Ranking in keyphrase extraction problem: is it suitable to use statistics of words occurrences?

Svetlana V. Popova[1], Ivan A. Khodyrev[2,3]

1 Saint-Petersburg State University, Saint-Petersburg Russia
svp@list.ru
2 Saint-Petersburg State Electrotechnical University, Saint-Petersburg Russia
3 VISmart, Saint-Petersburg Russia
kivan.mih@gmail.com

Abstract. The paper deals with keyphrase extraction problem for single documents, e.g. scientific abstracts. Keyphrase extraction task is important and its results could be used in a variety of applications: data indexing, clustering and classification of documents, meta-information extraction, automatic ontologies creation etc. In the paper we discuss an approach to keyphrase extraction, its' first step is building of candidate phrases which are then ranked and the best are selected as keyphrases. The paper is focused on the evaluation of weighting approaches to candidate phrases in the unsupervised extraction methods. A number of in-phrase word weighting procedures is evaluated. Unsuitable approaches to weighting are identified. Testing of some approaches shows their equivalence as applied to keyphrase extraction. A feature, which allows to increase the quality of extracted keyphrases and shows better results in comparison to the state of the art, is proposed. Experiments are based on Inspec dataset.

Keywords: Keyphrase extraction, keyphrase ranking, statistical features for keyphrase ranking, information extraction, scientific abstracts processing.

1 Introduction

The paper deals with the keyphrase extraction problem for single documents. We define keyphrase as a word or a group of words, which reflects the domain-specific of the text Keyphrase extraction could be used further in different natural language processing applications such as data indexing [1], clustering documents [2-4], automatic ontology creation etc. We are using results of this paper in an academic search system [4], we are mainly interested in a keyphrase extraction task from abstracts of scientific papers, because most abstracts are freely available and texts of papers are usually not. We focus on analysis of approaches to keyphrase selection from a set of candidates, built for a document [5-8]. The weighted approach is used to evaluate quality of a particular candidate, then after the ranking procedure, the best candidates are selected as keyphrases. In the paper we use only statistical information related to the word frequency in single documents and in a document collection. It is also shown that a number of measures is not adequate and some other measures are almost equivalent. We have shown that usage

of some measure estimated by researchers as suitable, in reality leads to the situation where measured phrases are selected almost randomly and thus such measures could be considered equivalent for the annotation task. The novel feature which is proposed in the paper, is based on the exclusion of one-word phrases from candidates, that increases significantly the annotation quality. The reminder of the paper is organized as follows. Section 2 is dedicated to the state of the art. In Section 3 experiment is described and description of test collection is provided. In section 4 the experiment's results are presented and discussed. In Section 5 additional experiment and its results are presented and discussed. Section 6 contains conclusions.

2 State-of-the-Art

There are two main approaches to solve the keyphrase extraction task. The first is based on single word ranking, best words selection and concatenation of best words following each other in the text [9-12]. The dominating approach [5-8, 12-16] consists of two stages: a selection stage, when candidate phrases are selected, and a classifying or ranking stage. On the selection stage a number of procedures is used to extract candidate phrases: n-gram extraction, noun phrase extraction, word sequence extraction or their combinations, which satisfy some limitations. The examples of limitations are following: length limit of a phrase (usually not more than 4-5 words per phrase), parts of speech limits, etc. It has been shown that keyphrases should consist from nouns and adjectives to achieve the best results and this result is actively used. In [14] the author proposes to use part of speech information in classification process. In pioneer systems on the second stage supervised methods were used to decide for each candidate whether it is keyphrase. In [15] a Naive Bayes classifier is used. In [16] a keyphrase extraction process is based on a number of threshold values of some variables which are optimized using genetic algorithm. These methods [14-16] could be used for the case, when there is a set of documents with keyphrases already extracted by the expert. On the ranking stage all candidate phrases are weighted and ranked. Then *k*-best candidate phrases are selected as keyphrases. Ranking methods are usually based on phrase weight measurement [5-7, 12, 13]. In this case, statistical measurements are often used for phrases and phrase words as well as information about the first position of a phrase in text and the size of a phrase with its frequency. However, researchers do not address and analyze possibilities of different variants of phrases' weight evaluation based on in-phrase word's weights. In this paper, we fill the gap. We evaluate several approaches to phrase weighting and use a number of statistical measures for this task. Experiments have shown that selected statistical measures do not allow identifying correct keyphrases among other phrases. It seems that simple exclusion of some set of candidates is more efficient, that is the set where most keyphrases are not correct apriori. In presented paper we have shown that the set of one-word candidate phrases is a set of this kind and its exclusion leads to relatively good results. As a result of current research we make a statement about possible reasons, why information about the length of a phrase influences the result of keyphrase extraction.

3 Experiment Description

3.1 Candidate Phrase Ranking

One of the goals of the presented paper is to analyze a number of approaches to phrase weight measurement. We deal only with weight measurement based on in-phrase words evaluation. We are using the following notations. The phrase with *n* words is denoted as (w_1, w_2,…,w_n), where *w* is a single word. Phrase weight is denoted as *weight*(w_1, w_2,…,w_n) and the weight of a word as *weight*(*w*). We measure weights of phrases as:

1. Average weight among in-phrase words:

$$weight(w_1, w_2, \dots, w_n) = \frac{\sum_{i=1}^{n} weight(w_i)}{n}. \quad (1)$$

2. Geometric mean of word weights in phrase:

$$weight(w_1, w_2, \dots, w_n) = \sqrt[n]{\prod_{i-1}^{n} weight(w_i)}. \quad (2)$$

3. Degree of relationship between words in a phrase and a main word in a phrase.

For the case 3 (degree of relationship between words in a phrase and a main word in a phrase) six measuring approaches described below were used to determine a main word in a phrase. Word *w* is determined as w^{main} for the phrase if its weight is the best weight in a phrase compared to the weights of other words in a phrase. When the main word has been chosen the relationship value between each other word *w* in a phrase and main word w^{main} is calculated. In our research Two measures were used to calculate words relation:

- Pointwise mutual information, calculated between the main word w^{main} and every other word *w* in a phrase:

$$MI(w^{main}, w) = \frac{p(w^{main}, w)}{p(w^{main}) * p(w)}, \quad (3)$$

where $p(w^{main},w)$ is a probability to meet word w^{main} next to every other word in-phrase *w* (in window 3), $p(w^{main})$ and $p(w)$ are probabilities of meeting words w^{main} and *w*. A phrase weight is defined as an average among the obtained values:

$$weight(w_1, w_2, \dots, w_n) = \frac{\sum_{i=1}^{n-1} MI(w^{main}, w_i)}{n-1}. \quad (4)$$

- Word w^{main} and word *w* relationship:

$$rel(w^{main}, w) = \max\{p(w^{main}|w), p(w|w^{main})\}, \quad (5)$$

$$p(w_1|w_2) = \frac{\sum_{d \in D_{w_2}} tf^d(d, w_1)}{\sum_{d \in D_{w_2 w'}} tf^d(d, w')}, \quad (6)$$

where D_{w_2} – set of all documents that contains w_2, $tf^d(d,w_1)$ – the number of occurrence of the word w_1 in the document d, w' belong to words in D_{w_2}. An average of obtained values is defined as a weight of a phrase as in (4) but $rel(w^{main}|w)$ is used instead $MI(w^{main},w)$.

To evaluate the weight of a word *weight*(*w*) in a text *d* for (1) and (2) or for a selection of main word in phrase, we use the following six values:

- Number of documents where the word *w* occurs at least once (*df*).
- Within collection word *w* frequency (*tf*).
- Within document *d* word *w* frequency (tf^d);
- Ratio: *tf*/*df*
- *tf-idf* [17]:

$$weight(w) = tf^{\mathrm{d}}(\mathrm{w}) \cdot log\frac{N}{df(w)}, \quad (7)$$

where *N* is the number of documents in the collection.

- The evaluation of word's *w* context narrowness (word context).

Concept of narrow context is borrowed from [18]. Words with narrow context are domain-specific. For example, "motherboard" is the word with narrow context. If a document contains this word we can conclude with high probability that this document is about computer hardware. The word "computer" has wide context. If a document contains such word it is difficult to define the content of this document. It can be about hardware, art, health, e.t.c. with almost the same probabilities. Simplifying the method of detection words with narrow context [18], we define for each word *w* its context $p(\mathrm{Y}|w)$ by using $p(y|w)$ (6), where *y* belongs to collection's vocabulary. Then entropy *H* is calculated for every obtained context. Based on assumption that the context of word with narrow context has low entropy, we use word's context entropy to evaluate words:

$$weight(w) = H(\mathrm{Y}, w) = -\sum_y p(y|w) log p(y|w). \quad (8)$$

The best word's weight in a phrase for *df*, *tf*, tf^d, *tf*/*df*, *tf-idf* is the highest weight and for word context is the lowest weight.

3.2 Data Preprocessing and Candidate Phrase Extraction

Presented paper is focused on the problem of ranking of candidate phrases. Thus, we used basic algorithm for candidate extraction described as follows. The POS-tagged text is fed to the input of the algorithm (we used Stanford POS-tagging tool [19]). The sequences of nouns and adjectives are extracted from the text. Stop words, punctuation and other parts of speech, excluding nouns and adjectives, are used on this stage as delimiters. The size of obtained sequences is limited to 5. All extracted sequences are considered as candidate phrases.

3.3 Dataset

We have used Inspec dataset collection for our research, because in presented paper we are focusing on keyphrase extraction from abstracts of scientific articles. Inspec contains annotations to scientific articles in English (from disciplines "Computers and Control", and "Information technology"). Inspec collection contains three sub-collections: training dataset (1000 documents), evaluation dataset (500 documents) and testing dataset (500 documents). Each text has a gold standard, which contains phrases, extracted by an expert. Gold standard includes two types of annotations: contr set and uncontr set. As in most other papers [9, 12, 14, 20, 21] test dataset and uncontr gold standard set are used for this paper. A detailed collection description is presented in [14].

3.4 Evaluation

To measure the quality of extracted keyphrases we use the traditional approach based on F-score, which is a combination of Precision and Recall [17], and is one of the most popular quality measures in keyphrase extraction domain:

$$F - score = \frac{2 * Precision * Recall}{Precision + Recall},$$

$$Precision = \frac{|G \cap C|}{G}, Recall = \frac{|G \cap C|}{C},$$

where *G* is the number of automatically extracted kephrases from all documents and *C* is the number of all keyphrases extracted by expert (number of phrase in the gold standard). In the case when a number of extracted keyphrases is less than given in the gold standard Precision is used instead F-score as it depends on the number of correct phrases among the extracted keyphrases. Otherwise, F-score declines with decrease of the number of extracted phrases because Recall also declines. When the number of extracted keyphrases is the same as in the gold standard, F-score and Precision are identical because *G* equals *C*.

3.5 Experiment

On the first stage, candidate phrases were extracted for each text using approach proposed in section 3.2. For each phrase in a document its weight is calculated. Weight calculation is done using strategies described in 3.1 as average weight of all words in a phrase (1), as a geometric mean of all words in a phrase (2), as an average weight of relation between main word and other words in a phrase (3-6). One of six measures presented in 3.1 was used for a word's weight evaluation: *tf-idf* (7), *df*, *tf*, tf^d, *tf*/*df*, word context (8).

It is important to say that if a phrase contains only one word, then (3) and (6) are not usable, because they need at least two words to be calculated. For these cases one-word phrases were excluded. To compare this weight evaluation approach with the other available approaches, we have conducted experiment, where for each approach mentioned above one-word phrases were filtered. This experiment has shown interesting

results which are presented further. After weight evaluation, phrases were ranked according to their weights and *k*-best were selected as keyphrases. We have examined a number of cases to determine *k*:

- *k* was taken according to the number of phrases mentioned in the gold standard [12];
- *k* equals to 7;
- all candidate phrases are selected as keyphrases (no ranking was performed).

4 Experimental Results and Discussion

4.1 Experiment Results

Results of keyphrase extraction experiment are presented in Tables 1 and 2, the weight of a phrase was calculated as an average of word weights, contained in a phrase (1). To calculate the word's weight six approaches were tested (3.1) and appropriate results are presented in columns. The number of phrases to select was defined as follows: the same number as in the gold standard and 7 (this information is presented by rows). Table 1 presents results, when no phrase was filtered. Table 2 presents results, when all one-word phrases were filtered. Experiments, which results are presented in Table 3 and Table 4, differ to the experiments in Tables 1 and 2 only in the change of keyphrase weight function, for these experiments geometric mean was used (2). Table 5 presents results of experiments, where the phrase weight was calculated using main word, which was chosen among the words in the phrase and then pointwise mutual information (3) was calculated for each pair, where the first word was the main word and second word - every other word in the phrase. One-word phrases were filtered. The main word was selected as a word with the best weight in the phrase. To evaluate word weights measures, described in 3.1, were used: *tf-idf* (7), *df*, *tf*, tf^{l}, *tf*/*df*, word context (8). In Table 6 results of a similar experiments are shown for the case, when relationship of each word with the main word was calculated (5). Table 7 contains results of extracted keyphrases for the case, when the candidate phrases were not ranked and all of them were selected as keyphrases. Table 8 contains results for the case when keyphrases were selected randomly from the set of candidate phrases and the number of extracted keyphrases was equals the keyphrase number in the gold standard.

Table 1. Results: keyphrase weight was calculated as an average of weights among words in phrase weights

The number of extracted keyphrases	Evaluation measure	tf-idf	df	tf	tf^{l}	tf/df	word context
The same number as in gold standard	F-score	0.31	0.20	0.23	0.29	0.31	0.28
7	Precision	0.29	0.18	0.20	0.27	0.30	0.25

Table 2. Results: keyphrase weight was calculated as an average of weights among words in-phrase weights, when one-word phrases were filtered

The number of extracted ke-phrases	Evaluation measure	tf-idf	df	tf	tf^{l}	tf/df	word context
The same number as in gold standard	F-score	**0.40**	0.36	0.37	**0.39**	**0.40**	**0.39**
7	Precision	**0.39**	0.34	0.36	**0.39**	**0.40**	**0.40**

Table 3. Results: keyphrase weight was calculated as a geometry mean of weights among words in-phrase weights

The number of extracted ke-phrases	Evaluation measure	tf-idf	df	tf	tf^{l}	tf/df	word context
The same number as in gold standard	F-score	0.31	0.18	0.19	0.29	0.31	0.29
7	Precision	0.28	0.14	0.16	0.26	0.29	0.25

Table 4. Results: keyphrase weight was calculated as a geometry mean of weights among words in-phrase weights, when one-word phrases were filtered

The number of extracted ke-phrases	Evaluation measure	tf-idf	df	tf	tf^{l}	tf/df	word context
The same number as in gold standard	F-score	**0.40**	0.36	0.37	**0.39**	**0.41**	**0.40**
7	Precision	**0.40**	0.34	0.35	**0.39**	**0.42**	**0.40**

Table 5. Results: main word was selected, then pointwise mutual information was calculated between main word and other words in-phrase and average values was calculated as a score of a phrase

The number of extracted ke-phrases	Evaluation measure	tf-idf	df	tf	tf^{l}	tf/df	word context
The same number as in gold standard	F-score	**0.39**	**0.39**	**0.39**	**0.40**	**0.40**	**0.40**
7	Precision	**0.40**	**0.40**	**0.40**	**0.40**	**0.40**	**0.40**

Table 6. Results: main word was selected, words relationship was calculated betwee main word and other words in-phrase (5), average values was calculated as a score of a phrase

The number of extracted key-words	Evaluation measure	tf-idf	Df	tf	tf^{l}	tf/df	word context
The same number as in gold standard	F-score	**0.40**	**0.40**	**0.40**	**0.40**	**0.40**	**0.40**
7	Precision	**0.41**	**0.41**	**0.41**	**0.41**	**0.41**	**0.41**

Table 7. Results: all candidate phrases were selected as keyphrases

Including/Excluding one-word phrase candidates	F-score
Without one-word phrase filtering	0.30
One-word phrase filtering was used	**0.40**

Table 8. Results: keyphrases were selected randomly

Including/Excluding one-word phrase candidates	F-score
Without one-word phrase filtering	0.23
One-word phrase filtering was used	0.38

4.2 Discussion

Results presented in Table 1 and Table 3 show that usage of *tf* (within collection term frequency) and *df* (within collection document frequency) measures to evaluate words weight decreases the quality of extracted keyphrases even in comparison with arbitrary selection (Table 8). Other measure's results do not differ much regardless the way how phrase weight is calculated and these measures we will discuss below.

Experiments show that results in Tables 2, 4, 5, 6 are very similar. Thus we can conclude that all methods give near the same results in respect to one-word phrases filtering, regardless of a way to weight words and regardless of the number of extracted keyphrases. Slightly better result is achieved when keyphrase weight is calculated as a geometry mean and *tf/df* is used.

Another interesting observation is the fact that filtering one-word phrases significantly increases quality of remained keyphrases and improves results of the state of the art [9, 12, 14]. It is interesting that if we only filter out all the one-word keyphrases without performing resulting ranking at all, we will get F-score=0.40, the same result as with ranking. So it seems that ranking doesn't improve quality of keyphrases.

In fact experiments show that filtering of one-word keyphrases makes significantly greater impact than phrase weighting, based on statistics mentioned above. We have made an assumption as well, that all ranking approaches, mentioned above, essentially select keyphrases randomly and thus the results of different approaches are very close. To prove it an additional experiment was conducted, which goal was to show that the

ratio between correct and incorrect keyphrases before and after ranking remains almost the same.

5 Additional Experiment

5.1 Experiment Description

The goal of proposed additional experiment is to show that all phrase-ranking approaches, used to select keyphrases in this paper, essentially select keyphrases randomly. Input data to the experiment is a set of pre-ranked phrase candidates. For this set for each phrase-length a number of phrases is set, and also known the number of correct and incorrect phrases. The ranking algorithm forms the output data, which is a set of selected keyphrases with the information about the number of selected phrases for each phrase length, including information about correctness of such selection. Number of selected keyphrases is the same as in the gold standard. The goal is to evaluate the ratio between all phrases and correct phrases before and after keyphrase selection step.

5.2 Experiment Results and Discussion

Because experiments in section 4 give almost the same results for a number of measures, here we are using only one of them – tf^{d} (within document frequency) measure. Experimental results are described in Table 9. In first column phrase length is presented and also the information about one-word phrases inclusion during experiment: are they filtered or not. In other columns additional information is presented: number of candidate phrases, how many of them are correct, ratio between the number of candidates and the number of correct among them and the same information for the case when ranking is performed.

For keyphrases of 2-4 words length ratio between the number of phrases to the number of correct keyphrases lies inside range 2-3 (before and after ranking) and for one-word phrases this ratio is close to 8 on input data and is close to 6 on output data. It means that the set of one-word keyphrases contains much more incorrect keyphrases than correct ones. Notice that the number of one-word phrases in input data is the third part of all phrases. Thus it becomes obvious why filtering one-word phrases yields much better results. When we filter one-word phrases and arbitrary select the number of keyphrases as in the gold standard the F-score = 0.38 which is better than state of the art results for Inspec, which use complex ranking techniques [9][12][14]. Analysis of experimental results in Table 9 shows that the ratio between all keyhrases and correct keyphrases after ranking slightly improves the result before ranking. Taking this fact and results from Section 4 (in which it was shown, that using one-word phrase filtering, results of all methods are nearly the same) into account we can conclude that the results of all methods, which were investigated in this paper (excluding *tf* and *df*) are quite close to results of random pick of phrases from initial set. This result also shows that methods that weight phrases using information about phrase length should work good

Table 9. Results of additional experiment

INSPEC Phrase's length	The number of extracted candidate phrases	The number of correctly extracted candidate phrases	Ratio between the number of candidates to the number of correct among them	The number of extracted keyphrases after ranking	The number of correctly extracted keyphrases after ranking	Ratio between the number of keyphrases to the number of correct among them (after ranking)
With filtering one-word phrases						
2	4349	1552	2.80	2873	1233	2.33
3	1577	625	2.53	1195	513	2.33
4	370	128	2.89	299	109	2.74
5	130	34	3.82	116	31	3.74
Without filtering one-word phrases						
1	3056	392	**7.80**	1450	244	**5.90**
2	4349	1552	2.80	1698	798	2.13
3	1577	625	2.53	780	351	2.22
4	370	128	2.89	203	84	2.42
5	130	34	3.82	81	24	3.38

on Inspec dataset (longer phrases usually evaluate with more weight than short phrases and so one-word phrases become filtered). Remind that one-word phrase consists of alone noun/adjective and separated from other nouns and adjectives by punctuation, stop-words and other words excluding nouns and adjectives.

6 Conclusion

The results of presented research show that investigated approaches to phrase weighting (excluding *tf* and *df*) show almost equal results and only slightly increase random phrase selection from phrase candidates. They differ mostly in the way how they rank one-word phrases. If one-word phrases are excluded, all methods would give rather similar results. Exclusion of one-word candidate phrases increases extraction quality, because in one-word phrases ratio between correct keyphrases and all phrases is significantly bigger comparing to the phrases of other lengths.

Experiments were based on Inspec dataset, which is popular for the task of keyphrase extraction from scientific abstracts. Experiments prove that for this collection good results will be given by algorithms which filter one-word phrases, even if other phrases are ranked randomly. This result should be considered when working with Inspec collection and further evaluating approaches, investigated in this paper.

Acknowledgments: This work is supported by federal target program "Kadry". The program Research project 16.740.11.0751. Code of competition 2011-1.2.1-302-031. Code of application 2011-1.2.1-302-031/5.

Reference

1. Gutwina, C., Paynterb, G., Wittenb, I., Nevill-Manningc C., Frankb E.: Improving browsing in digital libraries with keyphrase indexes. Journal of Decision Support Systems, 27(1-2), pp. 81–104 (1999)
2. Zhang, D. and Dong, Y.: Semantic, Hierarchical, Online Clustering of Web Search Results. In: 6th Asia-Pacific Web Conference. Hangzhou, China (2004)
3. Zeng, H.J., He, Q.C., Chen, Z., Ma, W.Y., Ma, J.: Learning to cluster web search results. In: the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 210-217 (2004)
4. Popova, S., Khodyrev, I., Egorov, A., Logvin, S., Gulyaev, S., Karpova, M., Muromtsev, D.: 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, (2013) (accepted)
5. Pudota, N., Dattolo, A., Baruzzo, A., Ferrara, F., Tasso, C.: Automatic keyphrase extraction and ontology mining for content-based tag recommendation. International Journal of Intelligent Systems, vol 25, pp. 1158-1186 (2010)
6. You, W., Fontaine, D., Barhes, J.-P.: An automatic keyphrase extraction system for scientific documents. In: Knowl Inf Syst 34, pp. 691-724 (2013)
7. El-Beltagy, S. R., and Rafea, A.,: KP-Miner: A keyphrase extraction system for english and arabic documents. In: Information Systems, 34, pp. 132-144 (2009)
8. Popova, S., Khodyrev, I.: Keyphrase extraction and ranking in annotation problem. Journal Nauchno-Texnicheskiy Vestnik Informatsionnix technologiy mechaniki i optiki, Vol. 1 (2013) (Попова С. В., Ходырев И. А.: Извлечение и ранжирование ключевых фраз в задаче аннотирования. Научно-технический вестник информационных технологий, механики и оптики, выпуск 1, 2013)
9. Mihalcea, R., Tarau, P.: TextRank: Bringing order into texts. In: Conference on Empirical Methods in Natural Language Processing, pp. 404–411 (2004)
10. Xiaojun, W. and Xiao, J.: Single document keyphrase extraction using neighborhood knowledge. In: Proceedings of the 23rd AAAI Conferenceon Artificial Intelligence, pp. 855–860 (2008)
11. Xiaojun W., Xiao J.: Exploiting Neighborhood Knowledge for Single Document Summarization and Keyphrase Extraction ACM Transactions on Information Systems, 28(2), Article 8 (2010)
12. Zesch, T., Gurevych, I.: Approximate Matching for Evaluating Keyphrase Extraction. In: International Conference RANLP 2009. pp. 484–489, Borovets, Bulgaria (2009)
13. Kim, S.N., Medelyan, O., Yen, M.: Automatic keyphrase extraction from scientific articles. Language Resources and Evaluation, Springer Kan & Timothy Baldwin (2012)

14. Hulth A.: Improved automatic keyword extraction given more linguistic knowledge. In: Conference on Empirical Methods in Natural Language Processing, pp. 216–223 (2003)
15. Frank, E., Paynter, G.W., Witten, I.H., Gutwin, C., Nevill-Manning, C.G.: Domain-specific keyphrase extraction. In: Proc. of IJCAI. pp. 688–673 (1999)
16. Turney, P.: Learning to Extract Keyphrases from Text. In: NRC/ERB-1057, pp. 17– 43 (1999)
17. Manning, C., Raghavan, P., Schutz,e H.: Introduction to Information Retrieval. Cambridge University Press (2009)
18. Dobrynin, V., Patterson, D., Rooney, N.: Contextual Document Clustering. In Advances in Information Retrieval. Lecture Notes in Computer Science. 2997, pp.167-180 (2004)
19. Standford POS tagging tool DOI: http://nlp.stanford.edu/software/tagger.shtml (09.11.2012).
20. Tsatsaronis, G., Varlamis, I., Norvag, K.: SemanticRank: Ranking Keywords and Sentences Using Semantic Graphs. In: Proc. of the 23rd International Conference on Computational Linguistics, pp. 1074–1082 (2010)
21. Hasan, K. S., Ng, V.: Conundrums in Unsupervised Keyphrase Extraction: Making Sense of the State-of-the-Art. In: Coling, Poster Volume, Beijing, pp. 365–373 (2010)