Exploring Graph Bushy Paths to Improve Statistical Multilingual Automatic Text Summarization

Abdelkrime Aries, Djamel Eddine Zegour, Walid Khaled Hidouci

Exploring Graph Bushy Paths to Improve Statistical Multilingual Automatic Text Summarization

Abdelkrime Aries *, Djamel Eddine Zegour, and Walid Khaled Hidouci

École nationale Supérieure d'Informatique (ESI, ex. INI), Algiers, Algeria
{ab_aries, d_zegour, w_hidouci}@esi.dz

Abstract. Statistical extractive summarization is one of the most exploited approach in automatic text summarization due to its generation speed, implementation easiness and multilingual property. We want to improve statistical sentence scoring by exploring a simple, yet powerful, property of graphs called bushy paths represented by the number of node's neighbors. A graph of similarities is constructed in order to select candidate sentences. Statistical features such as sentence position, sentence length, term frequency and sentences similarities are used to get a primary score for each candidate sentence. The graph is used again to enhance the primary score by using bushy paths property. Also, we tried to exploit the graph in order to enhance summary's coherence. We experimented our method using MultiLing'15 workshop's corpora for multilingual single document summarization. Using graph properties can improve statistical scoring without loosing the multilingualism of the method.

1 Introduction

Summarization is an effective way to handle huge amount of information, especially in this century. Its main purpose is to keep the most important information and get rid of redundant one. Therefore, a summary must be short, representative, readable and not redundant. A summary has to present the main points and key details of the input document. Information redundancy is not a good aspect of a summary, since it prevents adding new useful details especially when that summary's size is limited. Readability is another important aspect of a summary; this later must be coherent and grammatically correct. Manual summarization is a very expensive task either in time or cost, this is why researches have been conducted since 1958 [15] to find a perfect automatic text summarization (ATS) system.

Recently, multilingual content started to grow, leading to a huge need for more adaptable methods for languages other than English. As a result, multilingual ATS began to receive more attention from research community [9, 10, 12].

* Abdelkrime Aries (✉) École nationale Supérieure d'Informatique (ESI, ex. INI), BP 68M Oued Smar, 16309, El-Harrach, Alger, Algeria. **E-mail:** ab_aries@esi.dz, **ORCID:** 0000-0002-3042-1477

A truly multilingual method must not rely on any language-dependent resource, either toolkits or corpora. ATS methods categorized in linguistic approach or machine learning need a lot of changes to handle another language. The problem with linguistic approach is its heavy dependency on languages and the lack of NLP toolkits. In case of machine learning (ML), using a corpus for training can cause the system to be more dependent on the corpus's language. The two remaining approaches are statistical and graph-based. Beside their multilingualism, statistical methods are fast and easy to implement. Even if they do not use deep linguistic techniques, they still give good results. As for graph-based approach, it is not as widely used as the statistical one. Its main factor is the coherence relations between sentences, where the score of a sentence is calculated based on its neighbors. In our work, we want to exploit the graph bushy paths property in order to improve statistical scoring in multilingual ATS. To place our work among others of the same domain, here is a summary of our contribution:

- Proposing a multilingual method based on statistical features avoiding their weights attribution when combining them linearly.
- Investigating the impact of using graph properties (here, bushy paths) to improve the original scores.
- Using the graph to select candidate sentences relying on the assumption that a sentence without either much or strong relations with its neighbors will not be considered for scoring. This will help decrease the number of sentences to be processed, resulting in a faster generation.
- Testing the impact of extraction methods on the summarization quality.

The remainder of this article is divided as follows. Section 2 discusses related works, namely those based on statistical or/and graph-based approaches. Section 3 provides the details of our method starting from preprocessing, through graph construction and scoring, ending with sentences extraction and summary construction. Section 4 presents the experimental evaluation of our method in the context of multilingual single document summarization. Finally, section 5 concludes the work carried in this research and provides future insights.

2 Related Works

Statistical approach was the first exploited for ATS and still widely used nowadays. It is based on statistical features which are used to judge the importance of a sentence. The first feature ever used for ATS is term frequency (tf) [15] which scores a sentence based on its terms distribution in the original document. Very high frequency terms can be related to the domain and not the subject, this is why the author fixes a high threshold to eliminate such terms. Another solution is by combining tf with inverse document frequency to get $tf-idf$ [19]. Position is another feature allowing to capture the importance of a sentence, either by its words positions [15], by its position in the original document [4, 6] or by its position in a paragraph [8]. Sentence position in the document is based on the assumption that first and last sentences tend to be more important than others

[6]. To score a sentence using this feature, researchers tested with many different formulas [17, 8]. Sentence length feature is used to penalize short sentences with a size less than a given threshold [13]. Other length-based formulas can be found in [17, 8]. Mostly, these features are combined linearly into one score representing how pertinent a sentence is. The weight of each feature is fixed manually, or estimated using optimization or machine learning.

Graph-based approach transforms document sentences into a graph using a similarity measure. Then, this structure is used to calculate the importance of each sentence. In [20], a graph of similarities is constructed from document paragraphs, where similarities below a given threshold are ignored. The authors define a feature called bushiness which is the number of a node's connections. The most scored paragraphs in term of bushiness are extracted to form a summary. Node's bushiness can be used to score sentences rather than paragraphs, to be combined with other statistical features scores [8]. Likewise, we use this feature to score sentences, but we use it to enhance statistical scoring rather than consider it as one. Iterative scoring using graphs is very popular among graph-based ATS methods; TextRank [16] and LexRank [7] are the most known ones. They use a modified version of PageRank [5] in order to score sentences. LexRank incorporates $tf - idf$ into similarity calculation, while LexRank uses simple cosine similarity. When multiple documents are summarized, it is preferable to include temporal information to the score favoring recent documents [21]. To capture the impact of documents on sentences in multi-document ATS, sentence to document dimension can be used [22]. The author incorporates sentence position into the score to distinguish sentences from each other when they belong to the same document. Another work aiming to fuse statistical features with graph-based approach is iSpreadRank [23]. The method is based on activation theory [18] explaining the cognitive process of human comprehension. Each sentence is scored using some features (centroid, position, and first-sentence overlap), then these scores are spread from a sentence to its neighbors iteratively until equilibrium. The ranking in iterative graph-based ATS may never converge. Also, the ranking convergence depends on the initial scores and the damping factor.

3 Method Overview

3.1 Preprocessing

Preprocessing phase consists of: sentence segmentation, words tokenization, stop words removal, and words stemming. Sentence segmentation is the task of splitting a text into several sentences using punctuation. Sometimes it is difficult to detect sentence boundaries due to punctuation use in abbreviations (mr., ms., etc.). Words tokenization divides a text into words, which is not simple in some languages such as Japanese where the words are attached. Stop words removal aims to get rid of subject-independent words such as pronouns, prepositions, etc. Finally, stemming is the task of prefixes, suffixes and infixes removal; which

helps associating variants of the same term to a common root form. In our work, we use an open source tool (LangPi[1]) which affords these tasks for 40 languages.

3.2 Candidate Sentences Generation

The task of finding candidate sentences comes to find those less probable to be included in the summary and exclude them. That is, the stronger a sentence connection to other sentences in the document, the more it is important to be kept. This connection is represented, in our case, by cosine similarity between sentences. A graph of similarities $G(V, E)$ is constructed where each sentence is represented by a node, and arcs represent the similarities between the sentences. This graph can be reduced to keep just significant nodes and arcs.

Using a given similarity threshold, we decide the most important neighbors $MImpN(v_i)$ to a node v_i. We define an insignificant node v_i as the one with a sum of similarities less than $\frac{1}{|MImpN(v_i)|}$ where $|MImpN(v_i)|$ is the number of its important neighbors. When the number of important neighbors increases, the chance of a node to be kept as a candidate sentence will be higher. In this case, many factors determine if a node is good enough to be included in the summary. The first factor is the similarity threshold's value; When it is high, the number of important neighbors will be lower and thus a low chance. The second one is the number of connections; When it has a lot of connections, the sum of similarities (even if they are low) has a great chance to exceed the significance threshold. The third one is the similarities strength; If a sentence has high similarities with its neighbors, their sum can be higher enough to make it significant.

Once we have deleted the insignificant nodes, we want to remove weak arcs as well. We define a weak arc (s_i, s_j) from a node s_i as the one with a value (similarity) less than $\frac{threshold}{|MImpN(v_i)|}$. In here, we try to distribute the threshold uniformly between the different arcs joined to each node. This will result in a very small value, especially when the node has many neighbors. We should point out that, using this method, a node can consider another as a neighbor while the second one does not.

3.3 Sentence Scoring

We want to attribute to each candidate sentence a score based on statistical features reflecting its pertinence towards the main topic of the input document. Our intention is to come up with a method totally independent from the input document's language and domain. This is why we choose four features: its similarity to other candidates, the terms it contains, its size and its position in the document. These features can be calculated easily for any language without the need of affording language-dependent resources except the preprocessing task which is simple, mostly.

A sentence is more likely to be included in the summary if it can reflect the input document's content. In our method, we just calculate the similarity

[1] LangPi: https://github.com/kariminf/langpi

between a candidate sentence s_i and the remaining candidates $C \backslash s_i$ as shown in (1).

$$Score_{sim}(s_i) = sim(s_i, C \backslash s_i) \tag{1}$$

Term frequency has been used since the first days of ATS [15]. It is a very good sign of sentence importance, especially when it is combined with inversed document frequency (*idf*) [19]. Unfortunately, we can't use *idf* since we want a domain-independent method. But, we can use a similar concept which is inverse sentence frequency isf, first used in [1]. Our $tf-isf$ based sentence score follows the same one used in [17] represented by the Euclidean normalization of $tf - isf$ values, but using sentence terms instead of document terms. Equation 2 represents the $tf - isf$ score of a sentence s_i given its words (terms) w_{ik}.

$$Score_{tf-isf}(s_i) = \sqrt{\sum_{w_{ik} \in s_i} tf - isf(w_{ik})^2} \tag{2}$$

Sentence size (length) is another feature used to calculate its importance in a document. Mostly, short sentences do not carry so much information, this is why they are omitted [13]. Also, we believe that long sentences are not a good fit to be included in a summary. This is justified by the fact that a summary must be short, informative and without redundancy. Pushing long sentences into a summary reduces the number of ideas that can be extracted from the original document. Furthermore, they can contain redundant or unimportant information reduced just by another type of language-dependent summarization approach called sentence compression. So, in our case, we prefer the shortest ones; The shortest they are, the most scored they will be. The two previous scores, namely similarity and $tf-isf$, will favor sentences similar to the main topic. This score (size), in the other hand, will help them select pertinent sentences but with the least possible size. It is calculated by, simply, dividing 1 on the sentence's size, as shown in (3).

$$Score_{size}(s_i) = \frac{1}{|s_i|} \tag{3}$$

The last score is sentence position which is a good feature, since important sentences tend to occur at first or at last [4, 6]. Equation 4 represents the position score which is proposed among others in [17], where $|D|$ is the number of sentences in the input document.

$$Score_{pos}(s_i) = \max(\frac{1}{i}, \frac{1}{|D| - i + 1}) \tag{4}$$

The previous scores can be considered as probabilities that a sentence belongs to the summary using a given feature, after normalizing them of course. In our case, we won't normalize these scores even if they exceed 1 since they will be used to reorder sentences and the denominator will be the same in all sentences thus

a constant. So, the overall satistical score ($Score_{stat}$), which is the probability that a sentence belongs to the summary using all features, can be calculated as the multiplication of all these scores assuming features independence.

A big number of neighbors is a good sign that a sentence is representative and discusses topics covered by many other sentences [20], so its score must be high. In contrast of [20], which scores a sentence by the number of its neighbors, our bushy paths score is based on amplifying the statistical score of a sentence by the number of its neighbors plus one. A sentence with a little statistical score can be given another chance to be included in the summary when it has many neighbors. Pertinence with the main topics is not the only aspect of sentence importance; Coherence with other sentences has proven to have very large impact on summary's quality. To combine these two aspects, we simply amplify the statistical score by the sentence's neighbors number as presented in (5).

$$Score_{bushy}(s_i) = Score_{stat}(s_i) * (1 + |\{(s_i, s_j) \in E\}|) \tag{5}$$

3.4 Extraction

After the sentences are scored, they are extracted to form a summary. Most works extract the most scored ones without any further processing. One of the problems resulting from that, is information redundancy; If two sentences are highly scored and similar, they risk to be included in the summary though they almost have the same information. Some works try to handle redundancy after scoring, by exploring the similarity between the candidate sentence and the last added one [3]. Similarly, we try to investigate the extraction phase impact on the generated summary's quality. This is why we use three simple extraction methods.

The first one (e_0) is used by most works: extract the first scored sentences till reaching summary's maximum size. This method does not manage redundancy and therefore it is expected to give the least precision score among the others. It, also, does not consider presenting the summary in a more coherent way. We tried to formulate this in (6), where *next* represents the sentence to be selected next; C is the list of candidate sentences; S is the list of summary sentences.

$$next_{e0} = \arg\max_i score(s_i) \text{ where } s_i \in C \backslash S \tag{6}$$

The second variant (e_1) is used in [3] by adding the most scored sentences which are not similar to the summary. The idea is to reorder sentences using their scores then each time we want to add a sentence to the summary, we must compare it to the last added one. This way, we will have a summary with diverse information, but no coherence is considered. The extraction method is illustrated in (7), where *sim* is the similarity function (cosine similarity, in our case); *last* is the last sentence added to the summary S; Th is a given similarity threshold.

$$next_{e1} = \arg\max_i score(s_i) \text{ where } s_i \in C \backslash S \text{ and } sim(s_i, last_{e1}) < Th \tag{7}$$

In the third one (e_2), we want to incorporate the graph into the extraction task. The graph affords the information about which sentence is similar to the other. We believe consecutive sentences must have some similarity to preserve coherence. When a sentence discusses an idea, the next one must have some shared terms with it. In this case, we extract the highest scored sentence, then the highest scored one among its neighbors. This will ensure some shared information between a sentence and the next, but does not prevent redundancy. Equation 8 is a formulation of this variant, where E is the set of graph's arcs.

$$next_{e2} = \arg\max_i score(s_i) \text{ where } (last_{e2}, s_i) \in E \tag{8}$$

4 Experiments

4.1 Metrics

To evaluate our method's performance, we apply ROUGE method which is used widely in ATS workshops. ROUGE (Recall-Oriented Understudy for Gisting Evaluation) [14] is an automatic evaluation method for ATS systems. Its principle is to compare automatic text summaries against human made abstracts based on words grams. It includes five measures: ROUGE-N, ROUGE-L, ROUGE-W, ROUGE-S and ROUGE-SU. In our evaluation, we use two metrics of ROUGE: ROUGE-1 and ROUGE-2, which are used in most ATS workshops. The original ROUGE package supports just English, this is why we use another implementation afforded by LangPi project, used in our preprocessing phase. This implementation supports all MultiLing'15 languages.

4.2 Corpora

MultiLing 2015 Multilingual Single-document Summarization (MSS) corpora contains two sets: training and testing. Each of them affords 30 documents extracted from Wikipedia featured documents for every language out of 38. The documents are UTF-8 encoded and packed with their respective human-generated summaries. For each summary, the number of characters which the generated summaries must not exceed is afforded.

4.3 Baseline System

We choose to use AllSummarizer[2] system as a baseline for these reasons:

- It is an available open source system.
- It is a multilingual system.
- It participated on MultiLing'15 for all the proposed languages.

[2] AllSummarizer: https://github.com/kariminf/allsummarizer

AllSummarizer_TCC (threshold clustering and classification) method [2, 3] starts by detecting the different topics in the input document using a simple clustering algorithm based on cosine similarity and a threshold. Then, using Bayes classification and a set of features, it learns the characteristics of each cluster. Each sentence is scored based on its probability to represent all these clusters. In our experiments, we use the optimal parameters fixed for each language in [3]. Also, we use testing corpus since the training one was used to fix these parameters and using it will favor AllSummarizer_TCC method.

4.4 Tests and Discussion

To generate summaries, first, we have to fix similarity threshold which is an important parameter in our method. If the threshold is too low, the number of candidate sentences will grow and also the number of arcs connecting a sentence to another. If it is too high, we will have a few number of candidate sentences which can be selected without even scoring. Also, the number of arcs connected to a sentence may drop to one and hence, the graph will not be necessary since its score remains the same. To avoid all of these scenarios, we use the mean between non null similarities as a threshold.

We generated summaries using the two scoring schemes (just statistical or bushy paths) and the different extraction methods ($e0$, $e1$ et $e2$) for the 38 languages of test corpus. Figure 1 represents ROUGE-1 recall for the 38 languages using Multiling'15 test corpus, compared to AllSummarizer_TCC method.

Lets discuss how much did the introduction of bushy paths improve statistical scoring for ATS. Applying bushy paths surpasses statistical method in 20 languages using extraction method $e0$, in 24 languages using $e1$, and in 25 languages using $e2$. We can say that introducing inter-sentences coherence (graph-based approach) has the potential to improve the quantity of information in summaries. Also, exploring graph structure to help extracting sentences (the case of $e2$) can help increase the quantity of information, although the method $e2$ is intended to increase the coherence between the extracted sentences and not the quantity of information.

Regarding the impact of extraction methods on the generated summary's recall using bushy paths, the application of $e1$ improves 24 languages summaries over $e0$. Using $e2$ improves 29 languages over $e0$ and 18 languages over $e1$. Also, extraction method $e1$ with bushy paths proves to be better compared to Allsumarizer_TCC which uses the same method (overcomes it in 20 languages). While using $e2$, bushy paths beats Allsumarizer_TCC in 18 languages, and beats statistical method with $e0$ in 36 languages except for Hungarian and Malay.

It is also important to point out that summaries quality varies from language to another, even if we use different methods. Languages such as Arabic and Hebrew, which are from the semitic family, show low recall when the automatic generated summaries are compared against human abstracts. This is probably due to the assumption that summaries from these languages are made by abstraction rather than extracting some parts of the original documents. In the contrary, languages such as English and French tend to have better recall. So,

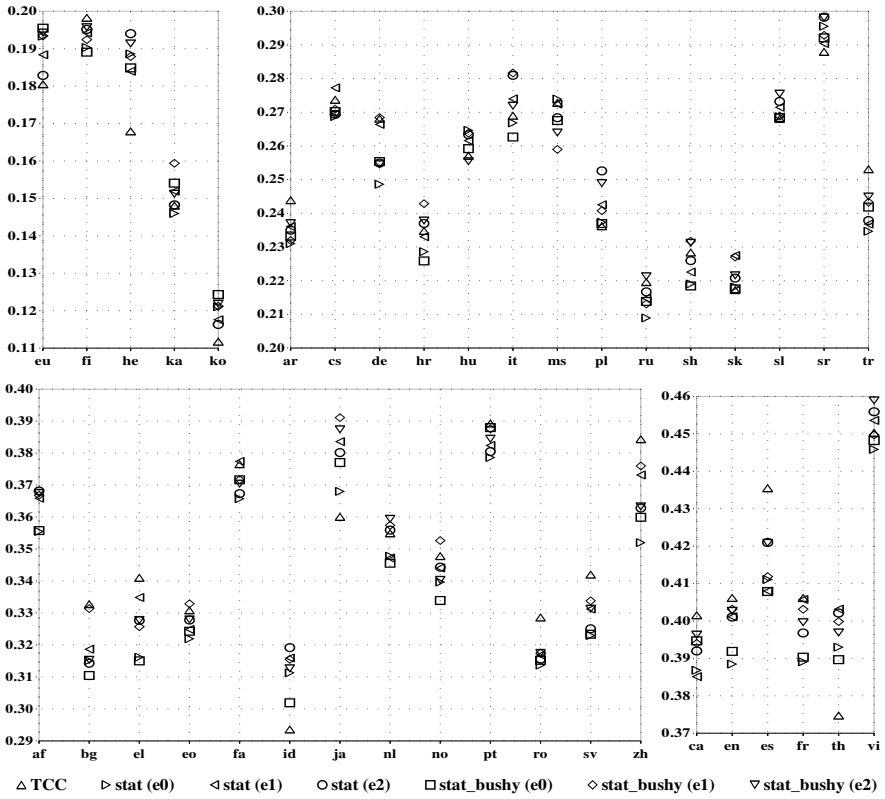

Fig. 1. ROUGE-1 Recall for 38 languages: comparison between AllSummarizer_TCC and different parameters of our method.

professional summarizers of these languages use parts of the original document to construct final summaries. In fact, it is shown in [11] that 78% of summaries sentences come from the original document while half of them have been compressed.

The average ROUGE-1 recall scores of every method over all 38 languages are represented in Fig. 2. It is clear that using statistical features with bushy paths property and extraction method $e1$ gives better average recall. Overall, bushy paths improves the recall score of summaries.

In English language, our method did not perform as well as the baseline system in term of ROUGE-1 recall score. In the contrary, looking to ROUGE-2 recall scores (see Table 1), it did a good job. Also, in term of precision, our method performs better which means it affords less redundant summaries.

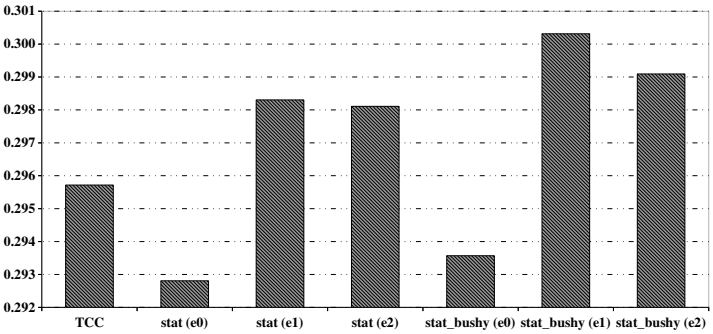

Fig. 2. ROUGE-1 Recall scores average for 38 languages.

Peer	ROUGE-1			ROUGE-2		
	R	P	F1	R	P	F1
TCC (allsummarizer)	**0.40619**	0.40866	0.40709	0.10053	0.10115	0.10077
score_stat (e0)	0.38848	0.40101	0.39431	0.10141	0.10478	0.10298
score_stat (e1)	0.40113	0.40504	0.40284	0.10447	0.10539	0.10487
score_stat (e2)	0.40104	0.41323	0.40658	**0.11030**	**0.11351**	**0.11176**
score_bushy (e0)	0.39187	0.40658	0.39874	0.10514	0.10881	0.10686
score_bushy (e1)	0.40314	0.41215	0.40703	0.10974	0.11202	0.11072
score_bushy (e2)	0.40267	**0.41380**	**0.40788**	0.10818	0.11097	0.10948

Table 1. ROUGE-1 and ROUGE-2 of AllSummarizer_TCC and our method for English testing set.

5 Conclusion

We tried to improve statistical scores by exploiting the bushy paths property of graphs. A graph of similarities between sentences is constructed. First, the graph is minimized by deleting unimportant nodes and weak arcs. This can help decreasing the number of sentences to be considered for scoring, especially when there is a big document to be processed. To score each candidate sentence, statistical are used to calculate some sort of probabilities of a sentence being relevant to the main topic of the input document. Then, using the minimized graph, each sentence is re-scored based on its initial statistical score and the number of its neighbors. We want to increase the amount of information a summary can express compared to an abstract made by humans, which is known as summary's informativeness. But, it is not the only issue of ATS; There are also redundancy and readability. These three properties are often inconsistent; i.e. we can not enhance one without affect the quality of another. So, we tried to use an extraction method which prefers non redundant sentences from the high

scored ones. Also, we tried to propose a testing version of an extraction method which aims to exploit graph's links when generating sentences in order to have some coherence.

To test our method, we used the testing corpus from MultiLing'15 workshop and one of its participant systems as a baseline. Our method shows improvements in recall score with many languages. In term of precision, using graph's bushy paths improves the original scoring method. Also, trying to extract coherent sentences can cause a little drop in summaries recall score. This is why we have to reformulate this method in order to find a trade-off between informativeness, non redundancy and coherence. We should, also, point out that we fixed our similarity threshold to be the mean of sentences similarities. The value of this threshold can affect our method's performance greatly, which means it has to be investigated in future works. We may, as well, consider iterative graphs with some conditions to prevent non convergence.

References

1. Allan, J., Wade, C., Bolivar, A.: Retrieval and novelty detection at the sentence level. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval. pp. 314–321. SIGIR '03, ACM, New York, NY, USA (2003), http://doi.acm.org/10.1145/860435.860493
2. Aries, A., Oufaida, H., Nouali, O.: Using clustering and a modified classification algorithm for automatic text summarization. Proc. SPIE, vol. 8658, pp. 865811–865811–9 (2013), http://dx.doi.org/10.1117/12.2004001
3. Aries, A., Zegour, E.D., Hidouci, W.K.: Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue, chap. AllSummarizer system at MultiLing 2015: Multilingual single and multi-document summarization, pp. 237–244. Association for Computational Linguistics (2015), http://aclweb.org/anthology/W15-4634
4. Baxendale, P.B.: Machine-made index for technical literature: an experiment. IBM J. Res. Dev. 2(4), 354–361 (Oct 1958), http://dx.doi.org/10.1147/rd.24.0354
5. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. In: Seventh International World-Wide Web Conference (WWW 1998) (1998), http://ilpubs.stanford.edu:8090/361/
6. Edmundson, H.P.: New methods in automatic extracting. J. ACM 16(2), 264–285 (Apr 1969), http://doi.acm.org/10.1145/321510.321519
7. Erkan, G., Radev, D.R.: Lexrank: Graph-based lexical centrality as salience in text summarization. J. Artif. Int. Res. 22(1), 457–479 (Dec 2004), http://dl.acm.org/citation.cfm?id=1622487.1622501
8. Fattah, M.A., Ren, F.: Ga, mr, ffnn, pnn and gmm based models for automatic text summarization. Comput. Speech Lang. 23(1), 126–144 (Jan 2009), http://dx.doi.org/10.1016/j.csl.2008.04.002
9. Giannakopoulos, G., El-Haj, M., Favre, B., Litvak, M., Steinberger, J., Varma, V.: Tac 2011 multiling pilot overview. In: Proceedings of the Fourth Text Analysis Conference (TAC 2011)–MultiLing Pilot Track. Gaithersburg, Maryland, USA (2011)

10. Giannakopoulos, G.: Multi-document multilingual summarization and evaluation tracks in acl 2013 multiling workshop. In: Proceedings of the MultiLing 2013 Workshop on Multilingual Multi-document Summarization. pp. 20–28. Association for Computational Linguistics, Sofia, Bulgaria (August 2013), http://www.aclweb.org/anthology/W13-3103
11. Jing, H., McKeown, K.R.: The decomposition of human-written summary sentences. In: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. pp. 129–136. SIGIR '99, ACM, New York, NY, USA (1999), http://doi.acm.org/10.1145/312624.312666
12. Kubina, J., Conroy, J., Schlesinger, J.: Acl 2013 multiling pilot overview. In: Proceedings of the MultiLing 2013 Workshop on Multilingual Multi-document Summarization. pp. 29–38. Association for Computational Linguistics, Sofia, Bulgaria (August 2013), http://www.aclweb.org/anthology/W13-3104
13. Kupiec, J., Pedersen, J., Chen, F.: A trainable document summarizer. In: Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 68–73. SIGIR '95, ACM, New York, NY, USA (1995), http://doi.acm.org/10.1145/215206.215333
14. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Proc. ACL workshop on Text Summarization Branches Out. p. 10 (2004), http://research.microsoft.com/ cyl/download/papers/WAS2004.pdf
15. Luhn, H.P.: The automatic creation of literature abstracts. IBM J. Res. Dev. 2(2), 159–165 (Apr 1958), http://dx.doi.org/10.1147/rd.22.0159
16. Mihalcea, R., Tarau, P.: Textrank: Bringing order into texts. In: Lin, D., Wu, D. (eds.) Proceedings of EMNLP 2004. pp. 404–411. Association for Computational Linguistics, Barcelona, Spain (July 2004), http://www.aclweb.org/anthology/W04-3252
17. Nobata, C., Sekine, S.: Crl/nyu summarization system at duc-2004. In: DUC (2004)
18. Quillian, M.R.: Semantic Memory, pp. 227–270. The MIT Press (1968)
19. Salton, G., Yang, C.S.: On the specification of term values in automatic indexing. Journal of Documentation. 29(4), 351–372 (1973)
20. Salton, G., Singhal, A., Mitra, M., Buckley, C.: Automatic text structuring and summarization. Inf. Process. Manage. 33(2), 193–207 (Mar 1997), http://dx.doi.org/10.1016/S0306-4573(96)00062-3
21. Wan, X.: Timedtextrank: Adding the temporal dimension to multi-document summarization. In: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 867–868. SIGIR '07, ACM, New York, NY, USA (2007), http://doi.acm.org/10.1145/1277741.1277949
22. Wan, X.: An exploration of document impact on graph-based multi-document summarization. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing. pp. 755–762. EMNLP '08, Association for Computational Linguistics, Stroudsburg, PA, USA (2008), http://dl.acm.org/citation.cfm?id=1613715.1613811
23. Yeh, J.Y., Ke, H.R., Yang, W.P.: ispreadrank: Ranking sentences for extraction-based summarization using feature weight propagation in the sentence similarity network. Expert Systems with Applications 35(3), 1451 – 1462 (2008), http://www.sciencedirect.com/science/article/pii/S0957417407003612