A Review of Graph-Based Extractive Text Summarization Models

Abdulkadir Abubakar Bichi[1](✉), Ruhaidah Samsudin[1], Rohayanti Hassan[1], and Khalil Almekhlafi[2]

[1] School of Computing, Universiti Teknologi Malaysia, Johor Bahru, Malaysia
{ruhaidah,rohayanti}@utm.my

[2] MIS Department, College of Business Administration - Yanbu, Taibah University, Yanbu 42353, Saudi Arabia

Abstract. The amount of text data is continuously increasing both at online and offline storage, that makes is difficult for people to read across and find the desired information within a possible available time. This necessitate the use of technique such as automatic text summarization. A text summary is the briefer form of the original text, in which the principal document message is preserved. Many approaches and algorithms have been proposed for automatic text summarization including; supervised machine learning, clustering, graph-based and lexical chain, among others. This paper presents a review of various graph-based automatic text summarization models.

Keywords: Natural languages processing · Text mining · Graph approaches

1 Introduction

The volume and quantity of documents available today both on the internet and offline storage, make it difficult and time consuming for one to read across and find the required information. This necessitate the used of computing methods to the problem, and the automatic text summarization (ATS), was found to be most promising option [1]. A text summary is a briefer form of the original text, in which the principal document message is preserved [2]. The ATS is classified using different criteria; based on number of input files, generated output, purpose and context. Based on number of input files, ATS is categorized into: single and multi-document ATS. The single or mono-document summarization generate separate summary for each individual document file while multi-document summarization generate one summary for many related documents [3].

The ATS is also classified based on the generated output into; extractive and abstractive ATS. The extractive type is achieved by choosing the vital and most informative document sentences and rearranged them according to their original index [4]. The abstractive on the other hand, involves intense content reformatting, paraphrasing and rewriting the text in entirely different words [5]. The process is complex and more challenging as the deep analysis of linguistic features required [6]. The ATS is further classified based on purpose into; query-focus and generic. In query-focus, a summary is

F. Saeed et al. (Eds.): IRICT 2020, LNDECT 72, pp. 439–448, 2021.
https://doi.org/10.1007/978-3-030-70713-2_41

generated based on the user biasness [7], usually the system considers the query words or phrases in scoring the document sentences. In contrast, the generic type includes all the documents subtopic [8], and generate unbiased summary regardless of the user preference. More so, ATS is classified based on context into indicative and informative. The indicative summary is less detail summary, which contains only the key outlines of the source document [9], whereas the informative summary cover in depth all topics of the original text, which in most cases are enough for major analysis without referring to the original source [10].

2 Extractive Text Summarization

The extractive ATS, is generated by selecting the salient sentences of a document and rearranging them together. Formally, for any given document D, let the set (D) represent a set containing all sentences in D, and $\mathscr{L}$ be an integer value such that $\mathscr{L} \leq |\mathscr{S}|/ 2$. The extractive summary is defined as a subset M of the set (D), $\mathrm{M} \subset \mathscr{S}(D)$ and $|\mathrm{M}| \leq \mathscr{L}$, where |M| is represent the total number of sentences in the subset M, as shows in Fig. 1. The Summary is a subset of the document sentences form by removing the redundant and unnecessary sentences from the original set, which their absent will not affect the fundamental documents concept.

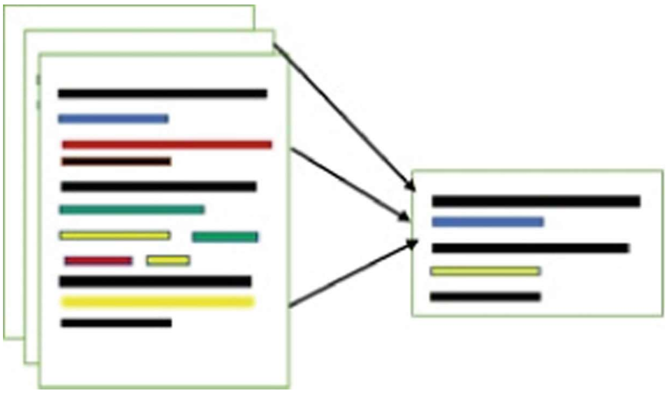

Fig. 1. Extractive summarization

The first model of extractive ATS has been proposed for more than 60 years [11]. But the field still remains one of the most challengeable area of research in the field of NLP [12]. The earlier techniques of ATS involve the use of text heuristic features like the term's frequencies [11], sentences position [13], and title words [14] among others. Far along, other techniques were used for extractive ATS, including clustering method, graph method, machine learning and lexical chain.

3 Graph-Based ATS Models

Graph-based ATS models are based on the concept of mathematical graph theory, in the model a graph node is drawn for each sentence in the document and edge is drawn for any two sentences with some similarity, as shows in Fig. 2.

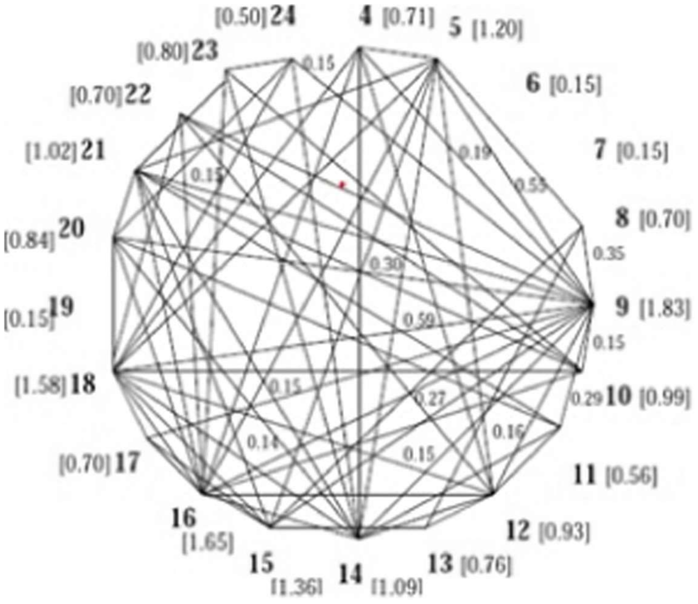

Fig. 2. Example of text graph [15]

In the graph-based ATS method, sentences recommend other similar sentences and the importance of sentence depend on the importance of the sentences that recommend it. Regardless of the algorithm or model used Mihalcea and Tarau [15], outline the following general steps for graph-based extractive text summarization:

1. Choose the text unit to be represented as the graph vertices.
2. Determine the relations between the text unit and use it to draw the graph edges.
3. Iterate the ranking algorithm to achieve convergence.
4. Arrange vertices according to their scores and select top ranked as summary.

Various models have been proposed for graph-based ATS, as discussed in the following subsections.

3.1 Static Graph-Based Model

The static graph-based models are based on the concept of earlier graph ranking algorithms developed for other applications, such as Hyperlink-Induced Topic Search (HITS) algorithm [16], Positional Power Function algorithm [17] and PageRank algorithm [18]. TextRank algorithm [15], was the first graph-based ATS algorithm, based on the concept of PageRank algorithm. The algorithm represents text sentences as graph vertices and graph edge is draw between any two sentences with some similarity. A words' overlap is used to determine the similarity between sentences. Unlike the original PageRank algorithm for web analysis that used digraph, the TextRank model used undirected graph and a weight wij is introduced to indicates the degree of causality between sentences i and j. The TextRank algorithm ranks sentence using modified PageRank, as shows in Eq. 1.

$$WS(Vi) = (1 - d) + d * \sum_{Vj \in In(Vi)} \frac{wji}{\sum_{Vk \in Out(Vj)} wkj} WS(Vj) \tag{1}$$

As almost the same time, LexRank algorithm [19], was proposed by different group. It does same function as TextRank algorithm but uses cosine similarity of tf-idf vectors, to determine the similarity of sentences, as show in Eq. 3.

$$P(u) = \frac{d}{N} + (1-d)\sum\nolimits_{V\in adj|u|} \frac{idf - \mathrm{mod}\,ified - \cos ine(u,v)}{\sum\nolimits_{z\in adj|v|} idf - \mathrm{mod}\,ified - \cos ine(z,v)} P(v) \tag{2}$$

The LexRank support multi-document summarization and it use other text features like sentence length and position for scoring sentences. A research by Mallick, Das [20], modified TextRank algorithm by using inverse sentence frequency (is) based cosine similarity for the similarity measurement. Similarly, Elbarougy, Behery [21], modified TextRank algorithm for Arabic language ATS by using the value of noun count in the document sentences as additional sentences scores. In the same way, Sikder, Hossain [22], modified PageRank for summarization of Bengali text; by including others sentences features like sentence position and length in the ranking.

In a research by Woloszyn, Machado [23], cosine-similarity is combined with keyword-similarity for sentence scoring. And a graph-based ATS algorithm by Natesh, Balekuttira [24], used noun position for scoring sentence, where the inverse of distance between two nouns in a sentence is used to determine the sentences weights. Alzuhair and Al-Dhelaan [25], proposed Graph-based ATS hybrid ranking algorithm, by combining PageRank algorithm with HITS algorithm using harmonic mean. Barrios, López [26], combined TextRank algorithm with BM25 ranking algorithm for efficient ranking of sentences. Mussina, Aubakirov [27] proposed symmetric ranking in graph-based ATS; where a sentence is ranked symmetrically using the length of the longest common substring in the sentence.

3.2 Dynamic Graph-Based Model

The previously discussed ATS algorithms like TextRank and LexRank algorithms work on static graph model. Ziheng [28], proposed the used of evolutionary graph model for ATS, the model consider the arrival of sentences into the documents. The sentences are arranged in chronological order from first to last, and modelled using a directed graph. The Author ranks the documents sentences by considering both their similarities with other sentences in the cluster and their similarities to the previously selected sentences in the documents using modified MMR re-ranker equation [29], as shows in Eq. 3.

$$MMR_{\mathrm{mod}\ 2} = \underset{si\in R-S}{\arg\max} [\lambda.Score(Si) + (1-\lambda).sim(s_i, Q) - \delta \sum_{s_k \in S} sim(s_i, s_k) - \gamma. \sum_{s_j \in P} sim(s_i, s_j) \tag{3}$$

Where Score(s) is the score of sentence s, is called the penalty factor introduced to check the redundancy. Gallo, Popelínský [30], enhanced the concept of timestamps graph with time abstraction using a signal function. The method further improved the quality of scoring by selecting the best pattern and discarding the irrelevant edges, as shows in Eq. 4.

$$Score(s) = Score_{\sin gle}(s) + Score_{multi}(s) \tag{4}$$

3.3 Graph Pruning-Based Model

The graph pruning-based models of extractive summarization reduces the number of graph nodes and edges by pruning unnecessary graph edges and vertices, thus reducing the time of the graph search. Patil and Brazdil [31], modified LexRank by pruning the graph before applying the ranking algorithm. Miranda-Jiménez, Gelbukh [32], developed a model for single-document summarization using the concept of graph pruning based on HITS ranking algorithm. Similarly, Al-Khassawneh, Salim [33], used graph triangle method for pruning graph in extractive text summarization.

More so, a research by Hark and Karcı [34], introduced Karcı method, a graph entropy algorithm to filter out irrelevant graph vertices and select most informative sentences in each paragraph, for multi-document summarization. Likewise, the used of maximum independent set method to filter out less relevant graph nodes was proposed by Uçkan and Karcı [35]. The pruning graph models reduces the graph searching time but has additional time of graph pruning, thus the overall process time is not improved in the model but the accuracy of ranking and selection is better in smaller graphs.

3.4 Hypergraph-Based Model

Hypergraph allows one edge called hypergraph incidence to connect more than 2 vertices, thus enable more advance relations between the graph vertices. Wang, Wei [36], Wang, Li [37], proposed a model for query-focus text summarization based on the concept of hypergraph. The hypergraph model was extended for multi-document ATS using vertex-reinforced random walk [38]. Similarly, Lierde and Chow [39], applied clustering technique to hypergraph model for query-focus text summarization; by first grouping the document into clusters and then construct a hypergraph for each cluster.

3.5 Affinity Graph-Based Model

The concept of affinity graph involves grouping nodes representing similar objects from different graphs. Wan and Yang [40], used the concept of affinity graph for multi-document summarization by utilizing both inter and intra documents diversity to determine the similarity between sentences. Another research applied random walk algorithm to affinity graph-based ATS [41]. Similarly, Hu, He [42], proposed affinity model with manifold ranking and Kanitha, Mubarak [43], scores sentences using the sum of their affinity weights for extractive ATS of Malayalam language.

3.6 Semantic Graph-Based Model

The semantic graph-based model used a semantic similarity measure to determine relations between document sentences. Ullah and Al Islam [44], utilized the idea of semantic graph for extractive text summarization by first extracting the Predicate Argument Structure (PAS) of sentences; the sematic similarity between sentences is measures using their PAS. The graph vertices in the approach are ranks using PageRank algorithm and re-rank using MMR algorithm to minimize redundancies. Sevilla, Fernández-Isabel [45], proposed hybrid approach for semantic similarity graph using both knowledge source and linguistic features. Similarly, Han, Lv [46], used Frame-Net and word embedding to measure sematic similarity in semantic graph model for extractive text summarization. Mohamed and Oussalah [47], introduced semantic graph-based ATS framework that support both single and multi-document generic summarization; the semantic similarity is determine using both SRL and Wikipedia knowledge.

3.7 Multigraph-Based Model

Multigraph model allows more than one edges between two adjacent vertices. The number of edges indicates the strength of the connection, which is regarded as a weight of the vertex. AlZahir, Fatima [48], used multigraph graph model to represent text for extractive text summarization. In the model an edge is drawn for every two similar words in the adjacent sentences, which later represented using a symmetric matrix.

4 Discussion

The graph-based approach uses the graph structure to determine relation and ranks the documents sentences. The most common method to determine the degree of causality between sentences in the approach is similarity measure. The technique has been implemented for diverse type of summarizations, including single-document, multi-document, generic and query-specific. As a typical unsupervised technique, the method does not require training with annotated data, therefore less expensive to implement. The majority of the graph-based ATS algorithms do not depend on the semantic meaning of words, therefore easily applied to many languages. The method considers the relation of sentence with all other sentences in the documents from all positions for a final ranking; therefore, generate summary which are readable and coherent. Like the heuristic features-based and clustering methods, the graph-based algorithms are simple to implement.

The research based the taxonomy on graph structure and classified the models into: static graph-based, dynamic graph-based, graph pruning-based, hypergraph-based, affinity graph-based, semantic graph-based, and multigraph-based models. The static graph-based models are the initial but still effective and most commonly used models. In the model undirected weighted graph is used to represent text. A similarity measure is used to determine the weights of the graph edges. The most common similarity measure used in the algorithms is cosine similarity of tf-idf vectors. Some algorithms combined more than one similarity measures using either arithmetic mean or simple harmonic mean, such combination slow the models but gives more accurate scores. Some model like LexRank combined the similarity measures with other sentences features for scoring; but such features has no any significant effect. The efficiency of an algorithm in the model is largely depends on the accuracy of the similarity calculation and ranking function. The static graph-based models are popular for their simplicity, ease of implementation and fast computation. The model has been successfully applied to both single-document and multi-document summarization and it is good in resource utilization. The dynamic graph-based model on the contrary, considers the time of sentences arrival into the document in modelling the graph. Therefore, the model used directed graph to represent the text sentences. The dynamic graph-based models generate summary with good readability but the models are usually led to a slow and complex graph representation. Like the static graph-based model, the approach is good for both single-document and multi-document summarization.

The graph pruning methods like triangle counting and graph entropy methods reduce the number of the graph nodes, thus improved the efficiency and accuracy of the graph search. But the technique suffered with the addition time complexity of pruning the graph. The model is good for generic extractive text summarization and the low number of the graph vertices improve the efficiency of the scoring and selection of sentences. And the model has an advantage of generating summaries with less redundancies. The resource utilization in the approach can be minimized using some implementation techniques like dynamic programming. Similarly, the affinity graph-based model improves

Table 1. Comparison of various ATS graph-based models

Model	Similarity measure	Language dependency	Strengths	Weakness
Static graph-based	Lexical	No	Simple implementation fast computation, language independent	Less readability
Dynamic graph-based	Lexical	No	Coherency good readability, language independent	Additional computing time
Graph pruning-based	Lexical	No	More accurate scoring due to small size of the graph, language independent	Additional computing time
Hypergraph-based	Lexical	No	More accurate similarity calculation, language independent	Applied only for query-focus summarization
Affinity graph-based	Lexical	No	High coverage, language independent	Slow computation, poor readability
Semantic graph-based	Semantic	Yes	Good similarity scoring	Requires external knowledge source, language dependent
Multigraph-based	Lexical	No	Fast computation, language independent	Less accurate scoring

the quality of generated summary by sourcing information from other document; but the model also has high computing time and resource utilization compares to original static graph-based model. The model exploits the technique of global voting and recommendation by considering the sentences resemblance with sentences from other documents on similar topics, thus makes the ranking process of text sentences more accurate. The model is especially good for multi-document extractive summarization, in which many documents involves in the ranking and selection process and the generated summaries are highly informative. Likewise, the semantic graph-based models have more accurate similarity calculation, but the use of external database make the model slower and language dependent. The semantic similarity used by the model required linguistic tools and grammar of a particular language, thus make an algorithm proposed for one language very difficult to be modified for another language. On the other hand, hypergraph-based model has limited application, as it only used for query-focus summarization. But the process of determining the similarity in the model is powerful as it can group more than two sentences using hypergraph incidence. The different features of the graph-based model for extractive text summarization are analyzed in Table 1.

5 Conclusion

The field of ATS has been studied for more than 60 years, but still remain of one the most challengeable areas in natural language processing and information retrieval. There are many approaches for ATS but graph-based are prefer by many, for their less cost and language independency. The graph-based models are classified into: static graph-based, dynamic graph-based, graph pruning-based, hypergraph-based, affinity graph-based, semantic graph-based, and multigraph-based models. All the model has their pros and cons; a choice of a model depends on the human language and domain of summarization.

References

1. Aries, A., Zegour, D.E., Hidouci, W.K.: Automatic text summarization: What has been done and what has to be done. arXiv:1904.00688v1 [cs.CL] 1 (2019)
2. Narayan, S., Cohen, S.B., Lapata, M.: Ranking sentences for extractive summarizationwith reinforcement learning. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 1747–1759 (2018)
3. Cai, X., Li, W.: Ranking through clustering: an integrated approach to multi-document summarization. IEEE Trans. Audio Speech Lang. Process. **21**(7), 1424–1433 (2013)
4. Aker, A.: Entity Type Modeling for Multi-Document Summarization: Generating Descriptive Summaries of Geo-Located Entities. A thesis submitted in fulfilment of requirements for the degree of Doctor of Philosophy to Department of Computer Science University of Sheffield (2013)
5. Wan, X.: Using only cross-document relationships for both generic and topic-focused multi-document summarizations. Inf. Retrieval **11**(1), 25–49 (2008)
6. Khan, A., Salim, N.: A review on abstractive summarization methods. J. Theor. Appl. Inform. Technol. **59**(1), 64–72 (2014)

7. Zhong, S.-h., et al.: Query-oriented unsupervised multi-document summarization via deep learning model. Expert Syst. Appl. **42**(21), 8146–8155 (2015)
8. Gong, Y., Liu, X.: Generic text summarization using relevance measure and latent semantic analysis. In: Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (2001)
9. Narayan, S., Cohen, S.B., Lapata, M.: What is this article about? extreme summarization with topic-aware convolutional neural networks. J. Articial Intell. Res. **66**, 243–278 (2019)
10. Vollmer, M., et al.: Informative summarization of numeric data. In: 31st International Conference on Scientific and Statistical Database Management (SSDBM 2019). Santa Cruz, CA, USA (2019)
11. Luhn, H.P.: The automatic creation of literature abstracts. IBM J. Res. Dev. **2**(2), 159–165 (1958)
12. Rezaei, H., et al.: Features in Extractive Supervised Single-Document Summarization: Case of Persian News. arXiv:1909.02776v2 [cs.CL] 9 (2019)
13. Baxendale, P.B.: Machine-made index for technical literature: an experiment. IBM J. Res. Dev. **2**(4), 354–361(1958)
14. Edmundson, H.P.: New methods in automatic extracting. J. ACM **16**(2), 264–285 (1969)
15. Mihalcea, R., Tarau, P.: Textrank: bringing order into texts. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 404–411 (2004)
16. Kleinberg, J.M.: Authoritative sources in a hyper linked environment. J. ACM **46**(5), 604–632 (1999)
17. Herings, P.J., Van der Laan, G., Talman, D.: Measuring the power of nodes in digraphs. Technicalreport, TinbergenInstitute (2001)
18. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Comput. Netw. ISDN Syst. **30**(1–7), 107-117 (1998)
19. Erkan, G., Radev, D.R.: LexRank: graph-based lexical centrality as salience in text summarization. J. Artif. Intell. Res. **22**, 457–479 (2004)
20. Mallick, C., et al.: Graph-based text summarization using modified textrank. In: Soft Computing in Data Analytics, Advances in Intelligent Systems and Computing (2018)
21. Elbarougy, R., Behery, G., Khatib, A.E.: Extractive arabic text summarization using modified pagerank algorithm. Egyptian Informatics Journal (2019)
22. Sikder, R., Hossain, M.M., Robi, F.M.R.H.: Automatic text summarization for bengali language including grammatical analysis. Int. J. Sci. Technol. Res. **8**(6), 288–292 (2019)
23. Woloszyn, V., et al.: Modeling Comprehending and Summarizingtextual Content by Graphs. arXiv:1807.00303v1 [cs.CL] (2018)
24. Natesh, A.A., Balekuttira, S.T., Patil, A.P.: Graph based approach for automatic text summarization. Int. J. Adv. Res. Comput. Commun. Eng. **5**(2), 6–9 (2016)
25. Alzuhair, A., Al-Dhelaan, M.: An approach for combining multiple weighting schemes and ranking methods in graph-based multi-document summarization. IEEE Access **7**, 120375–120386 (2019)
26. Barrios, F., et al.: Variations of the Similarity Function of TextRank for Automated Summarization. arXiv:1602.03606 [cs.CL], pp. 65–72 (2016)
27. Mussina, A., Aubakirov, S., Trigo, P.: Automatic document summarization based on statistical information. In: 7th International Conference on Data Science, Technology and Applications (DATA 2018) (2018)
28. Ziheng, L.: Graph-based methods for automatic text summarization. In: School of Computing, National University of Singapore (2007)
29. Carbonell, J., Goldstein, J.: The use of MMR, diversity-based reranking for reordering documents and producing summaries. In: 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (1998)

30. Gallo, M., Popelínský, L., Vaculík, K.: To text summarization by dynamic graph mining. CEUR Workshop Proc. **2203**, 28–34 (2018)
31. Patil, K., Brazdil, P.: Text summarization: using centrality in the pathfinder network. In: IADIS International Conference Applied Computing (2007)
32. Miranda-Jiménez, S., Gelbukh, A., Sidorov, G.: Summarizing conceptual graphs for automatic summarization task. In: Pfeiffer, H.D., et al. (eds) Conceptual Structures for STEM Research and Education. ICCS 2013. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg (2013)
33. Al-Khassawneh, Y.A., Salim, N., Jarrah, M.: Improving triangle-graph based text summarization using hybrid similarity function. Indian Journal of Science and Technology, vol. 10, no. 8 (2017)
34. Hark, C., Karci, A.: Karci summarization: a simple and effective approach for automatic text summarization using Karci entropy. Inform. Process. Manag. **57**(3), 102187 2020
35. Uçkan, T., Karci, A.: Extractive multi-document text summarization based on graph independent sets. Egypt. Inform. J. **21**(3), 145–157 (2020)
36. Wang, W., et al.: Hypersum: hypergraph based semi-supervised sentence ranking for query-oriented summarization. In: 18th ACM Conference on Information and Knowledge Management. ACM (2009)
37. Wang, W., et al.: Exploring hypergraph-based semi-supervised ranking for query-oriented summarization. Inf. Sci. **237**, 271–286 (2013)
38. Xiong, S., Ji, D.: Query-focused multi-document summarization using hypergraph-based ranking. Int. J. Inform. Process. Manag. **52**(4), 670–681 (2016)
39. Lierde, H.V., Chow, T.W.S.: Query-oriented text summarization based on hypergraph transversals. Inform. Process. Manag. **56**(4), 1317–1338 (2019)
40. Wan, X., Yang, J.: Improved affinity graph based multi-document summarization. In: Human Language Technology Conference of NAACL (2006)
41. Wang, K., et al.: Affinity-preserving random walk for multi-document summarization. In: 2017 Conference on Empirical Methods in Natural Language Processing. Copenhagen, Denmark: Association for Computational Linguistics (2017)
42. Hu, P., He, J., Zhang, Y.: Graph-based query-focused multi-document summarization using improved affinity graph. In: Zhang, W.M., Zhang, S. (eds) Knowledge Science, Engineering and Management. KSEM 2015. Lecture Notes in Computer Science. Springer, Cham (2015)
43. Kanitha, D.K., Mubarak, D.M.N., Shanavas, S.A.: Malayalam text summarization using graph based method. Int. J. Comput. Sci. Inform. Technol. **9**(2), 40–44 (2018)
44. Ullah, S., Al Islam, A.B.M.A.: A framework for extractive text summarization using semantic graph based approach. In: ACM International Conference Proceeding Series (2019)
45. Sevilla, A.F.G., Fernández-Isabel, A., Díaz, A.: Enriched semantic graphs for extractive text summarization. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pp. 217–226 (2016)
46. Han, X., et al.: Text summarization using framenet-based semantic graph model. Scientific Programming. Hindawi Publishing Corporation (2016)
47. Mohamed, M., Oussalah, M.: SRL-ESA-TextSum: a text summarization approach based on semantic role labeling and explicit semantic analysis. Inf. Process. Manage. **56**(4), 1356–1372 (2019)
48. AlZahir, S., Fatima, Q., Cenek, M.: New graph-based text summarization method. In: IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM) (2015)