A Graph-Based Keyphrase Extraction Model with Three-Way Decision

Tianlei Chen[1,2], Duoqian Miao[1,2](✉), and Yuebing Zhang[1,2]

[1] Department of Computer Science and Technology, Tongji University,
Shanghai 201804, China
ctlchentianlei@163.com, dqmiao@tongji.edu.cn
[2] Key Laboratory of Embedded System and Service Computing,
Ministry of Education, Tongji University, Shanghai 201804, China

Abstract. Keyphrase extraction has been a popular research topic in the field of natural language processing in recent years. But how to extract keyphrases precisely and effectively is still a challenge. The mainstream methods are supervised learning methods and graph-based methods. Generally, the effects of supervised methods are better than unsupervised methods. However, there are many problems in supervised methods such as the difficulty in obtaining training data, the cost of labeling and the limitation of the classification function trained by training data. In recent years, the development of the graph-based method has made great progress and its performance of extraction is getting closer and closer to the supervised method, so the graph-based method of keyphrase extraction has got a wide concern from researchers. In this paper, we propose a new model that applies the three-way decision theory to graph-based keyphrase extraction model. In our model, we propose algorithms dividing the set of candidate phrases into the positive domain, the boundary domain and the negative domain depending on graph-based attributes, and combining candidate phrases in the positive domain and the boundary domain qualified by graph-based attributes and non- graph-based attributes to get keyphrases. Experimental results show that our model can effectively improve the extraction precision compared with baseline methods.

Keywords: Keyphrase extraction · Three-way decision · Graph-based

1 Introduction

Keyphrase extraction has been a popular research topic in the natural language processing research field. Especially with the current increasing requirements for applications of texts, keyphrase extraction has attracted widespread attention from researchers. Although it has been greatly developed in recent years at home and abroad, the extracted results are far from the ideal.

With the rapid growth of text applications, the analysis of text data has become an important research area that has attracted much attention. Among them, how to extract keyphrases that reflect the subjects of texts has always been a research hotspot in the field of natural language processing, and its research results can be widely used in text retrieval, text summarization, text classification and question answering systems.

R. Bello et al. (Eds.): IJCRS 2020, LNAI 12179, pp. 111–121, 2020.
https://doi.org/10.1007/978-3-030-52705-1_8

Especially with the rise of research on unstructured big data of texts in recent years, the issue of keyphrase extraction has received in-depth research, and many researches have appeared in the international top conferences of artificial intelligence and natural language processing, such as the International Joint Conference on Artificial Intelligence (IJCAI) [1], The Annual Meeting of the Association for the Advance of Artificial Intelligence (AAAI) [2–4], International Computational Linguistics Association The Annual Meeting of the Association for Computational Linguistics (ACL) [5], The International Conference of World Wide Web (WWW) [6] and Conference on Empirical Methods in Natural Language Processing (EMNLP) [7], etc.

Researchers generally believe that the extracted keyphrases [8] should meet the following basic standards: (i) Keyphrases should be meaningful phrases. For example, "keyphrase extraction" is a meaningful phrase, but "and" does not meet the standard. (ii) Keyphrase extraction should meet the relevance standard that keyphrases must be closely related to the subjects of texts, which is the most essential requirement for keyphrase extraction. For example, the subtitle "Introduction" in this paper is not an appropriate keyphrase obviously. (iii) Keyphrase extraction should correspond to the coverage standard. Keyphrases should be able to cover various topics of the text and the main aspects of each topic, not just focus on only one topic and ignore others. (iv) Keyphrases extraction should meet the coherence standard. Several keyphrases of the text should be semantically and logically related. For an instance, a piece of academic paper that mainly introduces a graph-based keyphrase extraction model. The set of keyphrases is {"keyphrase extraction", "graph-based"}, which is more suitable than {"keyphrase extraction", "target detection"}. (v) Keyphrase extraction should correspond the conciseness standard. The number of keyphrases is limited, and the set of keyphrases should not contain any redundant phrase.

To meet any of the above standard, there is a huge challenge. Although there are many methods to solve this scientific problem such as statistical-based methods, supervised learning methods and graph-based methods, how to extract keyphrases precisely and efficiently is still a challenge.

In this paper, we propose a new model that applies the three-way decision theory to the graph-based keyphrase extraction model. In our model, we propose algorithms dividing the set of candidate phrases into the positive domain, the boundary domain and the negative domain depending on graph-based attributes, and combining candidate phrases in the positive domain and the boundary domain qualified by graph-based attributes and non-graph-based attributes to get keyphrases. Experimental results show that our model can effectively improve the extraction precision compared with baseline methods.

In Sect. 2, we briefly introduce the three-way decision theory and some related works in the field of keyphrase extraction. In Sect. 3, we describe the structure of our model and algorithms we proposed. In Sect. 4, we report the experimental results and analysis. Finally, we make a conclusion in Sect. 5.

2 Related Work

2.1 Statistical-Based Methods

Using statistical-based methods to extract keyphrases of texts is relatively simple, because it requires neither training data nor external knowledge. After the preprocessing of texts, simple statistical rules can be used to form a set of candidate phrases. The estimation of candidate phrases usually uses quantification of feature values. The main statistical-based keyphrase extraction method is TF-IDF (Term Frequency-Inverse Document Frequency) [9] and its improved methods. The advantage of the TF-IDF algorithm is that it is simple and fast. However, the traditional TF-IDF algorithm also has obvious shortcomings that it is not comprehensive enough to measure the importance of phrases based on the frequency. Sometimes important phrases may not appear frequently.

2.2 Graph-Based Methods

The graph-based keyphrase extraction method is the most effective and widely studied unsupervised keyphrase extraction method, because the method considers the co-occurrence relationship between phrases in the text. If there is a co-occurrence relationship between two phrases, it indicates that they are semantically related in the text. On the other hand, the graph-based method can incorporate more other features, so it has reached better effect of Extraction. The graph-based method has been widely concerned by researchers, from the TextRank method proposed by Mihalcea [10] to the PositionRank method proposed by Florescu [4]. In this paper, we propose a new model that applies the three-way decision theory to graph-based keyphrase extraction method.

2.3 Three-Way Decision

As generally considered, there are only acceptance and rejection in making a decision, which is a two-branch decision model, but it is often not the case in practical application. Based on the rough set theory proposed by Pawlak [11], Yao's three-way decision theory [12] provided a third alternative. The idea of three-way decision is based on three categories: acceptance, rejection and non-commitment. The goal is to divide a domain into three disjoint parts. Positive rules acquired from positive domain are used to accept something, negative rules acquired from negative domain are used to deny something, and rules that fall on boundary domain need further observation, which called delayed decision-making. Miao [13] has made some researches about three-way decision theory with multi-granularity, and Zhang [14] has applied it to the application of sentiment classification. The way of three-way decision describes the thinking mode of human beings in solving practical decision-making problems.

3 The Model with Three-Way Decision

3.1 Structure of the Model

We propose a graph-based keyphrase extraction model with three-way decision. As Fig. 1 illustrated, we could obtain candidate phrases through the preprocessing of texts from the raw, and then transform texts to text graphs with candidate phrases as nodes to get their graph-based attributes and non-graph-based attributes. With the support of the three-way decision theory, we divide the set of candidate phrases into the positive domain, the boundary domain and the negative domain depending on their graph-based attributes, and combine candidate phrases in the positive domain and the boundary domain qualified by their graph-based attributes and non-graph-based attributes to get keyphrases.

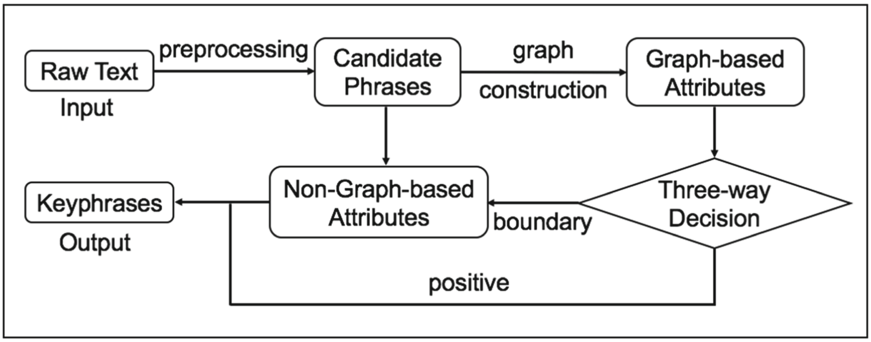

Fig. 1. The structure of the model with three-way decision

3.2 Preprocessing of Texts and Graph Construction

The step of preprocessing of texts from the raw plays an important role in the process of extracting keyphrases due to its output affecting the result deeply. The generic preprocessing way of graph-based keyphrases extraction: (i) Tokenizing: The process of tokenizing is to split strings into phrases. (ii) Tagging [15]: The task of tagging is to tag part-of-speech of phrases preparing for filtering. (iii) Filtering: Filter out phrases that do not meet the part-of-speech requirements according to the result of tagging. (iv) Stemming [16, 17]: Stemming phrases is in order to eliminate the effects of phrases forms that can get the main part of phrases. The differences between the phrases before stemming and after stemming are as follows (Table 1):

After preprocessing the raw, we construct the text graph to obtain graph-based attributes of candidate phrases. The text graph $\mathrm{G} = (\mathrm{V}, \mathrm{E}, \mathrm{W})$, V is the set of nodes representing candidate phrases, E is the set of edges and W is the set of corresponding

Table 1. Examples for stemming results

Before stemming	After stemming
Harmonic	Harmon
Effective	Effect
Axiomatized	Axiom
Reality	Real
Validated	Valid

edge weights where weight w_{ij} for an edge e_{ij} indicates the frequency of two phrases v_i and v_j co-occurring in consecutive sentences, adopting the context-aware graph construction method from Duari [18] due to its simple construction method and well performance. The higher value of w_{ij} is, the stronger relationships between v_i and v_j are.

3.3 Keyphrase Extraction with Three-Way Decision

In our opinion, the three-way decision is making a delayed decision on uncertainty, and decides based on other information in the future. In this paper, we propose two Algorithms, which applies the three-way decision theory to the graph-based keyphrase extraction model (see Algorithm 1 and Algorithm 2). The main notations in this paper are listed in Table 2.

Table 2. The list of main notations

Variable	Explanation
c_i	Candidate phrases
ga_i	Graph-based attributes
nga_i	Non-graph-based attributes
r_i	Keyphrases extraction results
p	The positive domain
b	The boundary domain
n	The negative domain
C_i	The set of candidate phrases
G_i	The set of graph-based attributes
NG_i	The set of non-graph-based attributes
R	The set of keyphrases extraction results
Th	The threshold of the three-way decision

Algorithm 1 Classify the candidate phrases by graph-based attributes

```
Input: The set of candidate phrases C = $\{c_1, c_2, \ldots, c_n\}$ and its graph-based attributes
G = $\{ga_1, ga_2, \ldots, ga_n\}$
Output: $C_p = \{c_{(1)}, c_{(2)}, \ldots, c_{(Th*n)}\}$; $C_b = \{c_{(Th*n+1)}, c_{(Th*n+2)}, \ldots, c_{((1-Th)*n)}\}$
and $C_n = \{c_{((1-Th)*n+1)}, c_{((1-Th)*n+2)}, \ldots, c_{(n)}\}$
for i = 1 to n do
    if $ga_i$ ranked in top Th $*$ n then
        put $c_i$ from C into $C_p$;
    else if $ga_i$ ranked in bottom Th $*$ n then
        put $c_i$ from C into $C_n$;
    else
        put $c_i$ from C into $C_b$;
    end if
end for
```

From Cohen's [19] Trusses theory, for a weighted, undirected and simple graph $\mathrm{G} = (\mathrm{V}, \mathrm{E}, \mathrm{W})$, a k-truss subgraph of G is the maximal subgraph $G_k = (V_k, E_K, W_k)$, such that each edge $e_{ij} \in E_k$ belongs to at least $(\mathrm{k}-2)$ triangles. The truss level of an edge e_{ij} is k if it lies in k-truss but not in $(\mathrm{k}+1)$-truss. Kaur [20] expanded the concept of truss to nodes and defined truss level λ_i of node $\mathrm{v_i}$ as follows.

$$\lambda_i = max_{v_j \in N_i}\{l_{ij}\} \tag{1}$$

where $\mathrm{N_i}$ is the set of neighbours of node v_i and l_{ij} is the truss level of edge e_{ij}.

Based on the definition of the truss level of nodes, Duari [18] defined the semantic strength χ_i of node v_i and the semantic connectivity SC_i of node v_i as follows.

$$\chi_i = \sum\nolimits_{v_j \in N_\mathrm{i}} w_{ij} \times \lambda_\mathrm{j} \tag{2}$$

$$SC_i = \frac{|\{\lambda_k : v_k \in N_i\}|}{maxtruss} \tag{3}$$

We take these attributes on the basis of the graph into account and define the graph-based attributes ga_i of node v_i as follows.

$$ga_i = \lambda_i \times \chi_i \times SC_i \tag{4}$$

In this paper, we propose Algorithm 1 to classify the candidate phrases by graph-based attributes and divide the set of candidate phrases C into the positive domain C_P, boundary domain C_b and negative domain C_n respectively.

Algorithm 2 Extract keyphrases with three-way decision

```
Input: Set of candidate phrases in the positive domain, boundary domain, negative domain
$C_p = \{c_{(1)}, c_{(2)}, \ldots, c_{(Th*n)}\}$, $C_b = \{c_{(Th*n+1)}, c_{(Th*n+2)}, \ldots, c_{((1-Th)*n)}\}$,
$C_n = \{c_{((1-Th)*n+1)}, c_{((1-Th)*n+2)}, \ldots, c_{(n)}\}$ and their corresponding attributes sets
$G_p = \{ga_{(1)}, ga_{(2)}, \ldots, ga_{(Th*n)}\}$, $G_b = \{ga_{(Th*n+1)}, ga_{(Th*n+2)}, \ldots, ga_{((1-Th)*n)}\}$,
$G_n = \{ga_{((1-Th)*n+1)}, ga_{((1-Th)*n+2)}, \ldots, ga_{(n)}\}$, $NG_p = \{nga_{(1)}, nga_{(2)}, \ldots, nga_{(Th*n)}\}$,
$NG_b = \{nga_{(Th*n+1)}, nga_{(Th*n+2)}, \ldots, nga_{((1-Th)*n)}\}$,
$NG_n = \{nga_{((1-Th)*n+1)}, nga_{((1-Th)*n+2)}, \ldots, nga_{(n)}\}$
Output: Set of keyphrases $R = \{r_1, r_2, \ldots, r_k\}$
if $Th * n \geq k$ then
    for $i = 1$ to $Th * n$ do
        put $ga_i * nga_i$ ranked top k from $C_p$ into R;
    end for
else if $(1 - Th) * n \leq k$ then
    put $C_p$ and $C_b$ into R;
    for $i = (1 - Th) * n + 1$ to n do
        put $ga_i * nga_i$ ranked top $k - (1 - Th) * n$ from $C_n$ into R;
    end for
else
    put $C_p$ into R;
    for $i = Th * n + 1$ to $(1 - Th) * n$ do
        put $ga_i * nga_i$ ranked top $k - Th * n$ from $C_b$ into R;
    end for
end if
```

Position information is an important factor in identifying keyphrases except for graph-based attributes. Florescu [4] proposed PositionRank and took the position of candidate phrases into account to identify keyphrases, we regard it as non-graph-based attributes nga_i of node v_i with the following definition.

$$nga_i = \sum_{j}^{n_i} \frac{1}{p_j} \tag{5}$$

In this paper, we propose Algorithm 2 taking graph-based and non-graph-based attributions of the candidate phrases into account in the boundary domain. Generally, both of the candidate phrases in the positive domain and the boundary domain are considered as the output of the Algorithm 2, where Th is the threshold of the three-way decision and the value of k represents the count of keyphrases to extract.

4 Experiments and Results

4.1 Benchmark Datasets and Baseline Methods

We evaluate the performance of the model with two widely used benchmark datasets, which are Hulth2003 and Krapivin2009. Hulth2003 is a dataset including about 2,000 abstracts of academic articles. Krapivin2009 consists of over 2,000 scientific papers from computer science domain published by ACM used for keyphrase extraction specially. We use the uncontrolled list of keyphrases of Hulth2003 and gold-standard keyphrases of Krapivin2009 for evaluation. We take Textrank [10], DegExt [21], k-core retention [22] and PositionRank [4] as baseline methods and evaluate our model against them.

4.2 Performance Results and Discussions

Duari [18] reported that values of k are 25 for Hulth2003 and 10 for Krapivin2009 that yield the highest F1-measure with all algorithms mentioned above, which correlate with the average number of labeled keyphrases in datasets, and we adopted the reported values of k and the results of baseline methods. In the experiment, we separate a part of data from data sets as validation sets to explore the most appropriate value of Th. The results show the value of Th is 0.1 for Hulth2003 and 0.4 for Krapivin2009 yields the best performance (see Table 3 and Table 4).

Table 3. The performance of Hulth2003 (k = 25)

Th	Precision	Recall	F1
0.1	**43.92**	**63.28**	**51.85**
0.2	43.20	62.25	51.01
0.3	42.62	61.40	50.31
0.4	42.90	61.81	50.65

Table 4. The performance of Krapivin2009 (k = 10)

Th	Precision	Recall	F1
0.1	27.57	29.69	28.60
0.2	39.07	42.07	40.52
0.3	41.78	44.99	43.32
0.4	**42.08**	**45.31**	**43.64**

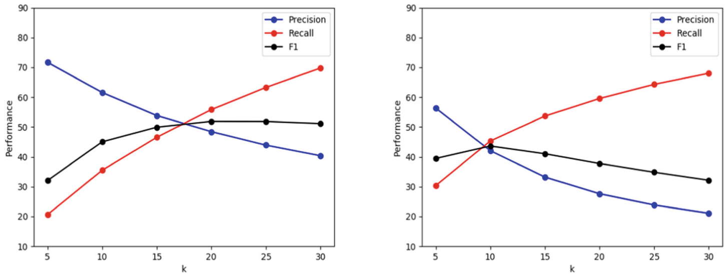

Fig. 2. The performance of Hulth2003 (threshold = 0.1) and Krapivin2009 (threshold = 0.4)

To verify the value of k yields the highest F1-measure mentioned above, we compared the F1-measure where the value of k was 5, 10, 15, 20, 25 and 30. The result shows that the F1-measure reaches the best when the value of k is 20 or 25 for Hulth2003 and 10 for Krapivin2009 (see Fig. 2). We find that the result of recall increases and the result of precision decreases when the value of k increases, which meets the fact.

The performance evaluation of keyphrase extraction can be divided into micro-statistical evaluation and macro-statistical evaluation. The micro one calculates the performance for each text first and then takes the average value. In comparison, the macro one statistics the result of extraction first and then calculates the performance at one time. We compared our model with Textrank [10], DegExt [21], k-core retention [22] and PositionRank [4] under the macro-statistical evaluation, where the value of k was 25 for Hulth2003 and 10 for Krapivin2009. The result shows that our model gets the best performance where the F1-measure reaches 51.85 for Hulth2003 and 43.64 for Krapivin2009 (see Table 5 and Fig. 3).

Table 5. The comparing performance with baseline methods

Dataset	DegExt	TextRank	K-core	PositionRank	Ours
Hulth2003	18.22	18.37	43.41	50.41	**51.85**
Krapivin2009	13.34	13.72	22.70	37.07	**43.64**

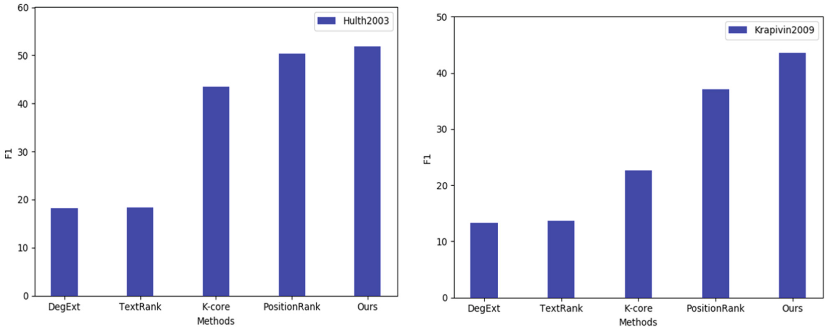

Fig. 3. The comparing performance with baseline methods

5 Conclusion

In this paper, we propose a new model that applies the three-way decision theory to graph-based keyphrase extraction model. In our model, we propose algorithms dividing the set of candidate phrases into the positive domain, the boundary domain and the negative domain depending on graph-based attributes, and combining candidate phrases in the positive domain and the boundary domain qualified by graph-based attributes and non-graph-based attributes to extract keyphrases. Experimental results show that our model can effectively improve the extraction accuracy compared with baseline methods. In future work, we will do more experiments to prove the performance of keyphrase extraction.

Acknowledgments. Authors would like to thank the anonymous reviewer for their critical and constructive comments and suggestions. This work was supported by National Natural Science Foundation of China (Grant No. 61976158). It was also supported by National Natural Science Foundation of China (Grant No. 61673301).

References

1. Zhang, W., Feng, W., Wang, J.Y.: Integrating semantic relatedness and words' intrinsic features for keyword extraction. In: Proceedings of the IJCAI, pp. 2225–2231. Morgan Kaufmann Publishers Inc., San Francisco (2013)
2. Gollapalli, S.D., Caragea, C.: Extracting keyphrases from research papers using citation networks. In: Proceedings of the AAAI, pp. 1629–1635. AAAI Press, Palo Alto (2014)
3. Gollapalli, S.D., Li, X.L., Yang, P.: Incorporating expert knowledge into keyphrase extraction. In: Proceedings of the AAAI, pp. 3180–3187. AAAI Press, Palo Alto (2017)
4. Florescu, C., Caragea, C.: A position-biased PageRank algorithm for keyphrase extraction, pp. 4923–4924. AAAI (2017)
5. Meng, R., Zhao, S.Q., Han, S.G., He, D.Q., Brusilovsky, P., Chi, Y.: Deep keyphrase generation. In: Proceedings of the ACL, pp. 582–592. ACL, Stroudsburg (2017)
6. Sterckx, L., Demeester, T., Deleu, J., Develder, C.: Topical word importance for fast keyphrase extraction. In: Proceedings of the WWW, pp. 121–122. ACM, New York (2015)
7. Sterckx, L., Caragea, C., Demeester, T., Develder, C.: Supervised keyphrase extraction as positive unlabeled learning. In: Proceedings of the EMNLP, pp. 1924–1929. ACL, Stroudsburg (2016)
8. Camacho, J.E.P., Ledeneva, Y., Hernández, R.A.G.: Comparison of automatic keyphrase extraction systems in scientific papers. Res. Comput. Sci. **115**, 181–191 (2016)
9. Salton, G., Buckley, C.: Term-Weighting approaches in automatic text retrieval. Inf. Process. Manag. **24**(5), 513–523 (1988)
10. Mihalcea, R., Tarau, P.: TextRank: bringing order into texts. In: Proceedings of the 2004 Conference on EMNLP, pp. 404–411. ACL (2004)
11. Pawlak, Z.: Rough sets. Int. J. Comput. Inform. Sci. **11**, 341–356 (1982)
12. Yao, Y.Y.: Three-way decisions with probabilistic rough sets. Inform. Sci. **180**, 341–353 (2010)
13. Miao, D.Q., Wei, Z.H., Wang, R.Z., Zhao, C.R., Chen, Y.M., Zhang, X.Y.: Uncertainty Analysis in Granular Computing. Science Press, Beijing (2019)

14. Zhang, Y.B., Miao, D.Q., Wang, J.Q., Zhang, Z.F.: A cost-sensitive three-way combination technique for ensemble learning in sentiment classification. Int. J. Approx. Reason. **105**, 85–97 (2019)
15. Toutanova, K., Klein, D., Manning, C.D., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: Proceedings of the ACL, pp. 173–180. ACL, Stroudsburg (2003)
16. Lovins, J.B.: Development of a stemming algorithm. Mech. Transl. Comput. Linguist. **11**, 22–31 (1968)
17. Bird, S.: NLTK: the natural language toolkit. In: Proceedings of the COLING/ACL on Interactive Presentation Sessions, pp. 69–72. ACL, Stroudsburg (2006)
18. Duari, S., Bhatnagar, V.: sCAKE: semantic connectivity aware keyword extraction. Inf. Sci. **477**, 100–117 (2019)
19. Cohen, J.: Trusses: cohesive subgraphs for social network analysis. National Security Agency Technical Report (2008)
20. Kaur, S., Saxena, R., Bhatnagar, V.: Leveraging hierarchy and community structure for determining influencers in networks. In: Bellatreche, L., Chakravarthy, S. (eds.) DaWaK 2017. LNCS, vol. 10440, pp. 383–390. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-64283-3_28
21. Litvak, M., Last, M., Aizenman, H., Gobits, I., Kandel, A.: DegExt—a language-independent graph-based keyphrase extractor. In: Mugellini, E., Szczepaniak, P.S., Pettenati, M.C., Sokhn, M. (eds.) Advances in Intelligent Web Mastering – 3, pp. 121–130. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-18029-3_13
22. Rousseau, F., Vazirgiannis, M.: Main core retention on graph-of-words for single-document keyword extraction. In: Hanbury, A., Kazai, G., Rauber, A., Fuhr, N. (eds.) ECIR 2015. LNCS, vol. 9022, pp. 382–393. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16354-3_42