citation network, where multiple neighborhoods accompanied with different weights are incorporated.

Our method also applies graph-based ranking algorithms. However, it is not similar to the above three contributions. Firstly, we focus on trigrams and extract trigrams from a large set of research paper titles. Secondly, we construct a co-occurrence graph of the extracted trigrams and use Jaccard similarity as the weight of the corresponding graph edge to apply a weighted version of PageRank. Our combination of Jaccard similarity with trigrams is not considered by any of the PageRank-type methods described above.

III. Method

A. Word Trigrams

Our method, first of all, extracts trigrams from a large set of research paper titles as shown in Table I. We focus on trigrams, because longer *n*-grams make us suffer from data sparseness problem. Especially, it may be difficult to obtain dense co-occurrence data for longer *n*-grams. On the other hand, unigrams and bigrams are too short to compose useful technical phrase frames. We expect that the extracted *n*-grams will contain at least one technical word, because we consider query expansion in research information retrieval as an application of our method. Therefore, shorter *n*-grams are also not appropriate. The usage of paper titles can be justified by the fact that the paper title works as a good concise description of the content of the paper. We use the natural language toolkit for Python (NLTK) to extract trigrams.

B. Co-occurrence Graph

Secondly, we construct a co-occurrence graph of the extracted trigrams. To construct the co-occurrence graph, extracted word trigrams are used as nodes, and co-occurrence relations of trigrams appearing in the same paper titles are used as edges as shown in Fig. 1, which depicts a small portion of the complete co-occurrence graph. This portion is obtained from the two paper titles given in Table I. While we may apply PageRank algorithm to this co-occurrence graph, our method has an additional step for exploring technical phrase frames.

C. Jaccard Similarity

Thirdly, we compute Jaccard similarity for every co-occurring pair of trigrams and use the similarity as the weight of the corresponding graph edge. Let (t_1, t_2) denote a pair of trigrams whose similarity is to be measured. Let $S(t_i)$ denote the set of the paper titles that contain the trigram t_i. We calculate the Jaccard similarity between t_1 and t_2 as follows:

$$sim(t_1, t_2) = \frac{|S(t_1) \cap S(t_2)|}{|S(t_1) \cup S(t_2)|} \quad (1)$$

As a result, if two trigrams appear in the same set of paper titles then they are deemed identical, and if they have no common set of paper titles then they are regarded as completely different. By using the similarity as the weight of the edges of the co-occurrence graph, we can apply a weighted version of PageRank algorithm. As Choi et al. discussed in their survey paper [14], many binary similarity measures have been proposed. There are two reasons why we choose Jaccard similarity. The one is its simplicity. The other is that Jaccard similarity is defined with no reference to *negative matches*. In our case, negative matches correspond to the paper titles where both trigrams are absent. For example, Russell & Rao similarity is defined with reference to *negative matches*. However, for most pairs of trigrams, the number of negative matches is large and comparable with the total number of paper titles. Therefore, the inclusion of negative matches makes the similarity evaluation less reliable in our case.

TABLE I. EXAMPLES OF PAPER TITLES AND THEIR WORD TRIGRAMS

"Recognition of Linear Context-Free Rewriting Systems." [('Recognit', 'of', 'Linear'), ('of', 'Linear', 'Context-Fre'), ('Linear', 'Context-Fre', 'Rewrit'), ('Context-Fre', 'Rewrit', 'System')]
"Optimal Head-Driven Parsing Complexity for Linear Context-Free Rewriting Systems." [('Optim', 'Head-Driven', 'Pars'), ('Head-Driven', 'Pars', 'Complex'), ('Pars', 'Complex', 'for'), ('Complex', 'for', 'Linear'), ('for', 'Linear', 'Context-Fre'), ('Linear', 'Context-Fre', 'Rewrit'), ('Context-Fre', 'Rewrit', 'System')]

D. PageRank Algorithm

Our method applies PageRank algorithm to the co-occurrence graph of the extracted trigrams. We first explain unweighted PageRank algorithm and then weighted PageRank algorithm. The latter is used in our proposed method. Let $PR(t_j)$ denote the PageRank score, assigned to the edge connecting the two nodes, i.e., two trigrams, t_i and t_j. PageRank algorithm provides the score of the trigram t_i as a stationary probability satisfying the following equation:

$$PR(t_i) = \frac{1-d}{N} + d \times \sum_{t_j \in M(t_i)} \frac{PR(t_j)}{L(t_j)} \quad (2)$$

where $t_1, \ldots, t_N$ are trigrams, $M(t_i)$ is the set of trigrams co-occurring with t_i, $L(t_j)$ is the number of trigrams co-occurring with the trigram t_j, and N is the total number of extracted trigrams. The parameter d is a damping factor which can be set between 0 and 1. In our experiment, we set d to 0.85 [4]. We can obtain the top-ranked trigrams in terms of this PageRank score.

We apply a weighted version of PageRank to the modified co-occurrence graph. Let w_{ji} denote the weight assigned to the edge connecting two nodes, t_i and t_j. We set w_{ji} to be equal to the corresponding Jaccard similarity. Then weighted PageRank algorithm gives the PageRank score of the trigram t_i as follows:

$$PR(t_i) = \frac{1-d}{N} + d \times \sum_{t_j \in M(t_i)} \frac{w_{ji}}{\sum_{t_k \in M(t_j)} w_{jk}} PR(t_j) \quad (3)$$

where $M(t_i)$ denote the neighbors of the node t_i and d is the damping factor, which is set to 0.85 as in case of unweighted PageRank. $\sum_{t_k \in M(t_j)} w_{jk}$ is the sum of the weights assigned to each neighbor node in $M(t_j)$. Basically, if a node has many high-scored neighbors, the node will receive a high score. However, our method combines Jaccard similarity and weighted PageRank algorithm. Therefore, if a node has many high-scored neighbors that are similar to the node in terms of Jaccard similarity, then the node will receive a high score. This special feature makes the top-ranked trigrams given by our method widely different from those obtained by unweighted PageRank.

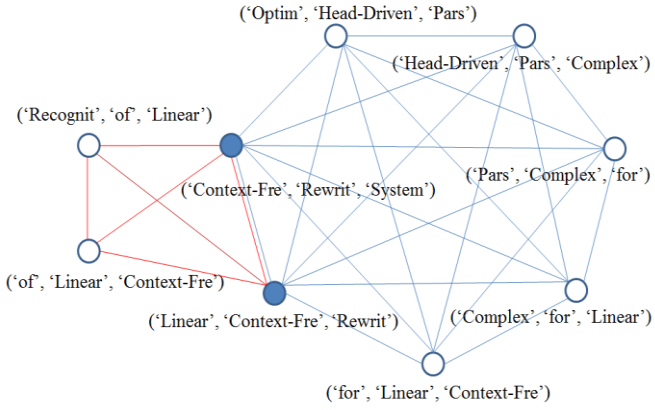

Fig. 1. A small portion of the complete co-occurrence graph

IV. DATA

We tested our method by choosing 37,367 research paper titles from DBLP (Digital Bibliography & Library Project) [5]. Each DBLP record includes a list of authors, title, conference name or journal name, page numbers, etc. The DBLP data consists of over 200,000 records, each stored in an XML file of

size around 1.6GB. We chose academic conferences in the two fields: Natural Language Processing (NLP) and Data Management (DM). We selected top conferences (venue) and only use the research paper titles associated with the selected conferences as shown in Table II, where we also present some examples of paper titles. We removed duplicate research paper titles. As a result, the total number of paper titles contained in NLP and DM datasets are 8,034 and 29,333, respectively. In this paper, we apply Porter Stemmer [6] in our pre-processing to stem words to their root forms. The total number of trigrams extracted from NLP and DM dataset are 43,863 and 163,661, respectively.

TABLE II. DATASETS

Fields	***Examples of Paper Titles***	***Venue***
NLP	"Web Text Corpus for Natural Language Processing" "Nonparametric Word Segmentation for Machine Translation" "A Framework of NLP Based Information Tracking and Related Knowledge Organizing with Topic Maps "	ACL, EACL, COLING, CICLing, NAACL, IJCNLP, EMNLP, NLDB, TSD
DM	"Performance Comparisons of Distributed Deadlock Detection Algorithms" "Using Bayesian Network Learning Algorithm to Discover Causal Relations in Multivariate Time Series" "Efficient Reasoning about Data Trees via Integer Linear Programming"	SIGMOD, VLDB, PODS, SIGIR, WWW, KDD, ICDE, ISWC, CIDR, ICDM, ICDT, EDBT, SDM, CIKM, ER, ICIS, SSTD, WebDB, SSDBM, CAiSE, ECIS, PAKDD

V. RESULTS

We apply weighted PageRank and unweighted PageRank to our datasets. The experimental results show that weighted PageRank can explore trigrams that can be regarded as technical phrase frames. In contrast, unweighted PageRank only explores trigrams as phrases with a self-contained meaning. We clarify the difference of these two types of trigrams by displaying examples.

Firstly, we present examples of similarities calculated between a pair of trigrams to explain what kind of trigram pairs are estimated as similar to each other. For the proposed method, we use Jaccard similarity given in Eq. (1). Tables III and IV show examples of the obtained Jaccard similarities for NLP dataset and DM dataset, respectively. A value of 0 indicates that two trigrams co-occur in no paper titles, whereas a value of 1 indicates that the set of the paper titles where the one trigram appears is completely the same with the set of the paper titles where the other trigram appears. In the bottom cells of Tables III and IV, we give two trigrams co-occurring in no paper titles. We, of course, consider no such pairs.

TABLE III. EXAMPLES OF JACCARD SIMILARITIES FOR NLP

Trigrams (t_1)	***Trigrams (t_2)***	***Jaccard Sim.***
('Web','Search','Engin')	('a','Web','Search')	1.000
('Approach','to','Natur')	('to','Natur','Languag')	0.833
('Statist','Natur','Languag')	('for','Statist','Natur')	0.750
('Evalu', 'of', 'an')	('of','an','Automat')	0.667
('Use','a','Genet')	('a','Genet','Algorithm')	0.500
('System','for','Gener')	('in','a','Multilingu')	0.400
('Word','Segment','and')	('and','Part-of-Speech','Tag')	0.375
('Acquir','Select','Prefer')	('of','Japanes','Text')	0.000

TABLE IV. EXAMPLES OF JACCARD SIMILARITIES FOR DM

Trigrams (t_1)	***Trigrams (t_2)***	***Jaccard Sim.***
('Graph','Partit','and')	('Partit','and','Data')	1.000
('in','the','Presenc')	('the','Presenc','of')	0.973
('World','Wide','Web')	('the','World','Wide')	0.895
('A','Gener','Approach')	('Gener','Approach','to')	0.875
('A','Machin','Learn')	('Machin','Learn','Approach')	0.857
('Languag','Base','on')	('Queri','Languag','Base')	0.750
('SQLServer','2000','and')	('the','Internet','to')	0.667
('Object','Orient','Databas')	('for','Microsoft','SQL')	0.000

TABLE V. PAGERANK SCORES FOR NLP

Unweighted PageRank	***Weighted PageRank***
('Word','Sens','Disambigu') 5.57×10^{-3}	('of','Linguist','and') 3.87×10^{-3}
('Statist','Machin','Translat') 4.71×10^{-3}	('for','Statist','Machin') 3.87×10^{-3}
('Name','Entiti','Recognit') 3.28×10^{-3}	('in','Natur','Languag') 3.13×10^{-3}
('Natur','Languag','Process') 3.05×10^{-3}	('of','Textual','Entail') 3.10×10^{-3}
('Machin','Translat','System') 2.62×10^{-3}	('of','Verb','in') 3.06×10^{-3}
('for','Statist','Machin') 2.49×10^{-3}	('for','Word','Sens') 2.89×10^{-3}
('Spoken','Dialogu','System') 2.39×10^{-3}	('of','Telegraph','Messag') 2.81×10^{-3}
('Condit','Random','Field') 2.33×10^{-3}	('for','Natur','Languag') 2.66×10^{-3}

TABLE VI. PAGERANK SCORES FOR DM

Unweighted PageRank	*Weighted PageRank*
('Design','and','Develop') 9.99×10^{-5}	('Use','and','Perform') 9.99×10^{-5}
('Approach','for','Inform') 9.99×10^{-5}	('Optim','Approach','Integr') 9.99×10^{-5}
('Extend','Entity-Relationship','Data') 9.99×10^{-5}	('Theoret','Approach','to') 9.99×10^{-5}
('Storag','Scheme','for') 9.98×10^{-5}	('a','Digit','Librari') 9.99×10^{-5}
('Emerg','Pattern','for') 9.98×10^{-5}	('of','Bibliograph','Retriev') 9.99×10^{-5}
('Avail','in','Partit') 9.98×10^{-5}	('in','Collabor','Filter') 9.99×10^{-5}
('Web','Servic','Composit') 9.98×10^{-5}	('of','the','Use') 9.99×10^{-5}
('Theoret','Framework','and') 9.97×10^{-5}	('in','the','Distribut') 9.99×10^{-5}

Secondly, we provide examples of trigrams having large PageRank scores. For unweighted PageRank algorithm, we use Eq. (2) and calculate stationary probabilities, where we utilize equal weights for all outgoing edges on every node, because we use no trigram similarities. For weighted PageRank algorithm, we use Eq. (3) and calculate stationary probabilities, where we utilize Jaccard similarities as weights for outgoing edges on every node. Table V and VI summarize the results for NLP conference paper titles and DM conference paper titles, respectively. For example, ('Word', 'Sens', 'Disambigu') and 5.57×10^{-3} in the top cell of the left column of Table V mean that the stationary probability is calculated as 5.57×10^{-3} for the trigram ('Word', 'Sens', 'Disambigu') by using no Jaccard similarities. Furthermore, ('of', 'Linguist', 'and') and 3.87×10^{-3} in the top cell of the right column of Table V mean that the stationary probability is calculated as 3.87×10^{-3} for the trigram ('of', 'Linguist', 'and') by using Jaccard similarities. While we calculate stationary probabilities for all possible combinations of trigrams, only the eight trigrams top-ranked in terms of PageRank score are displayed due to space limitation.

According to the result presented in the left columns of Tables V and VI, we can observe that unweighted PageRank gives higher ranks to trigrams like "Word Sense Disambiguation", "Extend Entity-Relationship Data", "Web Service Composition", etc. by recovering the original form from the root form of each word. It should be noted that less functional words like prepositions, articles, conjunctives, etc. are observed. That is, unweighted PageRank tends to provide trigrams that make sense without adding words before or/and after them. In contrast, the result presented in the right columns of Tables V and VI shows that weighted PageRank gives higher ranks to trigrams like "of Linguistic and", "for Statistical Machine", "of the Use", etc. where functional words appear as a grammatical component. Especially, the first word is a preposition in many trigrams, though such cases are rarely observed for unweighted PageRank. It can be said that the top-ranked trigrams given by our method are more *phraseological* than those given by unweighted PageRank. Below we discuss how the phraseological nature of our method may work in query expansion for information retrieval.

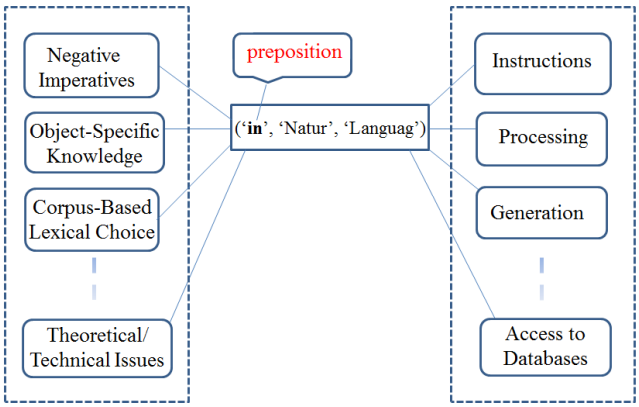

Fig. 2. Example of possible expansions using the technical phrase frames "in Natur Languag" in NLP dataset

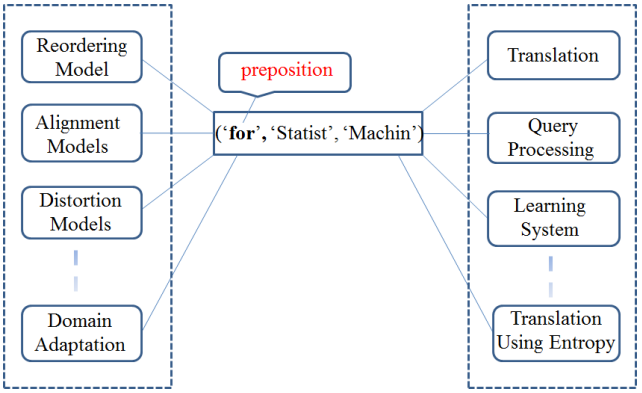

Fig. 3. Example of possible expansions using the technical phrase frames "for Statist Machin" in NLP dataset

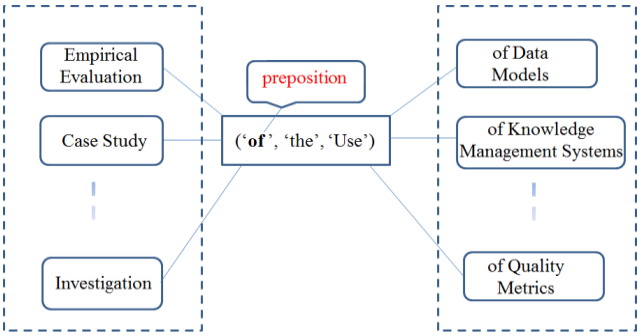

Fig. 4. Example of possible expansions using the technical phrase frames "of the Use" in DM dataset

For example, as we present in Fig. 2, the trigram (‘in’,‘Natur’,‘Languag’), whose original form is “in Natural Language”, can extend to the right the following technical phrases: “Negative Imperatives”, “Object-Specific Knowledge”, “Corpus-Based Lexical Choice”, “Theoretical/Technical Issues”, etc., and to the left the following technical words and phrases: “Instructions”, “Processing”, “Generation”, “Access to Databases”, etc. These examples are actually observable in existing research paper titles. In this manner, our method can be utilized for expanding technical words or phrases, which may be used as a query word or phrase for search system, by a technical phrase frame including functional words.

Fig. 3 gives another example. The trigram “for Statistical Machine” can extend “Reordering Model”, “Alignment Models”, “Distortion Models”, “Domain Adaptation”, etc. to the right and also “Translation”, “Query Processing”, “Learning System”, “Translation Using Entropy”, etc. to the left. These possible expansions are also actually obtained from real research paper titles. In addition, Fig. 4 provides the third example. The trigram “of the Use” may extend “Empirical Evaluation”, “Case Study”, “Investigation”, etc. to the right. However, this trigram may not be used for expanding any words or phrases to the left, because its last word is “Use”, a noun mainly followed by functional words. While we can find the paper titles in which the trigram “of the Use” is followed by the phrases like “of Data Models”, “of Knowledge Management Systems”, “of Quality Metrics”, etc., we may not use such phrases as a query. Therefore, this trigram may be used only for the expansion to the right.

In short, trigrams explored by our method can provide a query expansion more flexible in the sense that the expansion takes into consideration functional words. The concept of *technical phrase frame* implemented by our method will lead to a new query expansion scheme more oriented toward actual user needs and interests. A search system using this type of query expansion or query suggestion may excel as an environment for exploring and studying technical topics and evolving trends in academic data repositories. Further, our method may also be used as a tool for recommending candidate phrasings when we try to make our research papers attractive, because our method is likely to give phraseological trigrams.

VI. Conclusion

In this paper, we extracted trigrams from a large set of research paper titles. And then we constructed a co-occurrence graph of the extracted trigrams and used Jaccard similarity as the weight of the graph edge to apply a weighted version of PageRank. In particular, we observed that weighted PageRank could explore trigrams as technical phrase frames. In contrast, unweighted PageRank only explored trigrams as phrases with a self-contained meaning. Therefore, it can be concluded that our method may achieve query expansion in a more phraseological manner.

We have a plan to evaluate trigrams extracted by our method from a quantitative viewpoint. The extracted trigrams can be used as features of documents along with words. Therefore, for example by using naïve Bayes classifier, we can check in what kind of situation the extracted trigrams improve the classification accuracy. We also have a plan to implement query expansion using the extracted trigrams and evaluate the effectiveness of trigrams by measuring the quality of retrieved results.

Acknowledgment

This work has been supported by the Grant-in-Aid for Scientific Research (C) No.26330256 of the Ministry of Education, Science, Sports and Culture (MEXT) from 2014 to 2017. We are grateful for their support.

References

[1] T. McEnery and A. Hardie. *Corpus Linguistics: Method, Theory and Practice*. Cambridge University Press, 2011.

[2] http://phrasesinenglish.org/explorep.html

[3] C.D. Manning and P. Raghavan. *Introduction to Information Retrieval.* Cambridge University Press, 2008.

[4] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Comput. Netw. ISDN Syst.*, Vol. 30 No. 1-7, pp. 107-117, 1998.

[5] http://www.dblp.com/

[6] http://www.tartarus.org/martin/PorterStemmer/

[7] K.S. Hasan and V. Ng. Conundrums in unsupervised keyphrase extraction: Making Sense of the State-of-the-Art. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pp. 365-373, 2010.

[8] I.H. Witten, G.W. Paynter, E.Frank, C. Gutwin, and C.G. Nevill-Manning. KEA: Practical automatic keyphrase extraction. In *Proceedings of the Fourth ACM Conference on Digital Libraries*, pp. 254-255, 1999.

[9] R. Mihalcea and P. Tarau. TextRank: Bringing order into texts. In *Proc. of the 2004 Conference on Empirical Methods in Natural Language Processing*, pp. 404-411, 2004.

[10] R. Mihalcea, P. Tarau and E. Figa. PageRank on semantic networks,with application to word sense disambiguation. In *Proceedings of the 20th International Conference on Computational Linguistics*, Article No. 1126, 2004.

[11] X. Wan and J. Xiao. Single document keyphrase extraction using neighborhood knowledge. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2*, pp. 855-860, 2008.

[12] F. Liu, D. Pennell, F. Liu and Y. Liu. Unsupervised approaches for automatic keyword extraction using meeting transcripts. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 620-628, 2009.

[13] S.D. Gollapalli and C. Caragea. Extracting keyphrases from research papers using citation networks. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pp. 1629-1635, 2014.

[14] S. Choi, S. Cha, and C.C. Tappert. A survey of binary similarity and distance measures. *J. Syst. Cybern. Inf.*, Vol. 8, No. 1, pp. 43-48, 2010.