Expert Systems With Applications 133 (2019) 173–181

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Extractive summarization using supervised and unsupervised learning

Xiangke Mao[a,*], Hui Yang[b], Shaobin Huang[a], Ye Liu[a], Rongsheng Li[a]

[a] *College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China*
[b] *CETC Big Data Research Institute Co., Ltd., Guiyang 550022, China*

ARTICLE INFO

Article history:
Received 18 July 2018
Revised 9 May 2019
Accepted 9 May 2019
Available online 16 May 2019

Keywords:
Supervised learning
Unsupervised learning
Combination
Summarization
Biased-LexRank

ABSTRACT

In this paper, three methods of extracting single document summary by combining supervised learning with unsupervised learning are proposed. The purpose of these three methods is to measure the importance of sentences by combining the statistical features of sentences and the relationship between sentences at the same time. The first method uses supervised model and graph model to score sentences separately, and then linear combination of scores is used as the final score of sentences. In the second method, the graph model is used as an independent feature of the supervised model to evaluate the importance of sentences. The third method is to score the importance of sentences by supervised model, then as a priori value of nodes in the graph model, and finally use biased graph model to score sentences. On the data sets of DUC2001 and DUC2002, the ROUGE method is used as the evaluation criterion, which shows that the three methods have achieved good results, and are superior to the methods of extracting summary only using supervised learning or unsupervised learning. We also validate that priori knowledge can improve the accuracy of key sentence selection in graph model.

© 2019 Elsevier Ltd. All rights reserved.

1. Introduction

With the rapid development of Internet technology, electronic documents on the Internet have seen explosive growth. The increase in information has enriched human life, but at the same time it has added a burden for humans to obtain the information they really need quickly and accurately. The automatic text summarization technology is more and more concerned by researchers in such a background.

The task of automatic document summarization aims at generating short summaries for originally long documents (Yao, Wan, & Xiao, 2017). It has received attention in more and more fields, including information retrieval (Glavaš & Šnajder, 2014), review analysis (Liu chien-Liang et al., 2012), sentiment analysis (Shah et al., 2016) and so on. Automatic document summarization can be categorized from different perspectives. According to the input number of processing documents, it can be divided into single document summarization (Erkan & Radev, 2004; Fang, Mu, & Deng, 2017; Yang, Bao, & Nenkova, 2017) and multi-document summarization (Fattah & Ren, 2009; Tohalino & Amancio, 2018). The single document summarization is to obtain the summary from only one document at a time, and the multi-document summarization is to generate the summary from at least two documents or more. According to the output of summary, summarization can be divided into extractive summarization (Erkan & Radev, 2004; Dutta, Madhurima, et al., 2019; Fang et al., 2017) and abstractive summarization (Khan, Salim, & Kumar, 2015; Li, W., 2015). Extractive summarization is a method that extract sentences containing salient information from the original document only, while abstractive summarization can use words or phrases different from the original document to compose the summary. Now the extract summarization is more widely used because of better feasibility. In this paper, we study the methods of extractive summarization for single document.

The most important step in the extractive summarization is ranking the sentences in the document, that is to score the sentences. In recent years, with the development of natural language processing techniques and machine learning methods, more and more sentence scoring techniques have emerged. Based on whether manually labelled data is needed, these scoring methods can be divided into supervised learning and unsupervised learning methods.

When using the supervised learning methods to score sentences, the extractive summarization generation problem is usually regarded as a binary classification problem. The process is: First, the sentences in the training documents are tagged manually, and the sentences selected for the summary are labelled as positive ones, while the other sentences are labelled as counterexamples. Then we select features from sentences or document, and use the

[*] Corresponding author.

E-mail addresses: maotiamo@hrbeu.edu.cn (X. Mao), yanghui@cetcbigdata.com (H. Yang), huangshaobin@hrbeu.edu.cn (S. Huang), 1017263392@hrbeu.edu.cn (Y. Liu), dasheng@hrbeu.edu.cn (R. Li).

https://doi.org/10.1016/j.eswa.2019.05.011
0957-4174/© 2019 Elsevier Ltd. All rights reserved.

classification models (Logistic, SVM, NN, etc.) on training data to train model and get the final classifier. Finally, the classifier is applied to new document to predict the probability of each sentence being selected as the final summary content. Feature selection is a key step in this process, the selected features can be roughly divided into word-based and sentence-based (Christian, Agus, & Suhartono, 2016; Wong, Wu, & Li, 2008). The word-based includes TF (Term Frequency), TF-IDF (Term Frequency-Inversed Document Frequency), named entity, numerical information, etc. The sentence-based includes sentence position and sentence length, etc. However, these selected features ignore the relationship between sentences, and the classification models regard each sentence independently in the training process.

Unsupervised learning is becoming more and more popular because it does not need manually annotated data, especially in the current rapid growth of data. In the past, most of work calculates the score of sentence is based word importance, they calculate word score first, then according the word score to calculate the sentence score. The essence of these approaches is that the importance of a sentence is determined by the importance of the words it contains. In order to model the relationship between sentences, a graph model based on PageRank (Page et al., 1999) algorithm was proposed, this is a kind of totally unsupervised algorithm to get extractive summarization, which determines the importance of sentences based on two rules. The first is that a sentence is very important if it is linked by a number of other sentences, and the second is that a sentence is linked to other important sentences, indicating that the sentence is also important.

In the summary generation process of this paper, in order to make better use of the statistic features of sentences and relationship between sentences, we proposed three methods to combine them for scoring sentences, and finally generate summary based on final scores.

(1) The sentences are scored using supervised and unsupervised learning methods respectively, then the scoring results are normalized and linearly combined to get the final score of sentence.
(2) First, the unsupervised method is used to score the sentences, then add the scores as an independent feature of supervised learning methods to train the classifier, finally, compare the effect of summarization before and after add scores from unsupervised method as feature.
(3) A two-stage model was proposed. In the first stage, we used the classifier that trained by supervised method to score the sentences in the new document as the prior probability of the sentence was selected to compose the final summary. In the second stage, the document is represented as a graph, and then the nodes are scored using a biased random walk. In the random jump, the nodes with large prior values are more likely to be selected.

This paper conducts experiments on DUC2001 and DUC2002 datasets respectively, evaluates the effectiveness of the proposed method using ROUGE methods. Under the ROUGE-1, ROUGE-2, ROUGE-SU4 evaluation indexes, compared with the summary generated by the methods of location-based(Lead-based), LexRank (Erkan & Radev, 2004), Latent semantic analysis(LSA) and supervised methods trained by Logistic Regression(LR), Support Vector Machine(SVM), etc. The three summarization methods proposed in this paper all have achieved an improvement in effectiveness, which validates the effectiveness of our proposed summary methods in generating extractive summary for single document.

The main contributions of this paper are as follows:

(1) We proposed three methods to score sentences by combining sentence relations with statistical features of sentences. The effect of combination method is better than that of single method.
(2) This is the first time that we have used the sentences score obtained from supervised methods as priori value of biased-LexRank, which greatly improves the LexRank algorithm.

In the following sections, we will introduce the related work of automatic text summarization in the second section. In the third section, we will introduce our proposed methods. In the fourth section, we will design the experiments and analyse the experimental results. In the fifth section, we conclude and look forward to the future work.

2. Related work

Since the birth of automatic document summarization in 1958 (Luhn H P., 1958), it has been a hot topic in the field of natural language processing. Over the past 60 years, two kinds of summarization methods have been developed, namely supervised learning (Li, Qian, & Liu, 2013; Yang et al., 2017) and unsupervised learning (Erkan & Radev, 2004, Fang et al., 2017, Yousefi-Azar et al. 2017).

In Kupiec, Pedersen, and Chen (1995), they first applied supervised learning to the field of text summary generation. They selected five features: word frequency, uppercase words, length of sentence, position in paragraph, and structure of phrase, and considered that these features were independent of each other. A naive Bayesian classification model is proposed to determine whether a sentence in a document should be selected as a summary. Since the appearance of this method, researchers have done a lot of work on feature selection and model selection. In feature selection, more and more features are used to measure the importance of a sentence. Such as sentence position (García-Hernández & Ledeneva, 2013; Ouyang, You, et al., 2010), Proper Noun (Khan et al., 2015; Meena & Gopalani, 2015), Pronouns (Saziyabegum & Sajja, 2016), term frequency (Fattah, 2014). Some work has made a comprehensive study of sentence scoring techniques. Ferreira, de Souza Cabral, and Lins (2014) conducted research and evaluation on some commonly used 17 sentence scoring techniques such as sentence length, and TF-IDF, and experimentally verified the effectiveness of various scoring techniques. Oliveira, Ferreira, and Lima (2016) studied 18 scoring techniques and their combination strategies, and experimentally verified ensemble strategies lead to improvements over the individual scoring techniques. In terms of classifier selection, some popular classification methods are used, such as Hidden Markov Models (HMM) (Conroy & O'leary, 2001), Decision Trees (Lin, 1999), Support Vector Machine (SVM) (Ouyang, You, et al., 2011) and Neural Networks (Fattah & Ren, 2009).

In order to overcome the dependence on training corpus, unsupervised learning methods have attracted more and more attention in recent years. Unsupervised learning mainly includes language model (Gupta, Pendluri, & Vats, 2011), biased centrality modelling (Ribeiro & De Matos et al., 2011; Ribeiro et al., 2013), clustering (Alguliyev, Aliguliyev, Isazade, Abdi, & Idris, 2019; Shetty & Kallimani, 2017), graph-based methods (Erkan & Radev, 2004; Dutta, Madhurima, et al., 2019; Mallick, Chirantana, et al., 2019) and complex network based methods (Amancio, Nunes, Oliveira, & Costa, 2012; Tohalino & Amancio, 2017).

In this paper, the unsupervised methods used are based on graph model. The use of graph models for sentence scoring mainly involves the selection of nodes and the calculation of relationships between nodes. In the TextRank method proposed by Mihalcea and Tarau (2004), the sentences in the document are used as nodes in the graph, and Jaccard similarity is used to calculate the relationship between nodes. In Erkan and Radev (2004), the similarity between sentences is calculated based on the cosine similarity of TF-IDF value (Term Frequency-Inverse Document Frequency). Finally, PageRank algorithm is used to calculate the weight of sentences and select summary sentences from high to low. In AL-Khassawneh, Salim, and Jarrah, (2017), they proposed a single document summarization technology based on hybrid graph, four different similarity measures, including cosine similarity, Jaccard similarity, word alignment similarity and window-based similarity measure, were combined to create a hybrid similarity function to calculate the weight of the graph. In Erkan (2006), for the focused summarization, the similarity between sentences and topic descriptions is used as a priori value of nodes in the graph, and generative probabilities as link weights between sentences. In Wan and Yang (2008), they use documents and sentences as nodes in the graph, make use of the relationship between documents and documents, as well as the relationship between sentences in the document to form the edge of the graph, and use the iterative ranking algorithm to select sentences with rich information as the final summary content. Fang et al. (2017) use the relationship between words and sentences to rank sentences. They represent the document as two matrices, one is the relationship between sentences, the other is the relationship between words and sentences, finally, the sentences are scored by combining the two relations.

In recent years, deep learning has been heavily used in summarization, such as Yousefi-Azar et al. (2017); Nallapati, Zhai, and Zhou (2017); Yao et al. (2018). But it is not covered in the methods we used in this article. The three methods we proposed are actually a combination of supervised and unsupervised learning in the traditional summarization method. The advantage of doing this is that the model can use both the statistical characteristics of the sentence and the relationship between the sentences to evaluate the importance of the sentence.

3. Proposed method

In this section, we mainly introduce our approaches proposed in this paper. For the documents we used, We process the document according to the flow shown in Fig. 1. In pre-processing step, the work we do includes dividing document into sentences, cut sentence into words, removing stop words, stemming, tagging part of speech of words, identifying named entity and so on. In Feature Selection step, we get the features from sentences. In Graph Construction and LexRank two steps, we use graph to represent the document and get the score of each sentence. Following, we will introduce the two basic sentence scoring methods.

3.1. Two basic sentence scoring methods

We first introduce two basic scoring methods, one is supervised, the other is unsupervised. The supervised method mainly introduces the features we used in the model training, the unsupervised method mainly introduces how to build the graph model and the principle of sentence ranking.

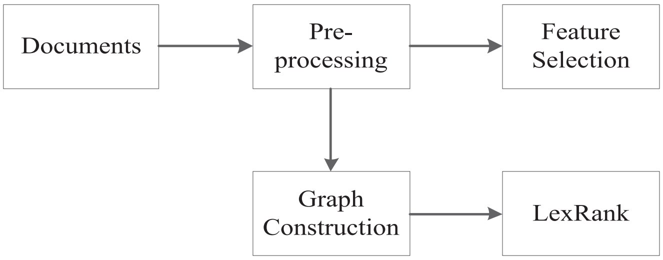


Fig. 1. Flow of process the document.

3.1.1. Supervised method

Among the sentence scoring techniques based on supervised models, the most important issues involved are the selection of sentence features and the classification models. The classification models used in this paper are Logistic Regression, SVM, NN and Bayes, we used these methods to train the classifier respectively. The features selected from sentences used in classifier training including sentence position, term frequency-inversed sentence frequency (TF-ISF), name entity, numerical information, sentence length, part-of-speech (POS), next we will describe these features respectively.

(1) Sentence position

In a document, the location information of sentences is the most instructive information in the process of generating summaries, especially for the generation of news document summaries. Here are two rough ways to calculate the importance of sentences. The first is that sentences located in front of the document contain more important information, as the position moves backward, the importance of the sentences gradually decreases. The second is to give the sentences at the beginning and end of document more importance. For some particularly long documents, we can calculate the sentence importance based on sections or paragraphs. In this paper, we mainly process the news document, which always put the importance information at the beginning of document, so we use the first strategy to score sentences, and the scoring method is shown in the formula (1).

$$\mathrm{SentPos}(\mathrm{s_i}) = \begin{cases} 1\ if\ \ pos(\mathrm{s_i}) \le 3 \\ 1 - \frac{i-3}{\mathrm{sen\_count(doc)}}\ \ if\ \ pos(\mathrm{s_i}) > 3 \end{cases} \tag{1}$$

where the function $\mathrm{pos(s_i)}$ return the sentence position in the document, function sent_count(doc) return the number of sentences contained in the doc.

(2) TF-ISF

TF-IDF (term frequency-inversed document frequency) is a method for calculating the weight of words in the entire document collection in information retrieval. In this paper, TF-ISF is used. Since we are aiming at the summary generation of a single document, we only perform calculation at the sentence and word level, not include the document level. In this method, TF represents the frequency of words in the entire document, not the frequency in one sentence. The ISF measure the degree of importance of a word in all sentences. If a word appears in all sentences, the ISF value of this word is 0, indicating that the word can't be used to distinguish the importance of sentences. Such as words like “a”, “the”, appeared in majority of sentences, the ISF value is small. If a word appears in a few sentences, then the ISF value is big. This method assumes that the words with high frequency, but appeared in very few sentences, have greater TF-ISF value. If a sentence contains more words with high TF-ISF value, the more likely the sentence be selected to compose the final summary. The method of calculating the TF-ISF value of the word as shown in the formula (2), the weight of sentences as shown in the formula (3).

$$\mathrm{TF - ISF(t_i) = TF(t_i) \times \log\left(\frac{S}{S_{t_i}}\right)} \tag{2}$$

$$\mathrm{TF - ISF(s_i)} = \sum_{\mathrm{t_j \in T}}^{\mathrm{S}} \mathrm{TF - ISF}\left(\mathrm{t_j}\right) \tag{3}$$

where $\mathrm{TF(t_i)}$ represent the word frequency in the document, S represents the number of sentences in document. $\mathrm{S_{t_i}}$ represents the number of sentences containing words $\mathrm{t_i}$, and T represents the set of words in the sentence.

(3) Named entity

Named Entity usually refer to people, objects, geographical locations, time, organization, etc. in the real word. In the news document, this information constitutes the most basic elements of the description of news, so if the sentence contains the information of named entity like these, we think that the sentence may contain import information. This paper recognized the named entity contained in the sentences by Stanford NER toolkit. As a relatively rough estimation method, in this paper, whatever category of named entities, we believe that the information it provides has the same importance. So we assume if a sentence has more named entities, the sentence is more important. The sentence score based named entity is calculated by the formula (4).

$$\mathrm{NER}(s_i) = \frac{\mathrm{entity}(s_i)}{\mathrm{max\_entity}(S)} \tag{4}$$

where function entity(s_i) returns the number of named entity contained in the sentence s_i, S represent the sentences set of document, function max_entity(S) return the maximum of the named entity number in S.

(4) Numerical Information

According to the study of Ferreira, et al. (2014), a sentence containing numerical information may be a good choice for being selected into summary. Usually numerical information means important pieces of information, description of objective facts, important date, time point, etc., and numerical information tends to attract people's attention. For example, in disaster-type news reports, the number of casualties, losses caused, etc., are intuitively presented to the recipient of the information. In this paper, if the proportion of numerical information in a sentence is larger, then more information it contains, and more likely to be included in the summary. Sentences are scored according to numerical information use the formula (5).

$$\mathrm{Num\_Score}(s_i) = \frac{\mathrm{num\_count}(s_i)}{\mathrm{length}(s_i)} \tag{5}$$

where function num_count(s_i) returns the number of numerical information, and the function length(s_i) returns the length of sentence s_i.

(5) Sentence length

A sentence that is too short or too long in the document will have a bad influence on summarization. Too short may contain too little information, or the information it contains is covered by other long sentences. Too long may cause space waste, because a long sentence may have only a small part content contain important information and other content contained information may be irrelevant, and the length of the selected sentence is too long, the chance of choosing other sentences is reduced. In order to satisfy the maximum length constraint, the generated summary cover as much information as possible, and some work optimizes the objective function based on the ILP method. In this paper, the method we used was deleting the longest and shortest sentences when pre-processing. For the rest of the sentences, we assume that the long sentences contain more information and are more likely to be included in the final summary. The scoring technique based on sentence length is shown in the formula (6).

$$\mathrm{Len\_Score}(s_i) = \frac{\mathrm{len}(s_i)}{\mathrm{max\_len}(S)} \tag{6}$$

where function len(s_i) return the length of sentence s_i, function max_len(S) return the most length of sentences in S, the sentence score based on sentence length within (0,1]

(6) POS

The nouns and verbs play a very important role in a sentence. The noun usually plays the role of subject, object, and complement in the sentence, corresponding to the person or thing, and the verb plays the main verb, auxiliary verb, complement, and object in the sentence. By removing modifiers such as adverbs and adjectives, we can still understand the main meaning of a sentence through nouns and verbs, because nouns and verbs can identify important roles and facts in a sentence. We calculate the sentence's score by the formula.

$$\mathrm{POS\_Score} = \frac{\mathrm{num\_POS}(s_i)}{\mathrm{max\_POS}(S)} \tag{7}$$

where S represents the sentences set of document, function num_POS(s_i) returns the verb and noun number in the sentence s_i, max_POS(S) gets the maximum number of nouns and verbs in S.

3.1.2. Unsupervised method

The unsupervised method used in this paper is based graph model. This method mainly has two stage, one is representing the document as graph, the other is scoring the node in graph.

(1) Graph construction

Nodes and Edges are basic elements of graph model, we represent the document using graph, where the nodes in the graph represent the sentences in the document, whether or not the edges are connected is determined by the relevance between the sentences. To determine the relationship of two sentences is key of constructing graph, here we use cosine similarity to measure the relationship. First, we pre-process the sentences in document, including stop-words removing, words stemming, to get the TF-ISF value of words, and then use the vector space model to represent the sentences respectively. Finally, using the vectors we get to calculate the similarity of any two sentences. The formula of getting the similarity between two sentences is shown in (8).

$$\mathrm{sim}\left(s_i, s_j\right) = \frac{\vec{V}(s_i) \cdot \vec{V}\left(s_j\right)}{\left|\vec{V}(s_i)\right|\left|\vec{V}\left(s_j\right)\right|} \tag{8}$$

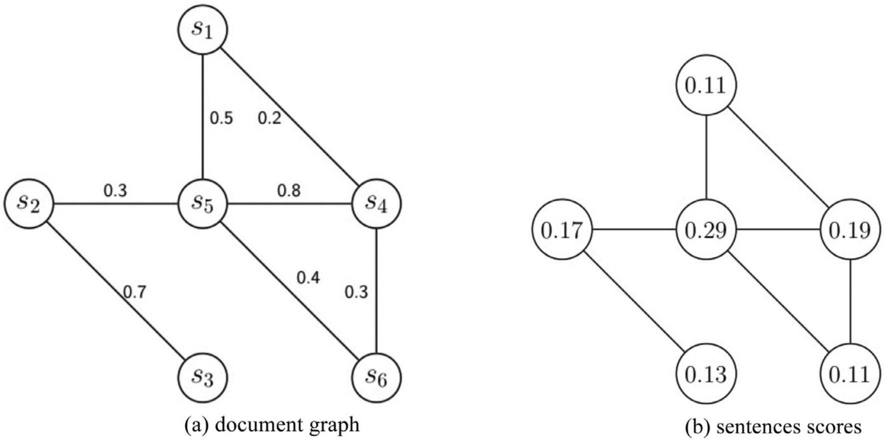
where $\vec{V}(s_i)$ and $\vec{V}(s_j)$ represent the vector of sentence in the document. |x| can return the module of vector. In order to highlight the importance of nodes in the graph and prevent the appearance of a complete graph, we set a similarity threshold θ, if the similarity value is greater than θ, then we will use an edge to connect two nodes, this prevents two nodes with small similarity be connected, which in turn makes the degree of inconsistency among the nodes in the graph, so that the importance of each node can be better distinguished.

(2) Scoring nodes

Follow the first stage, we get a graph from document, then we use the Markov Random Walk to get score of every node, the scoring process we can use the formula shown in (9) to represent.

$$\mathrm{LR}(u) = \frac{d}{N} + (1-d) \sum_{v \in adj[u]} \frac{w(u,v)}{\sum_{z \in adj[v]} w(v,z)} \mathrm{LR}(v) \tag{9}$$

where LR(u) represent the LexRank value of node u, N is the number of nodes in graph, w(u, v) is the similarity value between node u and v, d (d ∈ [0,1)) is damping factor. When Markov Random Walk is performed on the graph, adjacent nodes are selected with probability (1-d), and any node in the graph is randomly selected with probability d/N as the next state. This makes it possible to select nodes with zero degrees and avoid falling into loop traps



(a) document graph

(b) sentences scores

Fig. 2. Illustration of LexRank.

when perform random walks. We can represent the LexRank using matrix notation, which is easy to use Power Method to iterate.

$$\mathrm{P}^{\mathrm{t}} = [\mathrm{dU} + (1 - d)\mathrm{B}]^{\mathrm{T}}\mathrm{P}^{t-1} \tag{10}$$

where U is a square matrix, all elements are 1/N, B is a similarity matrix of all sentences.

For illustrating the process of LexRank, we use the Fig. 2 to show. In the Fig. 2(a), s_1,s_2,..., $s_6$denote the sentences in document, the numbers on the edges indicate the similarity between the sentences. The Fig. 2(b) is the scores of all nodes that have been computed, s_5 has the highest score, because it has the most nodes number linked to it.

3.2. Three methods to combine supervised and unsupervised learning

There are three methods to combine the scoring results get from supervised and unsupervised learning methods, following, we will describe the detail of these methods.

(1) Linear combination method

In this scoring method, we linearly combine the scoring results of the supervised method with the score of the unsupervised method. Synthesize the importance of sentences using the statistical features and the relationship between the sentences. The score of supervised methods is measured by the statistical features of sentences, and the score of unsupervised method is measured by the relationship of sentences. The linearly combination method we can use the formula (11) to describe.

$$\mathrm{score} \;= \alpha \;*\; \mathrm{sup} \;+\; (1 - \alpha)*\mathrm{graph} \tag{11}$$

where the range of α is [0,1], sup is the scores of supervised methods, graph-score is the scores from unsupervised method. If the α is 0, then the final score is equal to the score of unsupervised method. If the α is 1, then we get the score totally from unsupervised method. When combine these two scores, we should standardize the score separately, the method we used is shown in formula 12.

$$\mathrm{standard\_score}(\mathrm{s_i}) = \frac{\mathrm{score}(\mathrm{s_i})}{\sum_{\mathrm{s_j} \in \mathrm{S}} \mathrm{score}(\mathrm{s_j})} \tag{12}$$

We use this method to map the scores of two methods into the range of [0,1], and sum of scores is 1. In this paper, we use the Logistic, SVM, NN and Bayes models to training the classifier for predicting the scores of sentences.

(2) LexRank score as a feature of supervised methods

In this method, we use the sentences scores from LexRank methods mentioned in Section 3.1.2 as a feature for all supervised methods.

(3) Biased-LexRank

In this method, we divide the sentence scoring process into two stages. In the first stage, supervised learning is used to obtain the priori probability that each sentence is selected as the summary content. In the second stage, according the scores we get, we use the scores as the prior knowledge for nodes in graph and the biased-LexRank to score every node. When random walks are performed on the graph, it is preferable to select the node with large prior probability. The statistical features of sentences are integrated into the graph model. This makes it possible to comprehensively consider information such as sentence position, name entity in sentences when rank the nodes in graph.

The biased-LexRank is a variant of the classic LexRank method. In the construction of the graph model, the biased-LexRank is the same with the LexRank. The most essential difference between them is that the initial score of nodes in graph, in LexRank, all nodes are same, but different in biased-LexRank. We use the scores get from supervised methods as the initial scores of nodes in biased-LexRank. So that when random jump, it is preferable to select the nodes that have high score. We can formulate biased-LexRank as (13).

$$\mathrm{LR}(\mathrm{u}) = \frac{\mathrm{d} \cdot \mathrm{p}(\mathrm{u})}{\sum_{\mathrm{z} \in \mathrm{N}} \mathrm{p}(\mathrm{z})} + (1 - d) \sum_{\mathrm{v} \in \mathrm{adj[u]}} \frac{\mathrm{w}(\mathrm{u}, \mathrm{v})}{\sum_{\mathrm{z} \in \mathrm{adj[v]}} \mathrm{w}(\mathrm{v}, \mathrm{z})} \mathrm{LR}(\mathrm{v}) \tag{13}$$

where N is set of all nodes in graph. We can represent (13) using matrix notation as (14).

$$\mathrm{P}^{\mathrm{t}} = [\mathrm{dM} + (1 - \mathrm{d})\mathrm{B}]^{\mathrm{T}}\mathrm{P}^{t-1} \tag{14}$$

where M is the initial score of all nodes, B is the similarity matrix of document, P^{t} is the score of every node in time t. When d is 1, the $P=M$, so the final score of all sentences is the initial score. When 0 < d <1, we get the final score of sentences from matrix M and B combined.

3.3. Sentences selection

When sentences are scored by the methods we proposed, we should select the sentences to compose the final summary of document, this is the last step for extractive summarization. The usual method we select sentences from document is according the score of sentences with descending order. At the same time, in order to

Table 1
Statistic of experimental datasets.

Corpus	Clusters	Documents	Sentences	Words
DUC 2001	30	309	11,026	269,990
DUC 2002	59	576	14,370	348,012

ensure the readability of the content, the final selected sentences are arranged according to the order in the original document. In order to reduce the repeated information between the selected sentences and make the fixed length summary content contains more information. In what follow, we present a method to roughly eliminate the redundancy.

Let S denotes a set of sentences that have been selected as the summary content, s_i is a candidate sentence of summary, we use the cosine similarity to determine whether add s_i into S. The formula is shown in (15).

$$\mathrm{Sim}(S, s_i) = \max_{s_j \in S} \ \mathrm{Cosine}\left(s_j, s_i\right) \tag{15}$$

By setting the threshold of redundancy θ, if the $\mathrm{Sim}(S, s_i)$ is big than θ, we neglect the sentence s_i, otherwise, add the s_i into S. Repeat this process, until the length of selected sentences over the maximum length. It is not good if the threshold is set too high or too low. If it is set too high, the purpose of removing redundancy cannot be achieved. If it is set too low, some useful information may be deleted. In our experiments, we set the threshold $\theta = 0.7$.

4. Experimental analysis

In this section, we evaluate the effectiveness of our proposed methods and candidate algorithms on two datasets.

4.1. Datasets

In this paper, DUC2001 and DUC2002 corpus are selected as datasets for English single document summarization. DUC[1] (Document Understanding Conference) is a famous international evaluation conference in the field of automatic summarization, aiming at promoting the progress of automatic summarization. DUC2001 including 30 clusters, 309 documents, and DUC2002 including 59 clusters, 576 documents. Each cluster consists 5 to 15 documents and every document has at least 10 sentences. Every cluster belongs to a topic, the topics of two datasets include natural disasters, social news, political news, scientific events, personal biography, etc. The detail of the two datasets are shown in the Table 1.

4.2. Evaluation standard

In this paper, the ROUGE (Lin, 2004) method is used to evaluate the quality of the summary, which is based on n-gram cooccurrence statistical method. The basic idea is to generate artificial summary by several experts to form a standard summary set, to compare the automatic summaries generated by the system and the standard summaries generated by the manual, and to evaluate the quality of the summary by the statistics of the number of basic units (n-gram, word sequence and word pairs) overlapped between the two. The evaluation indexes used in these experiments are ROUGE-1, ROUGE-2, ROUGE-SU4. For the sake of generality, we use ROUGE-N to represent it, which is define as

$$ROUGE - N = \frac{\sum_{s \in \{eval-summary\}} \sum_{gram_N \in S} Count_{match}(gram_N)}{\sum_{s \in \{eval-summary\}} \sum_{gram_N \in S} Count(gram_N)}$$

where N is the length of n-gram, {eval-summary} denotes the reference summary, which is a standard summary set obtained from experts. $\mathrm{Count}_{\mathrm{match}}(\mathrm{gram}_{\mathrm{N}})$ get the number of n-gram that appeared in the candidate summary and the reference summary. $\mathrm{Count}(\mathrm{gram}_{\mathrm{N}})$ gets the number of n-gram that appeared in the reference summary. ROUGE-SU4 is based on skip bigram with a maximum skip distance of 4.

4.3. Candidate algorithms

In order to verify the effectiveness of summarization methods we proposed, we use five basic methods, including Lead, Random, LSA, LexRank and Supervised, to compare the effects on two datasets.

(1) Lead: The main idea of this method is that sentences located in the front of document are more important, so we select the first N words in the document as summary. Here N is 100.
(2) Random: First, we shuffle sentences in the document, then select the sentences using the Lead method.
(3) LSA: It is a method that based latent semantic analysis, we mainly re-implement the method proposed by Gong and Liu (2001).
(4) LexRank: As described in Section 3.1.2, we followed this method to get the summary of document.
(5) Supervised: Using supervised learning methods to get summary of document. In classifiers training process, Logistic, Neural Network, SVM, and Bayes methods were used. The features we used for training are from the six features described in Section 3.1.1.

4.4. Experimental evaluation

4.4.1. Overall evaluation

First, we use the candidate methods and proposed methods to get summary from documents, and use the ROUGE-1, ROUGE-2, ROUGE-SU4 values to evaluate the experimental results. The results we get from DUC2001 and DUC2002 are shown in Tables 2 and 3. Here, because the results of the first and third combination methods are affected by the parameter α, so we only choose the best value of all methods we get under the parameter α, and the effect of parameter α on experimental results will be analyzed in Section 4.4.2.

In the Tables 2 and 3, *Linear-x-T* is the first combination method we proposed, *x-with-T* is the second, and *Sup-x-T* is the third,

Table 2
DUC2001 overall comparison in terms of multiple metrics.

Methods	ROUGE-1	ROUGE-2	ROUGE-SU4
Lead	0.4401	0.19494	0.2162
Random	0.37797	0.12058	0.15269
LSA	0.40293	0.17342	0.19341
LexRank	0.40883	0.14995	0.17916
Sup-Logistic	0.43285	0.18476	0.20666
Sup-SVM	0.43589	0.18588	0.20766
Sup-NN	0.44149	0.19485	0.21489
Sup-Bayes	0.41163	0.1638	0.18802
Sup-Logistic-T	0.43739	0.18474	0.20706
Sup-SVM-T	0.44069	0.19063	0.2123
Sup-NN-T	0.44483	0.19522	0.21618
Sup-Bayes-T	0.41954	0.16497	0.1911
Linear-Logistic-T	0.43511	0.18755	0.20897
Linear-SVM-T	0.44201	0.19137	0.21275
Linear-NN-T	0.4462	0.19661	0.21733
Linear-Bayes-T	0.4186	0.16511	0.19028
Logistic-with-T	0.43521	0.18827	0.20954
SVM-with-T	0.44225	0.19433	0.21482
NN-with-T	0.44337	0.19685	0.26683
Bayes-with-T	0.41166	0.16375	0.18801

[1] https://duc.nist.gov.

Table 3
DUC2002 overall comparison in terms of multiple metrics.

Methods	ROUGE-1	ROUGE-2	ROUGE-SU4
Lead	0.46688	0.21447	0.22994
Random	0.40126	0.1396	0.1676
LSA	0.42869	0.17692	0.18654
LexRank	0.43815	0.17938	0.20116
Sup-Logistic	0.46106	0.20848	0.22381
Sup-SVM	0.46371	0.20981	0.22537
Sup-NN	0.47113	0.21712	0.23162
Sup-Bayes	0.4407	0.18659	0.20586
Sup-Logistic-T	0.46416	0.21006	0.22586
Sup-SVM-T	0.46571	0.21174	0.22696
Sup-NN-T	0.47229	0.2178	0.23225
Sup-Bayes-T	0.4475	0.19232	0.21112
Linear-Logistic-T	0.4643	0.21057	0.22627
Linear-SVM-T	0.46605	0.21201	0.22711
Linear-NN-T	0.47255	0.21778	0.2323
Linear-Bayes-T	0.44687	0.19079	0.21043
Logistic-with-T	0.46518	0.21168	0.22677
SVM-with-T	0.47196	0.21758	0.23231
NN-with-T	0.47239	0.2177	0.23212
Bayes-with-T	0.44073	0.18661	0.20589

where *x* represents Logistic, SVM, NN, Bayes respectively. From the tables, we can find that the results getting from the simple Lead-based method is better than most of methods, this is shown sentences in the front of document are more likely to be selected as summary content. Random method is the worst in all methods we used, indicating that it does not work well without any prior knowledge for sentence selection. When using the same features and different training methods to generate the document summary, the neural network method achieves the best results. In DUC2001, ROUGE-1 value is 0.44149, ROUGE-2 value is 0.19485, in DUC2002, the ROUGE-1 value is 0.47113 and the ROUGE-2 value is 0.21712, which also shows that use the same features but different training model, the summary we get has a big difference.

In this experiment, we used four supervised models to predict sentences scores, and then used the scores as the prior of nodes in graph model. It can be found that in DUC2001 and DUC2002 datasets, when using biased method to generate summary, the quality is better than any of LexRank or supervised method. Such as Sup-NN-T method is better than LexRank and Sup-NN. Moreover, according to the ROUGE-1 results of four supervised methods, Sup-NN>Sup-SVM>Sup-Logistic>Sup-Bayes, and in biased-LexRank, we can get Sup-NN-*T*>Sup-SVM-*T*>Sup-Logistic-*T*>Sup-Bayes-T. From here, we can infer that in biased-LexRank, if the prior value of nodes in a graph is given the better, the sentences ranking effect of the model is better. When combine two scoring methods linearly, the effect is improved over the LexRank and Supervised methods in both datasets, and get the best result in all methods, which indicates scoring the sentences from statistical features and sentences relationship comprehensively are more accurate. When using the scores get from LexRank method as a feature added into the supervised models, we can get a better summary of document sets than before. Among them, the effect of SVM method has the most significant improvement, while the Bayes method has almost no effect.

Based on the results of Tables 2 and 3, the results of summary generation are better than those using only statistical features or sentence relationship after combining sentence relationship and statistical features. The reasons are as follows. For the first combination method, there are two kinds of scores we get when evaluate a sentence, one is from statistical characteristics, the other is from the relationship between sentences. After linear combination, it can evaluate the importance of a sentence more comprehensively. For the second combination method, new features are added to make the model better able to distinguish between important and non-important sentences, but in this paper, we have not evaluated the impact of each feature on sentence scoring. The third combination method has the most obvious effect on the improvement of LexRank method, because the Biased-LexRank method can choose the sentences with higher priori values when it makes random jumps by adding supervised method as a priori knowledge. At the same time, for those nodes with lower priori values but more degrees in the graph, it can also select them by iteration algorithm.

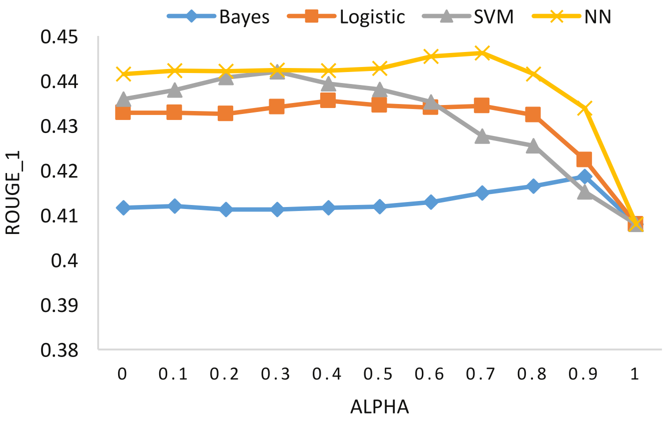


Fig. 3. DUC2001 ROUGE-1 value with α change in linear combination method.

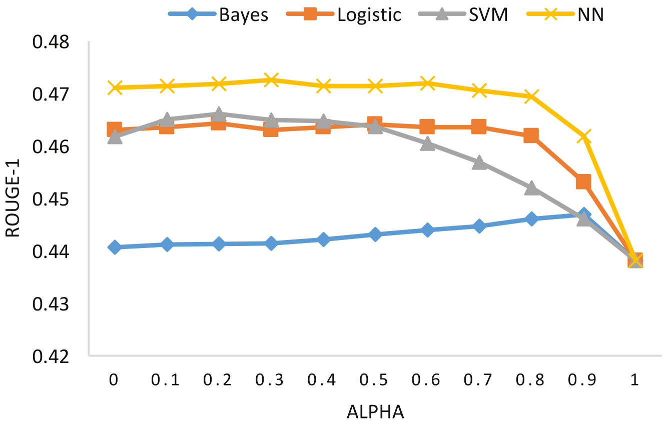


Fig. 4. DUC2002 ROUGE-1 value with α change in linear combination method.

4.4.2. Parameter analysis

In the first method, the parameter α is used to balance the proportion of sentences score between the LexRank method and the Supervised method. In the third method, the parameter α is used to control the probability of random jump in graph model, if the α value is larger, the probability of using the random jump to select the node is bigger. In order to study the influence of the parameter α value on summarization, here we set the ranges of α in the first method from 0 to 1 and the second method from 0 to 0.9.

Figs. 3 and 4 are the ROUGE-1 value of DUC2001 and DUC2002 datasets with the change of α. We observed that when α is 0, the ROUGE-1 value is the same with supervised methods, and when α is 1, the ROUGE-1 value is the same with the LexRank method. By combining the scores from two methods linearly, the ROUGE-1 values of the summary are increased. In addition to the SVM method, when the value of α is less than 0.8 in other three methods, the ROUGE-1 values are less affected by the α, so we can infer the sentences score from SVM is easy to affected by the LexRank score.

Figs. 5 and 6 are the DUC2001 and DUC2002 datasets used the third combination method, the ROUGE-1 values change with parameter α. In DUC2001, the NN and Logistic get the best result

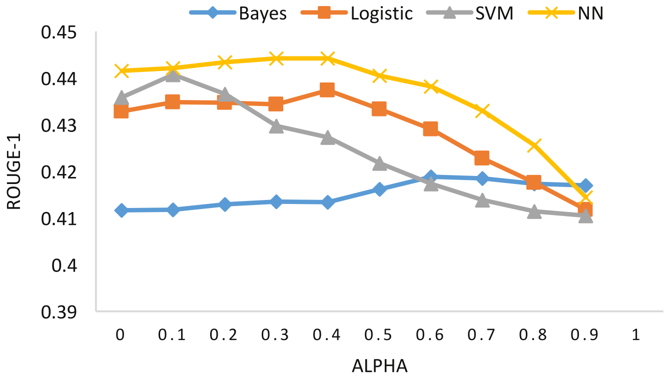


Fig. 5. DUC2001 ROUGE-1 value with α change in biased-LexRank.

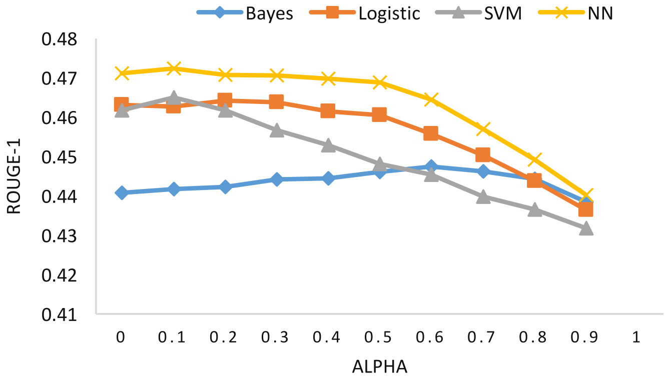


Fig. 6. DUC2002 ROUGE-1 value with α change in biased-LexRank.

when α is 0.4, the values are 0.44423 and 0.43739, and the SVM gets the best result 0.44069 at 0.1, Bayes gets the best result 0.41881at 0.6. In DUC2002 datasets, NN and SVM get the best result when α is 0.1, the values are 0.47229 and 0.4649, Logistic get the best value 0.4616 at 0.3 and Bayes get the best value 0.4474 at 0.6. In this method, when the value of α is smaller, the probability of random jump is larger, and the random jump is based on the prior probability (the sentences score of supervised methods) to select the nodes, so the node with high probability is more likely to be selected. With the increasing of α value, in addition to the Bayes method, the ROUGE-1 values of other methods show a decreasing trend, but all the methods are better than the LexRank method, indicating that there is a prior guidance can improve the LexRank method. However, when the α value is too large, the LexRank method dominates the score of sentences, that the scores are biased to LexRank, so the ROUGE-1 value decreases obviously. And when α is smaller, the supervised method dominates the scoring, so the ROUGE-1 value biased to supervised methods. But the SVM method is easy affected by the α, when α takes 0.1, the positive influence is produced, so the ROUGE-1 value is better than both supervised and LexRank, but when α takes other values, the ROUGE-1 value decreases quickly.

5. Conclusion

This paper presents three methods of combining supervised learning with unsupervised learning to generate single document summary. The first method combines the scores of two methods linearly. The second one regards the scores of unsupervised methods as a feature of supervised learning. The third one regards the scores of supervised learning as a priori value of nodes in the graph model. When using these three methods to score sentences, the statistical characteristics of sentences and the relationship between sentences are integrated, which can evaluate the importance of sentences in documents more comprehensively, thus improving the accuracy of sentence scoring.

The experimental results on two data sets of DUC2001 and DUC2002 show that the three combination methods proposed in this paper have improved compared with Lead, LSA, LexRank and the supervised methods under the ROUGE. At the same time, the best result obtained from the third combination method shows that the original LexRank algorithm is greatly improved after adding the prior value to the graph model.

Although our proposed methods have performed well, there exits much room for improvement. From the experimental results, we can infer that good priori values for nodes in graph can improve the effect of summarization. Hence, how to get better prior values for nodes in graph is the first future work we need to consider. Another meaningful future work is to use the proposed summarization methods for long documents, such as academic papers, audit reports, or books etc. Last but not least, we would like to explore more text modeling methods to represent documents such as complex networks or neural networks.

Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Credit authorship contribution statement

Xiangke Mao: Writing - original draft, Writing - review & editing. **Hui Yang:** Writing - review & editing, Formal analysis, Resources. **Shaobin Huang:** Investigation, Methodology, Supervision, Formal analysis. **Ye Liu:** Software, Validation. **Rongsheng Li:** Software, Validation.

References

Alguliyev, R. M., Aliguliyev, R. M., Isazade, N. R., Abdi, A., & Idris, N. (2019). COSUM: Text summarization based on clustering and optimization. *Expert Systems, 36*(1), e12340. https://doi.org/10.1111/exsy.12340.

AL-Khassawneh, Yazan, Salim, Naomie, & Jarrah, Mu'tasem (2017). Improving triangle-graph based text summarization using hybrid similarity function. *Indian Journal of Science and Technology, 10*(8), 1–15.

Amancio, D. R., Nunes, M. G., Oliveira, O. N., Jr, & Costa, L. D. F. (2012). Extractive summarization using complex networks and syntactic dependency. *Physica A: Statistical Mechanics and its Applications, 391*(4), 1855–1864.

Christian, Hans, Agus, Mikhael Pramodana, & Suhartono, Derwin (2016). Single document automatic text summarization using term frequency-inverse document frequency (TF-IDF). *ComTech: Computer, Mathematics and Engineering Applications, 7*(4), 285–294.

Conroy, J. M., & O'leary, D. P. (2001). Text summarization via hidden markov models. *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM. (pp. 406–407).

Dutta, M., Das, A. K., Mallick, C., Sarkar, A., & Das, A. K. (2019). *A graph based approach on extractive Summarization. In Emerging technologies in data mining and information security*. Singapore: Springer. (pp. 179–187).

Erkan, G. (2006). Using biased random walks for focused summarization. In *Proceedings of Document Understanding Conference (DUC)*.

Erkan, G., & Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization[J]. *Journal of Artificial Intelligence Research, 22*, 457–479.

Fang, C., Mu, D., Deng, Z., et al. (2017). Word-sentence co-ranking for automatic extractive text summarization[J]. *Expert Systems with Applications, 72*, 189–195.

Fattah, M. A., & Ren, F. (2009). GA, MR, FFNN, PNN and GMM based models for automatic text summarization[J]. *Computer Speech & Language, 23*(1), 126–144.

Fattah, Mohamed Abdel (2014). A hybrid machine learning model for multi-document summarization. *Applied intelligence, 40*(4), 592–600.

Ferreira, R., de Souza Cabral, L., Lins, R. D., et al. (2014). Assessing sentence scoring techniques for extractive text summarization[J]. *Expert systems with applications, 40*(14), 5755–5764.

García-Hernández, René Arnulfo, & Ledeneva, Yulia (2013). Single extractive text summarization based on a genetic algorithm. *Mexican Conference on Pattern Recognition*. Springer. (pp. 374–383).

Glavaš, Goran, & Šnajder, Jan (2014). Event graphs for information retrieval and multi-document summarization. *Expert Systems with Applications, 41*(15), 6904–6916.

Gong, Yihong, & Liu, Xin (2001). Generic text summarization using relevance measure and latent semantic analysis. *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM. (pp. 19–25).

Gupta, Pankaj, Pendluri, Vijay Shankar, & Vats, Ishant (2011). Summarizing text by ranking text units according to shallow linguistic features. *13th International Conference on Advanced Communication Technology (ICACT2011)*. IEEE. (pp. 1620–1625).

Khan, Atif, Salim, Naomie, & Kumar, Yogan Jaya (2015). A framework for multi-document abstractive summarization based on semantic role labelling. *Applied Soft Computing, 30*, 737–747.

Kupiec, J., Pedersen, J., & Chen, F. (1995). A trainable document summarizer. In *Proceedings of the 18th Annual International ACM CIGIR Conference on Research and Development in Information Retrieval* (pp. 55–60).

Li, Chen, Qian, Xian, & Liu, Yang (2013). Using supervised bigram-based ILP for extractive summarization. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1004–1013).

Li, W. (2015). Abstractive multi-document summarization with semantic information extraction. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 1908–1913).

Lin, C. Y. (1999). Training a selection function for extraction. *Proceedings of the eighth international conference on Information and knowledge management*. ACM. (pp. 55–62).

Lin, Chin-Yew (2004). ROUGE: A Package for Automatic Evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)*.

Liu, Chien-Liang, et al. (2012). Movie rating and review summarization in mobile environment. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42*(3), 397–407.

Luhn, H. P. (1958). The automatic creation of literature abstracts[J]. *IBM Journal of Research and Development, 2*(2), 159–165.

Mallick, Chirantana, et al. (2019). Graph-Based Text Summarization Using Modified TextRank. *Soft computing in data analytics*. Singapore: Springer. (pp. 137–146).

Meena, Yogesh Kumar, & Gopalani, Dinesh (2015). Evolutionary algorithms for extractive automatic text summarization. *Procedia Computer Science, 48*, 244–249.

Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing order into texts. In *Proceedings of the 2004 conference on empirical methods in natural language processing*.

Nallapati, Ramesh, Zhai, Feifei, & Zhou, Bowen (2017). Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. *Thirty-First AAAI Conference on Artificial Intelligence*.

Oliveira, H., Ferreira, R., Lima, R., et al. (2016). Assessing shallow sentence scoring techniques and combinations for single and multi-document summarization. *Expert Systems with Applications, 65*, 68–86.

Ouyang, You, et al. (2011). Applying regression models to query-focused multi-document summarization. *Information Processing & Management, 47*(2), 227–237.

Ouyang, You, et al. (2010). A study on position information in document summarization. In *Proceedings of the 23rd international conference on computational linguistics: Posters. Association for Computational Linguistics* (pp. 919–927).

Page, Lawrence, et al. (1999). *The pagerank citation ranking: Bringing order to the web*. Stanford InfoLab.

Ribeiro, Ricardo, & Martins de Matos, D. (2011). Centrality-as-relevance: Support sets and similarity as geometric proximity. *Journal of Artificial Intelligence Research, 42*, 275–308.

Ribeiro, Ricardo, et al. (2013). Self reinforcement for important passage retrieval. *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*. ACM. (pp. 845–848).

Saziyabegum, Saiyed, & Sajja, Priti S. (2016). Literature Review on Extractive Text Summarization Approaches. *International Journal of Computer Applications, 156*(12), 28–36.

Shah, Rajiv Ratn, et al. (2016). Leveraging multimodal information for event summarization and concept-level sentiment analysis. *Knowledge-Based Systems, 108*, 102–109.

Shetty, Krithi, & Kallimani, Jagadish S. (2017). Automatic extractive text summarization using K-means clustering. *2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)*. IEEE. (pp. 1–9).

Tohalino, J. V., & Amancio, D. R. (2018). Extractive multi-document summarization using multilayer networks. *Physica A: Statistical Mechanics and its Applications, 503*, 526–539.

Tohalino, J. V., & Amancio, D. R. (2017). Extractive multi-document summarization using dynamical measurements of complex networks. *2017 Brazilian Conference on Intelligent Systems (BRACIS)*. IEEE. (pp. 366–371).

Wan, Xiaojun, & Yang, Jianwu (2008). Multi-document summarization using cluster-based link analysis. *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*. ACM. (pp. 299–306).

Wong, Kam-Fai, Wu, Mingli, & Li, Wenjie (2008). Extractive summarization using supervised and semi-supervised learning. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1. Association for Computational Linguistics* (pp. 985–992).

Yang, Y., Bao, F., & Nenkova, A. (2017). Detecting (un)important content for single document news summarization. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2* Short Papers. (pp. 707–712).

Yao, Jin-ge, Wan, Xiaojun, & Xiao, Jianguo (2017). Recent advances in document summarization. *Knowledge and Information Systems, 53*(2), 297–336.

Yao, Kaichun, et al. (2018). Deep reinforcement learning for extractive document summarization. *Neurocomputing, 284*, 52–62.

Yousefi-Azar, Mahmood, & Hamey, Len (2017). Text summarization using unsupervised deep learning. *Expert Systems with Applications, 68*, 93–105.