Keyphrase extraction. Abstracts instead of full papers

Svetlana Popova
Saint-Petersburg State University and ITMO University
Saint-Petersburg, Russia
svp@list.ru

Vera Danilova
Autonomous University of Barcelona
Barcelona, Spain
maolve@gmail.com

***Abstract*—In the present paper[1] we consider keyphrase extraction problem from scientific articles. Finding an appropriate solution is important for the organization of fast navigation in databases, indexing, clustering and classification of academic papers. The base collection includes keyphrases selected by the experts for each text (SemEval2010). It is shown that the use of abstracts instead of full texts allows to improve the results obtained by processing full texts or abstracts with introduction and conclusion section. Our approach uses the extraction of keyphrases with linguistic patterns (part of speech-based); patterns are built on the basis of an auxiliary dataset. The use of abstracts in this approach allows to reduce the number of words sequences extracted with patterns, as compared to the use of full texts. It allows to simplify or totally omit the ranking stage. Ranking is usually needed, because out of many keyphrases candidates we have to choose only 10-15. This stage is the most difficult and its effectiveness depends on the number of the selected candidates to keyphrases. The use of abstracts makes it possible to considerably reduce the number of candidate phrases and at the same time yields high recall.**

Keywords—keyphrase extraction; keyphrase identification; indexing; abtract processing; informational retrieval

I. Introduction and Related Work

Keyword extraction has a large number of applications that include such prominent tasks as topic extraction, data structuring, summary construction, ontology population, search of related concepts in large data collections and other.

In the field of keyphrase extraction several main approaches can be distinguished. The first one deals with word (words sequences) ranking, selection of top-ranked units and phrase construction [1-3]. The second one is the most widely used and spans keyphrases construction from candidates, candidate ranking and selection of the best keyphrases or classification of candidates [4-11]. Candidate phrase building often uses n-grams, word sequences that satisfy a number of constraints, e.g. exclusion phrases with stop words, limited length, distance from the beginning of text and other.

According to the results reported in [5], the use of the feature indicating phrase position at the beginning of a document works for academic papers and does not lead to better performance in case of book chapters and scientific webpages that do not have an abstract. In our research we show that precisely at the beginning of a scientific publication, in an abstract section, which is not included in a literary text, the major part of keyphrases is gathered. Moreover, since the length of an abstract is limited, key phrase density is high. This fact should also be considered, as it allows to generate less candidate phrases during processing. In [5] it was mentioned that too many candidates negatively influence ranking and one of the most important tasks consists in elaboration of algorithms for the construction of small candidate sets.

In [7] we were challenged by the fact that the use of a large number of statistic features for candidate ranking (including TF-IDF for words in phrases) does not improve the quality of the extracted phrases. It turned out that the considered methods, described in [7], are close to random ranking and differ only as to the precision of one-word phrase ranking. Upon the exclusion of one-word phrases, despite the fact that the ranking is almost random, the evaluation shows good results as compared to the state-of-the-art in the domain. It becomes possible due to the good level of trade-off between precision and recall of the extracted phrases before ranking, achieved by removing one-word phrases. The percentage of keyphrases in one-word phrases tends to be much lower that in phrases of other length. Therefore, phrase length is another significant feature that should be considered. Longer phrases tend to have higher weight values during ranking as compared to short ones. Higher weight values are assigned to long phrases, because these are usually less frequent within a collection. According to our observation in [7], phrase length plays an important role, because there are different proportions between the correctly and incorrectly extracted candidates depending on this parameter.

Another observation that we take into account is related to the use of Part-of-Speech information. In [11], for the purposes of keyphrase extraction, n-grams (unigrams, bigrams and trigrams), NP-chunks and PoS Tag Patterns were applied. PoS Tag Patterns were selected for the final algorithm, because in case of NP-chunks, around 50% of correct phrases were missed. 56 PoS tag patterns were constructed by manual labeling of the training dataset. 51 patterns out of 56 contained one or more nouns. As it turned out, 5 most frequent patterns included adjective-noun combinations. In order to distinguish key phrases from non-key sequences, a classifier was created, which used 4 features to assign weights. One of the features represents a sequence of candidate phrase tags. The results proved that the use of this feature considerably influences the quality of keyphrase extraction for the three methods of candidates construction (n-grams, NP-chunks, PoS Tag Patterns). Thus, taking into account the syntactic information

[1] This work was partially financially supported by the Government of Russian Federation, Grant 074-U01

on PoS tag sequences in candidate phrases is of high importance. The use of PoS Tag Patterns ensures the highest recall value of candidate extraction. However, false positives create too much noise. Therefore, on the one hand, PoS tag information is significant, which justifies the use of patterns, and, on the other hand, it complicates the extraction of key phrases due to the excessive noise produced in the set of candidate sequences. In [11], dataset includes only academic abstracts. Despite the fact that the results obtained for the given collection using linguistic patterns do not outperform the state-of-the-art, they are comparable to those achieved while abstract processing the collection of full papers (SemEval 2010) [8].

The influence of PoS tag data on keyphrase extraction performance was noted in a number of works [1,3], where it was shown that the use of noun-adjective combinations alone positively contributes to the annotation quality. Our observations also prove that if the best patterns are selected out of the most frequent ones ("best" means the percentage of keyphrases is higher than that of non-key phrases), these tend to contain nouns and adjectives.

The aim of the present work is to show that the use of abstracts in the task of keyphrase extraction from academic papers may be more reasonable than the processing of the entire article texts. Practically, when considering abstracts we select the phrases that are located close to the very beginning of the full paper, which means that we take into account the position of the phrase in such a way. Thus, the use linguistic patterns and information on phrase length allows to outperform the state-of-the-art. Notice, the hypothesis that sentences, which are closer to the beginning of text, tend to contain the most informative phrases, is applied in the domain for full texts processing. Therefore, most ranking algorithms contain variables reflecting the distance from the text start point to a specific phrase.

However, the results of pattern-based extraction can be improved, because, according to the observations in the domain of keyphrase extraction from abstracts, pattern application is not the most effective method. Nevertheless, this approach outperforms the other algorithms that process full papers or abstracts with introduction and/or conclusion section. Our approach is based on the extraction of keyphrases with linguistic patterns (Part-of-Speech-based) that were constructed using an additional dataset. The analysis of abstracts instead of full texts allows to reduce the number of pattern-selected word sequences, which makes it possible to simplify or totally omit the ranking stage. Ranking stage is required when we need to select 5-15 keyphrases out of the large number of candidates. This step is considered the most difficult, and its effectiveness depends considerably on the number of candidates. The use of abstracts allows to reduce the amount of such candidates and at the same time it ensures high recall.

To conclude, we should mention the third approach to keyphrase extraction, which is not considered in detail in the present paper. It is based on the construction of a suffix tree or its modifications, and selection of the most frequent sequences [12-14].

Additionally it should be noted that abstracts and not the full papers are worth interest in the context of practical use, e.g. the task of search engine results representation [12-15], particularly in the systems of academic search [15, 16]. As opposed to full papers, abstracts are usually freely available. Earlier it was observed in [17, 18] that processing abstracts may be enough to obtain adequate results in such tasks as, for instance, clustering academic papers.

II. Evaluation and Datasets

A. Evaluation

One of the most commonly used quality evaluation measures in the domain is applied:

$$\text{Precision} = (C \cap G)/G,\ \text{Recall} = (C \cap G)/C, \qquad (1)$$

$$\text{F-score} = (2*\text{Precision}*\text{Recall})/(\text{Precision}+\text{Recall}),$$

where $C \cap G$ is the number of "true positives"– keyphrases that have been extracted properly in all considered documents, G is the number of automatically extracted keyphrases from all documents and C is the number of all keyphrases extracted by the expert (number of the phrase in the gold standard).

B. Test Dataset

In the present experiments we employ the collection that was used to resolve the automatic keyphrase extraction task at the Workshop on Semantic Evaluation 2010 (SemEval-2010) [8]. It represents a set of 244 scientific articles with keyphrase annotations made by authors and readers. In the present work we make use only of annotations introduced by readers. Further on we will refer to these annotations as gold standard, the first best result, which should be achieved by the algorithm. Collections contain documents belonging to 4 categories: C2.4 (Distributed Systems), H3.3 (Information Search and Retrieval), I2.11 (Distributed Artificial Intelligence—Multiagent Systems) and J4 (Social and Behavioral Sciences—Economics). The collection consists of three parts: TRIAL (40 documents), TRAIN (144 documents) and TEST (100 documents). In Table 1, topical distribution of documents is presented [8].

The gold standard of both subcollections, TEST and TRAIN, of the SemEval collection is represented in several ways: 1) by original phrases, 2) by stemmed phrases. Thus, extraction results can be presented in the form of stems that are compared during evaluation to the stems selected by the experts. In the present paper, as within the framework of the SemEval competition, word stems are used as gold standard.

TABLE I. Topical distribution of documents in SemEval 2010

DataSet	Total Docs.	Topic			
		C	***H***	***I***	***J***
TRAIN	144	34	39	35	36
TEST	100	25	25	25	25

In [8] the performance of 19 algorithms is described. It can be noted that some of the most effective algorithms use information about the section and position of a certain phrase, or extract keyphrases only from the title, abstract, introduction and/or conclusion. Algorithms were required to extract 15, 10 or 5 main phrases. Table 2 shows the best algorithm performance (readers' annotations as gold standard) [8].

TABLE II. BEST ALGORITHM PERFORMANCE ON SEMEVAL COLLECTION

Extracted keyphrases (number)	Algorithm	Precision	Recall	F-score
Top-15 keyphrases	HUMB	0.212	0.264	0.235
Top-10 keyphrases	HUMB	0.248	0.206	0.225
Top-5 keyphrases	HUMB	0.304	0.126	0.178

HUMB extraction performance turned out to be the best in all three cases. This algorithm processes the entire text of the article taking into account the section and position of keyphrases.

III. EXPERIMENT DESCRIPTION AND RESULTS

A. Main algorithm

At the first stage, we extract PoS tags that correspond to each phrase in the gold standard set of the TRAIN (SemEval) collection. For each of the extracted patterns we calculate the amount of phrases that satisfy the patterns selected by the expert for the documents of TRAIN collection. Ranking of the obtained patterns shows that the best ones are those, which match the most part of gold standard phrases. Out of the ranked patterns, 52 best patterns are chosen and further applied on the TEST collection of abstracts to extract words sequences. Abstracts are processed one-by-one and if a sequence that does not contain stop words and punctuation is detected, it becomes a candidate phrase for a given text. Obtained result was evaluated (1). Next, the problem of phrases overlapping was solved and all candidates representing certain document are ranked according to the occurrence frequency of a given phrase in all the documents of the TEST collection. The phrases with the highest frequency value are considered the best ones. Upon ranking, 15, 10 and 5 best phrases are selected and labeled as keyphrases for a given text. Therefore, for each text in the TEST collection, phrases are extracted using linguistic patterns and ranking. In order to evaluate the extracted phrases, (1) was applied.

B. Preprocessing stage

The following parameters were used in order to extract abstracts from the full papers. The word “ABSTRACT” indicated the start point and such phrases as “Categories and Subject Descriptors”, “Organization and Design” or “INTRODUCTION” defined the end point. The extracted annotations were labeled using Stanford POS tagger tool [19] and stemmed with Porter stemmer [20]. Punctuation and stop words were left as they were. The following strategy was used to perform pattern-based keyphrase extraction. All the phrases matching the patterns were retrieved. They could not contain punctuation marks (:;,./<>?!@#$%^&*()=+\|") and stop words. We used the standard stop words list with an addition of separate letters (s, d and others).

C. Experiment description and results

We extracted patterns for the gold standard from the TRAIN collection and selected 52 of the most frequent ones (NN_NN, NN, JJ_NN, NN_NN_NN, JJ_NN_NN, VBN_NN, NN_VBN_NN, VBG_NN, VBN_NN_NN, NN_VBG, NN_JJ_NN, NN_IN_NN_NN_NN, NN_NN_NN_NN, NNS_NN_NN, JJ_NN_NN_NN, NN_VBG_NN, JJ_JJ_NN, JJ_VBG_NN, VB_NN_NN, NN_IN_NN, NN_CC_NN, NN_VBN_NN_NN, VB_NN, VBG, JJ_VBG, RB_NN_NN, VBN_NN_NN_NN, NN_VBN_NN, NN_CC_NN_NN, VB_VBN_NN, JJ_VBN_NN, NNS_NN_NN_NN, NN_VBN, JJ_JJ_NN_NN, VBG_NN_NN, NN_TO_NN_NN, NN_VB, NN_NN_NN_NN_NN, NN_NN_VBG, NN_IN_NN_NN_NN_NN_NN, VB_VBN_NN_NN, NN_JJ_NN_VBG, NNS, IN_NN, NNS_NN, JJ_NN_VBG, NN_NN_IN_NN_NN_NN_NN, JJ_VB_NN, NN_NNS_NN, NN_VBN_VBG, JJ_RB_NN, JJ_JJ_VBG, where NN denotes noun, JJ – adjective, VBN – verb, past participle, VBG — verb, gerund or present participle, VB — verb base form, IN — preposition or subordinating conjunction, CC - coordinating conjunction). Next, we retrieved phrases from the collection TEST applying these patterns and evaluated the extraction quality with (1). The first assumption is that the patterns obtained from the TRAIN collection, which are representative for the keyphrases selected by the expert, may be successfully applied to keyword extraction from the similar collection (TEST).

Table 3 presents the evaluation of results quality for the retrieved phrases. The experiment results show that keyphrase extraction using patterns entails low quality.

TABLE III. EVALUATION OF KEY PHRASE EXTRACTION USING PATTERNS BUILT ON THE BASIS OF THE ¨TRAIN¨ GOLD STANDARD

DataSet	Precision	Recall	F-score
TEST	0.11	0.67	0.19

First of all, the poor extraction quality is due to the low precision value (the amount of word groups being extracted is too large). We can conclude that the use of patterns alone does not ensure effectiveness, as even the most representative patterns may have the same structure as common word groups. However, despite of this, the F-score value is comparable with the average results of the other algorithms that took part in the SemEval competition (they had to select 5-15 keyphrases). Recall values we obtained show the potential of achieving good results in keyphrase extraction in case of elaboration of mechanisms for proper filtering of false candidates.

It should be mentioned here that according to the results of the work on keyphrase extraction from abstracts [11], described in detail in the introduction section, the best recall is achieved by using 56 linguistic patterns and has similar order (0.66) in terms of the evaluation referred to in this paper. In [11], as in the present work, a collection of academic abstracts is used as a test dataset. It should be also noted that although the precision value is different in [11], values' order is quite similar: 0.15 vs 0.11 in the present experiments. A conclusion can be made on the presence of certain regularities in abstract processing.

At the next stage we relied on the observation that patterns may overlap, so we introduced the following rule. Let it be the case that two phrases were extracted and one is a part of another. We leave only the one with the highest frequency across TEST collection abstracts. Single-word phrases were discarded as they often represent parts of other phrases, have high frequency and create noise. If we omit overlap by excluding single-word phrases, it practically has no influence on the quality of the extracted keyphrases (Precision is slightly higher, Recall is lower, F-score is the same as shown in Table 4).

If we eliminate the overlap using the information on phrase frequency, it ensures a considerable increase of Precision and F-score (Table 5).

TABLE IV. EVALUATION OF KEYPHRASE EXTRACTION. SINGLE-WORD PATTERNS ARE EXCLUDED.

DataSet	Precision	Recall	F-score
TEST	0.12	0.52	0.19

TABLE V. EVALUATION OF KEYPHRASE EXTRACTION. SINGLE-WORD PATTERNS ARE EXCLUDED. IF ONE PHRASE IS A SUBPHRASE OF ANOTHER, WE LEAVE ONLY THE ONE WITH THE HIGHEST ABSOLUTE FREQUENCY VALUE.

DataSet	Precision	Recall	F-score
TEST	0.24	0.41	0.30

The interpretation of this result is straightforward. Since the shorter phrases are more frequent, than the longer ones, this approach to the overlap problem gives priority to the phrases of length 2. However, we assume that this method is effective only for those collections, where the experts (readers) label a large number of two-word phrases. In order to check it we selected the patterns of length 2 out of all the selected patterns and extracted key phrases from the collection TEST (NN_NN, JJ_NN, VBN_NN, VBG_NN, NN_VBG, VB_NN, JJ_VBG, NN_VBN, NN_VB, IN_NN, NNS_NN). Results are shown in Table 6 and totally correspond to those in Table 5.

TABLE VI. EVALUATION OF KEYPHRASE EXTRACTION USING PATTERNS OF LENGTH 2.

DataSet	Precision	Recall	F-score
TEST	0.24	0.41	0.30

In order to compare the results with other SemEval participants, we added the ranking stage to extract the first 5, 10 and 15 phrases. Phrases were ranged according to the absolute frequency of keyphrase occurrence in the dataset of abstracts. Single-word phrases were excluded. Results are presented in Table 7.

TABLE VII. EVALUATION OF EXTRACTION QUALITY AFTER RANKING. KEYPHRASES OF LENGTH 1 WERE DISCARDED.

Extracted keyphrases (number)	Precision	Recall	F-score
Top-15 keyphrases	0.25	0.30	0.27
Top-10 keyphrases	0.28	0.23	0.26
Top-5 keyphrases	0.34	0.14	0.20

This algorithm improves the results of the state-of-the-art best performing system. As opposed to HUMB, we used only abstracts instead of full papers. In Table 8 we show examples of abstracts, keyphrases annotated by readers and automatically extracted phrases according to the results of ranking for the cases, where 5 and 15 phrases were extracted.

TABLE VIII. EXAMPLES OF TEXTS, KEYPHRASES LABELED BY THE READERS AND AUTOMATICALLY EXTRACTED KEYPHRASES

Text 1	Distributed Task Allocation in Social Networks This paper proposes a new variant of the task allocation problem, where the agents are connected in a social network and tasks arrive at the agents distributed over the network. We show that the complexity of this problem remains NPhard. Moreover, it is not approximable within some factor. We develop an algorithm based on the contract-net protocol. Our algorithm is completely distributed, and it assumes that agents have only local knowledge about tasks and resources. We conduct a set of experiments to evaluate the performance and scalability of the proposed algorithm in terms of solution quality and computation time. Three different types of networks, namely small-world, random and scale-free networks, are used to represent various social relationships among agents in realistic applications. The results demonstrate that our algorithm works well and that it scales well to large-scale applications.
Text 1. Keyphrases labeled by the readers (gold standard)	[social network, multiag system, behavior, strateg agent, social relationship, interact, task alloc, commun messag, util, algorithm, alloc]
Text 1. Top – 15	[agent interact, util alloc, commun messag, strateg agent, social relationship, task alloc, messag behavior, alloc algorithm, multiag system, algorithm commun, system strateg, behavior multiag, social network, alloc util, relationship task]
Text 1. Top – 5	[social network, system strateg, multiag system, agent interact, task alloc]
Text 2	An Adversarial Environment Model for Bounded Rational Multiagent environments are often not cooperative nor collaborative; in many cases, agents have conflicting interests, leading to adversarial interactions. This paper presents a formal Adversarial Environment model for bounded rational agents operating in a zero-sum environment. In such environments, attempts to use classical utility-based search methods can raise a variety of difficulties (e.g., implicitly modeling the opponent as an omniscient utility maximizer, rather than leveraging a more nuanced, explicit opponent model). We define an Adversarial Environment by describing the mental states of an agent in such an environment. We then present behavioral axioms that are intended to serve as design principles for building such adversarial agents. We explore the application of our approach by analyzing log files of completed Connect-Four games, and present an empirical analysis of the axioms" appropriateness.
Text 2. Keyphrases labeled by the readers (gold standard)	[eval valu, axiomat model, benefici action, adversari interact, behavior axiom, evalu function, empir studi, adversari environ, multiag environ, zero-sum encount, treatment group, connect-four game, bilater and multilater instanti, interact]
Text 2. Top – 15	[studi axiomat, game empir, eval valu, axiomat model, model zero, benefici action, evalu function, axiom bilater, encount treatment, action connect, zero sum, treatment group, valu interact, environ behavior, sum encount]
Text 2. Top – 5	[game empir, axiomat model, benefici action, evalu function, axiom bilater]

We assume that this improvement of the best result is due to minimization of the ranking stage: instead of processing full papers or both abstract, introduction and conclusion we confined ourselves to the use of abstract.

IV. Future Work

A considerable improvement of keyphrase extraction from abstracts is possible by using an extended list of stop words [21]. However, the method of its generation remains an open issue. We implemented the following experiments in this respect. For the collection 'TEST' we selected 700 words that were the most frequent in the incorrectly selected phrases, and for each of these words we checked whether it is possible to achieve an improvement by adding this word to the stop words list. If such an addition leads to an improvement of 0.00005 in terms of F-score, the word was included. This approach to stop words list generation ensures the performance shown in Table 9 that represents a considerable improvement of extraction quality as compared to other algorithms. Unfortunately, in case of SemEval collection it is not possible to generate this list on the basis of the TRAIN collection. Moreover, stop words list obtained on 'TRAIN collection using the above described approach does not enhance the extraction quality for the collection TEST. Such a list cannot be constructed by retrieving words, which are the most frequent within the incorrectly selected phrases of the collection 'TRAIN, including the case, when we discard the words that are the most frequent in the correctly selected phrases. At the next stage we plan to focus on the methods of stop words list generation for the task of keyphrase extraction by using the annotated collections, as well as external corpora.

TABLE IX. Evaluation of extraction quality after ranking. Keyphrases of length 1 were discarded. Extended list of stop words was used

Extracted keyphrases (number)	Precision	Recall	F-score
No ranking	0.38	0.39	0.38
Top-15 keyphrases	0.34	0.28	0.31
Top-10 keyphrases	0.32	0.23	0.27
Top-5 keyphrases	0.40	0.16	0.23

V. Conclusion

The obtained results show high capabilities of using abstracts instead of full papers in the task of keyphrase extraction as compared to the analysis of the entire articles, as well as of abstracts combined with other sections (introduction, conclusion). First of all, abstracts use allows to reduce the number of the selected candidates, which positively influences the ranking stage. For instance, in case of the simple ranking method based on measuring the absolute frequency of keyphrase occurrences in abstract collection and pattern extraction approach, we improved the state-of-the-art results. Pattern-based keyphrase retrieval gives a Recall of 0.67, which promises good results in case of proper classification/ranking of the obtained phrases in order to identify the correct and wrong ones. We assume that the number of pattern-extracted candidates can be reduced by applying an extended stop words list, and this is an important issue to explore. Also, the results show the significance of taking into account the phrase length.

References

[1] R. Mihalcea and P. Tarau, “TextRank: Bringing order into texts,” in: Conference on Empiri-cal Methods in Natural Language Processing, 2004, pp. 404–411.

[2] W. Xiaojun and J. Xiao, “Exploiting Neighborhood Knowledge for Single Document Summarization and Keyphrase Extraction,” ACM Transactions on Information Systems, 28(2), Article 8, 2010.

[3] T. Zesch and I. Gurevych, “Approximate Matching for Evaluating Keyphrase Extraction,” in: International Conference RANLP 2009, pp. 484–489, Borovets, Bulgaria.

[4] N. Pudota, A. Dattolo, A. Baruzzo, F. Ferrara and C. Tasso, “Automatic keyphrase extraction and ontology mining for content-based tag recommendation,” International Journal of Intelligent Systems, vol 25, 2010, pp. 1158-1186.

[5] W. You, D. Fontaine and J.-P. Barhes, “An automatic keyphrase extraction system for scientific documents,” In: Knowl Inf Syst 34, 2013, pp. 691-724.

[6] S. R. El-Beltagy and A. Rafea, “KP-Miner: A keyphrase extraction system for english and arabic documents,” in: Information Systems, 34, 2009, pp. 132-144.

[7] S. Popova and I. Khodyrev, “Ranking in keyphrase extraction problem: is it useful to use statistics of words occurrences?” RuSSIR 2013, Proceedings of Kazan University journal, 2013.

[8] S. N. Kim, O. Medelyan and M. Yen, “Automatic keyphrase extraction from scientific articles,” Language Resources and Evaluation, Springer Kan & Timothy Baldwin, 2012

[9] E. Frank, G. W. Paynter, I. H. Witten, C. Gutwin and C. G. Nevill-Manning, “Domain-specific keyphrase extraction,” in: Proc. of IJCAI, 1999, pp. 688–673.

[10] P. Turney, “Learning to Extract Keyphrases from Text,” in: NRC/ERB-1057,1999, pp. 17– 43.

[11] A. Hulth, “Improved automatic keyword extraction given more linguistic knowledge,” in: Conference on Empirical Methods in Natural Language Processing, pp. 216–223, 2003

[12] A. Bernardini and C. Carpineto, “Full-Subtopic Retrieval with Keyphrase-Based Search Results Clustering,” Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT '09. IEEE/WIC/ACM International Joint Conferences on (Volume:1), 2009.

[13] D. Zhang and Y. Dong, “Semantic, Hierarchical, Online Clustering of Web Search Results,” 6th Asia-Pacific Web Conference, APWeb 2004, Hangzhou, China, April 14-17, 2004. Proceedings, Lecture Notes in Computer Science, Springer-Verlag Berlin.

[14] Hua-Jun Zeng, Qi-Cai He, Zheng Chen, Wei-Ying Ma and Jinwen Ma, “Learning to cluster web search results,” in Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, 2004, pp. 210-217.

[15] S. Popova, I. Khodyrev, A. Egorov, S. Logvin, S. Gulyaev, M. Karpova and D. Muromtsev, “Sci-Search: Academic Search and Analysis System Based on Keyphrases,” in: the 4th Conference on Knowledge Engineering and Semantic Web, Russia. Communications in Computer and Information Science series, V. 394, Springer, 2013, pp. 281-288.

[16] C. Gutwina, G. Paynterb, I. Wittenb, C. Nevill-Manningc and E. Frank, “Improving browsing in digital libraries with keyphrase indexes,” Journal Decision Support Systems - From information retrieval to knowledge management: enabling technologies and best practices archive, Volume 27, Issue 1-2, Nov. 1999, pp. 81 – 104.

[17] P. Makagonov, A. Alexandrov and A. Gelbukh, “Clustering Abstracts instead of Full Texts,” in: Text, Speech, Dialog, LNAI N 3206, Springer, 2004, pp. 129–135.

[18] M. Alexandrov, A. Gelbukh and P. Rosso, “An Approach to Clustering Abstracts,” in Proc. 10th Int. NLDB-05 Conf., volume 3513 of Lecture Notes in Computer Science, Springer-Verlag, 2005, pp. 8–13.

[19] Stanford POS tagger, DOI: http://nlp.stanford.edu/software/tagger.shtml

[20] The Porter Stemmer, DOI: http://tartarus.org/martin/PorterStemmer

[21] S. Popova, L. Kovriguina, D. Muromtsev and I. Khodyrev, “Stop-words in Keyphrase Extraction Problem,” in Proc. of 14th Conference of Open Innovations Association FRUCT. Helsinki, Finland, 2013, pp. 113–121