Multi-Document Extractive Summarization as a Non-linear Combinatorial Optimization Problem

Meghana N. Satpute, Luobing Dong, Weili Wu, and Ding-Zhu Du

Abstract Multi-document summarization deals with finding the core theme presented in multiple documents. This can be done by selecting the important information from the text in the multiple documents. Extractive summarization selects and extracts such sentences which represent the gist of the documents. In this paper, we have surveyed how research in multi-document summarization has evolved from simple sentence-based techniques like sentence position to complex neural network based supervised learning techniques. In recent years, more and more supervised learning methods are proposed to tackle this problem along with some unsupervised approaches described in LSA (Deerwester et al. J Am Soc Inf Sci 41(6): 391–407, 1990) and TextRank (Mihalcea et al. Textrank: Bringing order into text. In: Proceedings of the 2004 conference on empirical methods in natural language processing, 2004). In this chapter, we have proposed an alternative unsupervised method where the problem of multi-document summarization can be viewed as a non-linear combinatorial optimization problem. We have formulated the problem and discussed possible solution to this problem.

1 Introduction

Automatic multi-document summarization is a process of creating shorter version of given text from different but related documents in such a way that it retains the important information the documents are meant to convey. With the advent of Internet, vast amount of data became accessible to everyone. People are better equipped to gain knowledge and make decisions. From online shopping to reading

M. N. Satpute (✉) · W. Wu · D.-Z. Du
Department of Computer Science, The University of Texas at Dallas, Richardson, TX, USA
e-mail: mns086000@utdallas.edu; weiliwu@utdallas.edu; dzdu@utdallas.edu

L. Dong
School of Telecommunications Engineering, Xidian University, Xian, China
e-mail: lbdong@xidian.edu.cn

D.-Z. Du et al. (eds.), *Nonlinear Combinatorial Optimization*,
Springer Optimization and Its Applications 147,
https://doi.org/10.1007/978-3-030-16194-1_15

books, people can read reviews and summaries. But due to time constraint, it is not possible to read every webpage or document available. Thus, people are more inclined to read summaries, i.e., summary of book, summary of news articles, etc. Hence, multi-document summarization research is gaining momentum due to its practical usefulness in day to day life.

There are two main approaches of automatic summarization techniques: extractive and abstractive. Extraction-based summarizers extract individual sentences from the given text (multiple documents). Depending on some criteria, these sentences are deemed important by summarizer. The final summary is composed by using these extracted sentences. Thus, the sentences in the summary come directly from the given text. Abstraction-based summarizers select important sentences or paragraphs from the given text but the final summary is composed by generating new sentences using the selected sentences. Thus, the sentences in summary are often different from sentences in given text.

Initial research in automatic summarization has emphasized on extracting summary from single document. Later on, as more and more text becomes available online, reader wanted to gather information from different but related documents. Hence, the research has diverted more towards multi-document summarization.

Multi-document summarization is a complex problem. A good summarizer is expected to cover all important information from the text from different documents, avoid redundancy, and produce coherent sentences as summary. To train the summarizer, large annotated data is needed but often not available. Even the available data does not cover different document styles, i.e., email data and social network data. Furthermore, there is no consensus among researchers about which sentences are needed to be in summary and which are not, making it harder problem to solve.

In this paper, we study how summarization techniques have evolved from simple heuristic-based techniques to applying complex neural network based learning mechanisms. Lin and Blimes [16] first noticed submodularity of natural language processing (NLP) problems and proposed that these problems can be solved as optimization problems. Extractive multi-document summarization is essentially selecting subset of sentences from a set of related documents based on some constraints. We discuss this problem as a combinatorial optimization problem and formulate the problem.

Section 2 describes background and approaches of summarization problem in early days. Section 3 describes the survey of how summarization techniques have been evolved for summarizing texts. Section 4 proposes a new way to look at extracting summary problem as non-linear combinatorial optimization problem and depicts possible formulation of this problem as an optimization problem. Section 5 concludes the contributions of this work.

2 Background

Automatic summarization efforts were started by researchers in late 1950s when they wanted to have condensed version of scientific and research papers which covers the important content.

2.1 Heuristic-Based Methods

Even though initial summarization methods were not as complex as today's summarization techniques, they were efficient and good enough for summarization needs for that time. These methods were mainly based on some rules or heuristics about how to decide which sentences are important. Once this decision is made, the important sentences are extracted as summary. In the method proposed by Luhn et al. [18], first the stop-words are removed from the sentences since even if they occur frequently, they do not add much meaning to summary. For all remaining words, their frequency is calculated and frequent words are deemed important. Sentences having many frequent words are considered summary-worthy and included in summary. In the work by Baxendale et al. [2], sentence position is given more importance. First and last sentences in a paragraph are extracted as significant sentences and included in summary. It is also observed that this assumption is true in the data set of scientific papers for which the summary is desired. Another such observation was that, in case of news articles, first two sentences of a paragraph are more significant than remaining sentences [26]. Edmundson et al. [7] proposed four components to weigh sentences, instead of just word frequency like previous research. He experimented with different weights for the presence of high-frequency keywords, pragmatic words, title and heading words, and sentence location. This research indicated that considering several linguistic features while deciding extract-worthy sentences offers better results.

3 Multi-Document Summarization Approaches

In multi-document summarization problem, the information comes from multiple documents which are related and often complement each other. While deciding the sentences to be selected, we need to make sure that they are coherent, not redundant and cover all important content. Various approaches are used in the research of multi-document summarization. Some approaches are extension of the work done for single document summarization and some are newly evolved approaches.

3.1 Statistical Approaches

The paper by Gambir and Gupta [12] has described the process of automatic extractive summarization using the following block diagram in Figure 1. Automatic summarization process begins with collecting the different but related documents from different sources. These documents are then pre-processed, i.e., removal of stop-words, stemming, etc. Then linguistic and statistical features are extracted from the documents. Based on the occurrence of features in a sentence, each sentence is scored using score function. Sentences with high score are extracted as a summary.

In [10] by Ferreira and others, an unsupervised system is built based on statistical and linguistic features in the text. They proposed a clustering algorithm to ensure coherence and reduce redundancy when multiple statements of the same meaning are present in the documents. In [5] Latent Semantic Analysis (LSA) is used to index and find topics by creating vectors of a documents based on the semantics in the text. Considering features among sentences such as statistical similarity, semantic similarity, coreference, and discourse relations, text is converted into a graph model. Main sentences are identified by using TextRank algorithm [21]. Based on similarity among the sentences, clusters of sentences are formed. Finally, main sentences from the clusters are selected to form summary.

In research by Ko et al. [15], a hybrid method is proposed which makes use of contextual and statistical information in the given text. In this method, two consecutive sentences are merged and bigram pseudo sentences are formed. Several statistical features are combined to score the pseudo sentence, such as how far the sentence is from the title, the location of sentence, score of a sentence based on aggregation similarity (which is the sum of similarities with all other sentences), term frequency of terms in the sentence, and term frequency based query (where high-frequency terms are used to query the document to find important sentence). After the extraction of high score bigram pseudo sentences, the sentences are fragmented to original sentences and summary is generated. They achieved performance gain due to combination of several important features for deciding which sentences need to be extracted.

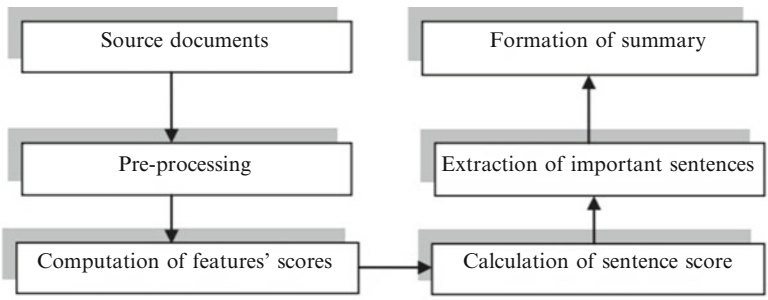

Fig. 1 Extractive summarization using statistical approach [12]

Yeh et al. [31] used different kinds of statistical and contextual features, including sentence position in the paragraph (first sentence in paragraph introduces paragraph and last sentence in paragraph summarizes paragraph), positive or negative keywords (positive keywords are most likely included in summary, while negative keywords are omitted), and centrality of sentence (similarity of the sentence with the other sentences in text. If sentence is too similar with other sentences, then it means it reflects central theme, resemblance with the title of the document. The summary is generated by using linear weighted combination of these features to obtain score function. Genetic algorithm is implemented to find optimal weights of the features while extracting the summary.

3.2 Topic Based Approaches

Topic signatures extract topic-related sentences as summary through two steps, i.e., topic recognition and interpretation. These two steps are considered as two basic steps in a typical automated text summary system. Lin and Hovy [17] proposed the idea of topic signatures. Each topic signature is represented as the terms related to the topic and weight of the term to that topic. One example mentioned in their paper is of topic *restaurant visit* which can be inferred by terms such as menu, waiter, order, etc. It is observed that often the topic words co-occur and hence their co-occurrence suggests that they belong to same topic. The sentences are scored based on their relevance to the topic signatures and high scored sentences are included in summary.

Harabagiu and Lacatusu [13] used two novel topic representations based on topic themes. Then based on their topics, the documents are classified as relevant or non-relevant to the pertaining topic. Sentences are ranked based on their score. In this paper, they considered relation between sentences and within the sentences. They used shallow semantic information from the text on top of lexical information. Extraction of summary sentences is done in different ways based on topic signature, sentence score, weights on topic relevant terms, etc.

3.3 Graph-Based Approaches

These methods converted text into graph by using vertices to represent sentences or concepts and edges to represent the semantic relatedness between two sentences or concepts. Graph-based summarization became effective summarization technology due to capturing contextual information among concepts.

In [20], Mani and Bleodorn described a graphical model which captures concepts shown by words, proper nouns, and phrases and then designate those as vertices. Edges represent the semantic relations between vertices. Figure 2 from [20] depicts three possible relations between concepts. Adj links are shown between adjacent

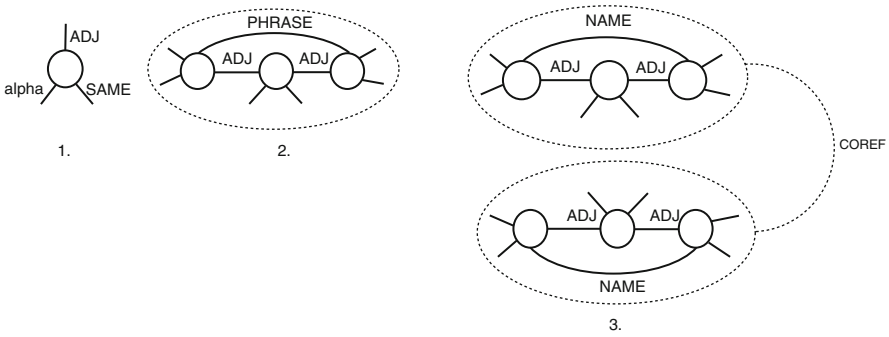

Fig. 2 Possible semantic relations between concepts [20]

concepts in the text. Name links can show person or entities. Phrase links can tie concepts together in a phrase. Coref links tie a concept to another whenever there is coreference. Alpha links are used when two concepts point to the same meaning, i.e., "President" in one sentence can be of same meaning as "Mr. Donald Trump" in another. Sentence selection for summary is based on the coverage of same vertices in the common lists and different lists. Sentences are selected greedily based on the average activated weight of the covered words.

In LexPageRank system by Erkan and Radev [8], sentences are depicted as vertices and link between vertices exists if the cosine similarity between two sentences exceeds predefined threshold. Sentence clusters are formed on the basis of sentence similarity. They hypothesize that the sentence which is more similar to other sentences contain main theme and hence central. The degree of each vertex is calculated. Each link or edge between vertices represents a vote. They used PageRank [24] algorithm to calculate vote of each link. Most voted sentences are included in summary. Their architecture also took care of sentence subsumption. When one sentence subsumes information from another sentence and possesses some additional information, then it is included in summary and another sentence is omitted from summary.

3.4 Machine Learning Based Approaches

As the research progressed, machine learning based methods caught the attention of scientific research. Machine learning methods enable computer to summarize documents by learning from the original documents and "understanding" the potential semantics. For example, methods using classifiers such as Naive Bayesian, support vector machine (SVM) [9], recurrent neural network (RNN) [23], neural convolution network (NCN) attention [30], and recursive neural network [4] have shown significant performance gains.

Machine learning based methods fall into one of the following categories: supervised, semi-supervised, or unsupervised. Supervised learning relies on large data sets to learn features and then using those features, it can classify the test data. Availability of big data set which covers all possible variations is a bottleneck in research of NLP tasks, as it is very time consuming to annotate the data. Unsupervised learning methods learn from the available target data for which the NLP task is to be performed. Classification task is done by learning features from that data itself. Semi-supervised approach relies on some seed examples provided by the user. From these examples patterns are learned and classification task is done.

Fattah et al. [9] experimented summarization problem using many machine learning classifiers such as Naive Bayesian, maximum entropy, support vector machine, decision trees, neural networks, mathematical regression, etc. The method considers summary generation task as a classification problem and each sentence is either included in summary or excluded. Fattah et al. [9] employed a hybrid model for summary generation task which constitutes the following three classifiers: maximum entropy, Naive Bayes, and support vector machine. Several features are taken into account to train the classifiers. For example, words similarity between sentences and between paragraphs, score using term frequency of document, key phrases, position of sentence, occurrence of not-needed information, text format, etc. These features are provided to the three classifiers in training phase. In testing phase, features are extracted and sentences are ranked by feature weights learned in the training phase. Using hybrid model of three classifiers, final summary is created.

Cao et al. [4] presented recursive neural networks (R2N2) to score sentences for extractive multi-document summarization. Each sentence is first converted into a parse tree. Information from different parts of sentence is gathered and fed to R2N2. Some features used in this process are term frequency, inverse document frequency, sentence length, named entity, position of sentence, etc. Sentence relevance is evaluated and sentence rank is given by a hierarchical regression process. On the basis of information from word level to sentence level, features are learned by recursive neural networks apart from the given features. Important sentences for summary are selected based on their ranking score. They employed greedy algorithm and integer linear programming (ILP) for selecting the sentences to be part of summary.

Nallapati et al. [23] presented SummaRuNNer, a model based on recurrent neural network (RNN) for extractive multi-document summarization. SummaRuNNer is trained using reference summaries. Summarization is considered as sequential binary classification problem where each sentence is classified as summary sentence or non-summary sentence. They used two-layer bidirectional RNN where one layer operates at word level, while another layer operates at sentence level. Using words and word embedding, hidden states are generated. They use greedy approximation to create labels from given summaries. The entire document is modeled as follows:

$$d = tanh\left(W_d * (1/N_d) \sum_{j=1}^{N_d} [h_j^f, h_j^b] + b\right), \tag{1}$$

where states starting with h are the hidden states corresponding to the sentence of the forward and backward sentence-level RNNs, respectively, and N_d is the number of sentences in the document. Features such as novelty of a sentence, position, and content are considered in their study. These features are learned rather than hand crafting and providing it to system.

3.5 Optimization Based Approaches

Lin and Blimes [17] were first to notice the submodularity in summarization. They modeled the problem as a knapsack constraint of selecting subset of sentences S from the sentences of whole document set V under the constraint of length of summary. They devised it as a maximization function of quality of summary shown in the following formula, where c_i is cost for adding sentence s_i in summary and b is budget constraint

$$\begin{aligned} S' &\in \arg\max_{S \subseteq V} F(S) \\ s.t. &\sum_{i \subseteq S} c_i \geq b. \end{aligned} \tag{2}$$

Although it is NP-hard problem, it can be solved using greedy algorithm. It becomes too computationally expensive for real-world applications [17].

Shigematsu and Kobayashi [29] used differential evolution approach to overcome the problem of computational complexity of optimization function for summarization. First they used LDA [3] to detect topics in the text. Sentences are ranked based on the topical information each sentence possesses. Resulting number of summary sentences will depend upon the length constraint over summary sentences. This method has reduced the calculation time to generate summary, to great extent but precision is worse than the method with an explicit solution technique using greedy algorithm [29].

Galanis and others used combination of support vector regression (SVR) along with integer linear programming (ILP) [11]. They used features such as sentence position, named entities, Levenshtein distance, word overlap, and content word frequency. These features are given to SVR to score each sentence from the document set. Instead of using the sentence scores directly to formulate summary, they first normalize the sentence scores. These scores are multiplied by the length of the sentence to take care of problem of the method picking short sentences. Importance of summary is calculated by adding the normalized scores of sentences. The importance of summary is maximized. While forming final summary, the number of distinct bigrams it can cover is also maximized. The underlying assumption was that the more the number of bigrams, the summary covers the less redundant it is. Like previous method by Shigematsu [29], length of summary sentences is considered the constraint over which ILP is done.

4 Our Approach to Multi-Document Summarization as a Non-linear Combinatorial Optimization Problem

In summarization, a group of sentences is selected from a bigger group of sentences. Whenever a subset of elements is to be selected from a set of elements based on some constraint, then we can formulate that problem as submodular or supermodular optimization problem.

The maximization or minimization of a set function can be formulated as combinatorial optimization problem. They are widely used in many areas of computer science and applied mathematics [6]. Minimization or maximization problems are defined on the set of subsets of a given base set S. In combinatorial optimization, submodular/supermodular functions have a role somewhat similar to that played by convex/concave functions in continuous optimization [1]. Researchers also proved that some existing extractive summarization methods can be viewed as a problem of submodular function maximization [16], such as maximum marginal relevance (**MMR**).

Given a finite set S, we use 2^S to denote the power set of S. A set function $f : 2^S \rightarrow \mathbb{R}$ is submodular if it satisfies one of the following equivalent conditions:

- For every $A, B \subseteq S$ with $A \subseteq B$ and every $a \in S - B$ we have that $f(A \cup \{a\}) - f(\{a\}) \geq f(B \cup \{a\}) - f(\{a\})$.
- For every $A, B \subseteq S$, we have that $f(A) + f(B) \geq f(A \cup B) + f(A \cap B)$.
- For every $A \subseteq S$ and $a_1, a_2 \in S - A$, we have that $f(A \cup \{a_1\}) + f(A \cup \{a_2\}) \geq f(A \cup \{a_1, a_2\}) + f(A)$.

A set function f is monotonically increasing, if for every $A \subseteq B$ we have $f(A) \leq f(B)$. A function f is supermodular if and only if $-f$ is submodular. A set function f is monotonically increasing, if for every $A \subseteq B$ we have $f(A) \leq f(B)$.

When a single element is added to an input set, as the size of the input set increases, the difference in the incremental value of a submodular function decreases. For a combinatorial optimization problem, a greedy algorithm can be designed if the objective function is submodular. Greedy algorithm can give an approximate solution in polynomial time with an approximation guaranteed to be within $\frac{e-1}{e} \approx 0.63$ of the optimal solution [22].

4.1 Diversity in Summary

Main objective of summary is to obtain maximum information from a given document set in short version in such a way that it captures the gist of the document set. In order to capture more information from document, it is necessary to cover diverse topics from the documents.

Diversity is a central theme in ecology. The diversity concept was first used by ecologists to measure the number of different species in community quantitatively.

Ecological communities with many species are more diverse than ecological communities with fewer species. Ecologists tried to sample with high diversity to increase the probability of finding small species [19]. Ecologists have proposed many methods to measure the diversity of species in these decades, such as the Shannon index or the Simpson index [28]. A diversity index called as quadratic diversity (Q) is proposed by Rao [27]. $Q = \sum_{i=1}^{S} \sum_{j=1}^{S} d_{ij}\, p_i\, p_j$ quadratic diversity incorporates both species relative abundances ($p_i\, p_j$) and a measure of the pairwise distances between species (d_{ij}).

The main theme of summarization is to get as much content from the documents as possible in a compact manner. Thus we want to ensure that the summary gathers information about all topics from the documents. If document set has sentences that convey the same meaning, then these are redundant sentences and must be omitted from summary. If there is sentence limit on summary and redundant sentences exist in the summary, then some other information, which should be part of summary, is missed from the summary. Summarization diversity controls this problem of redundancy and gives the reader insight into different distinct topics covered in the text.

Shannon entropy was originally proposed to quantify the amount of information in a signal or event. For a discrete random variable X with possible states $x_1, x_2, \cdots x_n$, its Shannon entropy is defined as Formula (3). In Formula (3), $p(x_i) = Pr(X = x_i)$ is the probability of x_i

$$H(X) = \sum_{i=1}^{n} p(x_i) \log_2 \left(\frac{1}{p(x_i)} \right). \tag{3}$$

Intuitively, when there is only one possible state of X, then $H(X)$ becomes 0. $H(x)$ increases as number of possible states of X increase denoting more diversity.

4.2 Problem Formulation

In this paper, we address the problem of how to extract a summary from a set of documents that covers as much real content of all subtopics as possible. We first identify all subtopics from the document set and then summarize the documents.

After extracting subtopics, assume that we get a subtopic set C. In this subsection, we propose a new formulation for a method to extract a small and limited set of sentences from set C which can be representative of the entire document set, which is also the goal of the summarization.

We cannot give equal weightage to all subtopics because some subtopics constitute a very few number of sentences, while other subtopics might have many sentences written about them. Thus the subtopics having small number of sentences do not contribute much to the document set and hence can be deleted. For simplicity,

we use the symbol C' to denote the subtopic set after the small sized subtopics are deleted.

In this step, for the process of calculation of semantic distance, single sentence is taken into consideration as the basic unit of calculation. C' can be represented as a subtopics covering set of sentences $C' = \{st_1, st_2, \ldots, st_q\}$, where each sentence st_i belongs to a subtopic s_k and it also belongs to a paragraph set c_k. If $st_i \in p_{ef}$ and $p_{ef} \triangleq s_k$, then $st_i \in c_k$.

We use notation $SDS(st_i, st_j)$ to represent the semantic distance between two sentences st_i and st_j. $SDSS(st_i, A)$ $(A \subseteq C')$ is used to denote the semantic distance between a sentence st_i to a sentence set A. It is defined as the following Formula (4):

$$SDSS(st_i, A) = \min_{st_j \in A} SDS(st_i, st_j). \tag{4}$$

The semantic distance between the two subsets of C' is represented as a function $SDTS : (2^{C'}, 2^{C'}) \rightarrow \mathbb{R}$. $SDTS$ can be defined as Formula (5)

$$SDTS(A, B) = \sum_{st_j \in (B-A)} SDSS(st_j, A) \qquad A \subseteq C', B \subseteq C'. \tag{5}$$

The set of summary sentences is a size limited subset I of C' which is a set of sentences from document set D. We need to find I such that the similarity between sets I and D is as high as possible, which intuitively means the semantic distance between I and C' is as small as possible. At the same time, we also want subtopic diversity, the more the subtopics that I can cover, the better. Shannon entropy can be used to measure the diversity (Formula (6))

$$HD(I) = \sum_{i=1}^{q} \frac{|I_k|}{|I|} \log_2 \left(\frac{|I|}{|I_k|} \right) \qquad I_k = \{st_i | st_i \in C', st_i \in c_k\}. \tag{6}$$

Therefore, there are two targets: minimizing the distance between I and C' (Formula (7)) and maximizing the subtopic diversity (Formula (8)). At the same time, we have a constraint that the sentence number of the final summary is less than some constant b ($|I| \leq b$). The summary subtopic diversity $HD(I)$ is known to be submodular and monotone increasing [25]. Interestingly, $SDTS(I, C')$ is monotone decreasing and supermodular.

$$\min_{I \subseteq C'} SDTS(I, C') \tag{7}$$

$$\max_{I \subseteq C'} HD(I). \tag{8}$$

We know that the minimizing value of the supermodular function $SDTS(I, C')$ maximizes value of $-SDTS(I, C')$ which is submodular and the maximization of a submodular function with cardinality constraint is NP-hard [14]. Fortunately, $HD(I) - \gamma SDTS(I, C')$ is submodular, and we can formulate our objective function as Formula (9) which can be approximately solved

$$\begin{aligned} &\arg\max_{I \subseteq C'} HD(I) - \gamma SDTS(I, C') \\ &s.t. \qquad\qquad |I| \leq b, \end{aligned} \tag{9}$$

where γ is a parameter which can be adjusted experimentally.

5 Conclusion

Through this work, we have surveyed different approaches in multi-document summarization and proposed it as a combinatorial optimization problem. The proposed formulation can extract meaningful summary from multiple documents. Using the sentence distances and subtopics as backbones we have formulated the problem as submodular combinatorial optimization problem of minimizing distance between summary and document set and maximizing subtopic diversity in the summary.

References

1. Bach, F., et al.: Learning with submodular functions: a convex optimization perspective. Found. Trends® Mach. Learn. **6**(2-3), 145–373 (2013)
2. Baxendale, P.B.: Machine-made index for technical literature: an experiment. IBM J. Res. Dev. **2**(4), 354–361 (October 1958)
3. Blei, D.M., Ng, A.Y., Jordan, M.I., Lafferty, J.: Latent Dirichlet allocation. J. Mach. Learn. Res. **3**, 2003 (2003)
4. Cao, Z., Wei, F., Dong, L., Li, S., Zhou, M.: Ranking with recursive neural networks and its application to multi-document summarization. In: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, AAAI'15, pp. 2153–2159. AAAI, Palo Alto (2015)
5. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. J. Am. Soc. Inf. Sci. **41**(6), 391–407 (1990)
6. Dong, L., Guo, Q., Wu, W.: Speech corpora subset selection based on time-continuous utterances features. J. Comb. Optim. 1–12 (2018). https://doi.org/10.1007/s10878-018-0350-2
7. Edmundson, H.P.: New methods in automatic extracting. J. ACM **16**(2), 264–285 (1969)
8. Erkan, G., Radev, D.R.: LexPageRank: prestige in multi-document text summarization. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (2004)
9. Fattah, M.A.: A hybrid machine learning model for multi-document summarization. Appl. Intell. **40**(4), 592–600 (2014)

10. Ferreira, R., Cabral, L.D.S., çalves de Freitas, F.L.G., Lins, R.D., de França Pereira e Silva, G., Simske, S.J., Favaro, L.: A multi-document summarization system based on statistics and linguistic treatment. Expert Syst. Appl. **41**(13), 5780–5787 (2014)
11. Galanis, D., Lampouras, G., Androutsopoulos, I.: Extractive multi-document summarization with integer linear programming and support vector regression. In Proceedings of COLING, pp. 911–926. IIT, Bombay (2012)
12. Gambhir, M., Gupta, V.: Recent automatic text summarization techniques: a survey. Artif. Intell. Rev. **47**, 1–66 (2016)
13. Harabagiu, S.M., Lacatusu, V.F.: Using topic themes for multi-document summarization. ACM Trans. Inf. Syst. **28**(3), 13:1–13:47 (2010)
14. Iyer, R.K., Bilmes, J.A.: Submodular optimization with submodular cover and submodular knapsack constraints. In: Advances in Neural Information Processing Systems, pp. 2436–2444 (2013)
15. Ko, Y., Seo, J.: An effective sentence-extraction technique using contextual information and statistical approaches for text summarization. Pattern Recogn. Lett. **29**(9), 1366–1371 (2008)
16. Lin, H., Bilmes, J.: A class of submodular functions for document summarization. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 510–520. Association for Computational Linguistics, Stroudsburg (2011)
17. Lin, C.-Y., Hovy, E.: The automated acquisition of topic signatures for text summarization. In: Proceedings of the 18th Conference on Computational Linguistics - Volume 1, COLING '00, pp. 495–501. Association for Computational Linguistics, Stroudsburg (2000)
18. Luhn, H.P.: The automatic creation of literature abstracts. IBM J. Res. Dev. **2**(2), 159–165 (1958)
19. Magurran, A.E.: Why diversity? In: Ecological diversity and its measurement, pp. 1–5. Springer, Dordrecht (1988)
20. Mani, I., Bloedorn, E.: Multi-document summarization by graph search and matching. In: Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Conference on Innovative Applications of Artificial Intelligence, AAAI'97/IAAI'97, pp. 622–628. AAAI, Cambridge (1997)
21. Mihalcea, R., Tarau, P.: Textrank: bringing order into text. In: Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing (2004)
22. Minoux, M.: Accelerated greedy algorithms for maximizing submodular set functions. In: Optimization Techniques, pp. 234–243. Springer, Berlin (1978)
23. Nallapati, R., Zhai, F., Zhou, B.: Summarunner: a recurrent neural network based sequence model for extractive summarization of documents. In: AAAI, pp. 3075–3081. AAAI, Cambridge (2017)
24. Page, L., Brin, S., Motwani, R., Winograd, T.: The PageRank citation ranking: bringing order to the web. Technical Report 1999-66, Stanford InfoLab, November 1999. Previous number = SIDL-WP-1999-0120
25. Polyanskiy, Y.: Lecture notes, chapter 1: Information measures: entropy and divergence (January 2016)
26. Radev, D.: [Artificial Intelligence - All in one]. (2016, April 5). Summarization Techniques (NLP) University of Michigan [Video file]. Retrieved from https://www.youtube.com/watch?v=N5N-HCUE3G4
27. Rao, C.R.: Diversity and dissimilarity coefficients: a unified approach. Theor. Popul. Biol. **21**(1), 24–43 (1982)
28. Ricotta, C., Szeidl, L.: Towards a unifying approach to diversity measures: bridging the gap between the Shannon entropy and Rao's quadratic index. Theor. Popul. Biol. **70**(3), 237–243 (2006)
29. Shigematsu, H., Kobayashi, I.: Topic-based multi-document summarization using differential evolution for combinatorial optimization of sentences. In: Proceedings of the 28th Pacific Asia Conference on Language, Information and Computing (2014)

30. Yasunaga, M., Zhang, R., Meelu, K., Pareek, A., Srinivasan, K., Radev, D.: Graph-based neural multi-document summarization. In: Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), pp. 452–462. Association for Computational Linguistics, Vancouver (2017)
31. Yeh, J.-Y., Ke, H.-R., Yang, W.-P., Meng, I.-H.: Text summarization using a trainable summarizer and latent semantic analysis. Inf. Process. Manag. **41**(1), 75–95 (2005). An Asian Digital Libraries Perspective