Extractive Multi-document Summarization using K-means, Centroid-based Method, MMR, and Sentence Position

Hai Cao Manh
Hanoi University of Science and Technology, Vietnam
caomanhhaipt@gmail.com

Huong Le Thanh
Hanoi University of Science and Technology, Vietnam
huonglt@soict.hust.edu.vn

Tuan Luu Minh
Hanoi University of Science and Technology, Vietnam
tuanlm@neu.edu.vn

ABSTRACT

Multi-document summarization is more challenge than single-document summarization since it has to solve the problem of overlapping information among sentences from different documents. Also, since multi-document summarization dataset is rare, methods based on deep learning are difficult to be applied. In this paper, we propose an approach to multi-document summarization based on k-means clustering algorithm, combining with centroid-based method, maximal marginal relevance and sentence positions. This approach is efficient in finding salient sentences and preventing overlapping between sentences. Experiments using DUC 2007 dataset show that our system is more efficient than other researches in this field.

CCS CONCEPTS

• Information systems → Summarization • Computing methodologies → Natural language processing

KEYWORDS

Extractive multi-document summarization, k-means, clustering algorithm, centroid-based, maximal marginal relevance, sentence position

ACM Reference format:

Hai Cao Manh, Huong Le Thanh, Tuan Luu Minh. 2019. Extractive Multi-document Summarization using K-means, Centroid-based Method, MMR, and Sentence Position. In *SoICT '19: The Tenth International Symposium on Information and Communication Technology*, *December 4–6, 2019, Hanoi – Ha Long Bay, Viet Nam.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3368926.3369688

1 Introduction

Nowadays, news is provided on the Internet in a large quantity. Several news mention to the same topic with some modifying details. There are demands to summarize all of such news in order to have concise information about the topic. Multi-document summarization is the solution for this problem.

Most researches on multi-document summarization follow extractive approaches by selecting sentences that best described the main idea of input documents and combining them to generate the summarization. Hu et al. [13] categories techniques to compute sentences' scores into three groups: feature-based methods [3, 5], lexical chain-based methods [16, 18], and graph-based ones [7, 19, 24]. The feature-based methods measure a sentence's score based on its features such as its position, its length, the keywords in this sentence, etc. The lexical chain-based methods define lexical chains by a coherent sequence of related nouns, verbs, etc., Sentences are then scored according to the lexical chains they belong to. The graph-based methods apply the idea of PageRank algorithms [20] to construct a graph that reflexes the semantic relationships among sentences. Sentences are then scored and extracted based on that graph.

Recently, several researches on document summarization move to deep learning approaches, as these approaches provide better results than traditional ones. A disadvantage of these approaches is that they require a large dataset for training. Such a dataset is available for single document summarization, however, such a dataset for multi-document summarization is not available. Because of that, deep learning approaches are mainly used for single-document summarization. Most research on multi-document summarization still bases on select salient sentences from input documents.

A problem in multi-document summarization is the information overlap among salient sentences, which causes redundancy in the summary. The methods based on calculating sentence scores do not solve this problem. To deal with the problem of data scarcity and information redundancy, we propose an approach to extractive multi-document summarization using the k-means clustering algorithm combining with the centroid-based method, MMR, and sentence position. Experimental results show that our system is significantly efficient comparing to existing methods on multi-document summarization.

SoICT 2019, December 4–6, 2019, Hanoi - Ha Long Bay, Viet Nam
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-7245-9/19/12…$15.00
https://doi.org/10.1145/3368926.3369688

2 Related works

Early researches on multi-document summarization group similar sentences from input documents into clusters and picking centroid sentences of each group to put in the summary [10, 14]. The cosine similarity measure is often used to compute the similarity between a pair of sentences, where sentences are represented as a weighted vector of tf.idf. The sentence with most frequent terms is considered as the centroid of a cluster. However, since this method does not consider the semantic meaning of each word in the text, the summary may not good at the semantic aspect. Another problem with this approach is that some clusters may contain unimportant information from the input documents.

Another strategy in multi-document summarization is to construct a graph based on the similarity among sentences and then calculate sentences' weight from that graph [7, 9, 19, 24]. This approach is often combined with word-weight adjustment techniques, which is one of the most important factors affecting summarization quality. The word-weight adjustment can be done by exploiting semantic relation between words using word embedding or WordNet [8]. Although this approach can identify important sentences across input documents, it cannot "understanding" the text since sentences are represented as bags of words. Because of that, the final summary may not be informative enough.

Several researches apply a centroid-based approach to generate text summary [6, 22]. This approach generates cluster centroids, which consist of words that are central to all input texts. The summary is generated by collecting sentences that contain words from the centroid. A disadvantage of this approach is that it is not good at preventing information redundancy in the summary. To solve this problem, Carbonell and Goldstein [11] introduced a Maximal Marginal Relevance (MMR) measure to produce summaries. However, this approach is sensitive to the selective topic sentences and does not guarantee to exclude unimportant sentence in the summary. Our solution to all of the above-mentioned problems will be discussed at the end of this section.

Instead of extracting the highest score sentences based on word-weight, several works have been done differently by analyzing word's latent semantic. The singular value decomposition (SVD) is used by Gong and Liu [25] to select the highest-ranking sentences. Arora and Ravindran [17] apply the Latent Dirichlet Allocation (LDA) to extract topics from the input documents and to generate a summary by selecting leading sentences representing for these topics.

Some research follows the approach of reconstructing sentences in the input documents to generate a summary [12, 15, 26]. He et al. [26] introduce two types of reconstruction (linear and nonnegative) and develop an efficient optimization methods for them. Le and Mikolov [15] reconstruct documents by summary sentences using a neural network model, selecting summary sentences to minimize reconstruct errors.

Although several works have been carried out in the task of multi-document summarization, generating a summary that best describes the input documents and containing minimum redundancy is still challenging. To deal with that problem, we propose an approach to extractive multi-document summarization that combines the k-means algorithm with a centroid-based method, maximal marginal relevance measure, and sentence position. The k-means algorithm is used to cluster sentences from the input documents. To overcome the problem of knowledge-poor in k-means, word embedding is integrated into the system to get the semantic relationships among sentences. To address the problem of picking sentences representing for unimportant clusters, the centroid-based method is used to find the most centroid sentences and to eliminate clusters that have poor information. In addition, the MMR is applied to eliminate the information overlapping among sentences in the summary. Finally, the final summary is generated with a reasonable chronological order based on sentence positions. Our proposed architecture for a multi-document summarization system is introduced next.

3 Proposed architecture

Our proposed multi-document summarization system consists of two main modules:

(1) **Input processing**: This module processes the input documents by removing stop words, word stemming, and converting each input sentence into a vector, which will be used as the input for the summarizing module. Three methods of representing input sentences have been used. These methods are introduced in Section 4.1.

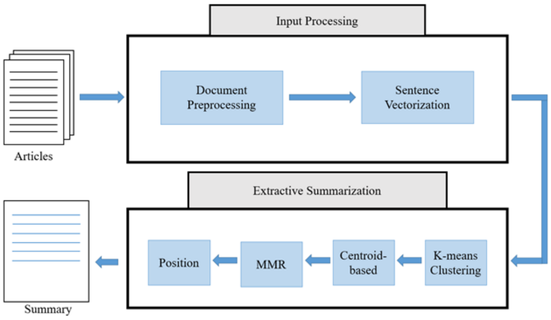

Figure 1: Our proposed architecture for a multi-document summarization system

(2) **Extractive summarization**: This module takes as input vectors of sentences and generates a summary by extracting the most informative sentences. We propose several summarizing models which base on k-means and combine with different methods or measures in order to find the best summarization system, including the centroid-based method, maximal marginal relevance measure, and sentence position in the original document.

The proposed architecture is represented in Figure 1. The basic components of our summarization system will be introduced next.

4 Basic components

4.1 Sentence vectorization

In our system, each input sentence is represented as a vector. The simplest way to represent sentences is to use the bag-of-words (BoW) model. In the BoW model, each word is represented as a one-hot vector whose dimensions is equal to the word vocabulary size. The vector consists of 0s in all elements with the exception of a single 1 in an element corresponding to word index in the vocabulary. Each sentence is a vector that has the same size as word vectors. Each element's value in the sentence vector is the sum of all values of elements at the same position in its word vectors. For example, if a word appears twice in the sentence, the value of the corresponding element in the sentence vector is 2.

A weakness of the BoW model is that it does not contain information about the importance of words in the document. To solve this problem, instead of using a word count value for the element at its word index position in the sentence vector as in BoW, our new model uses tf.idf (term frequency-inverse document frequency). Here, tf is the term frequency of each word in a document, and idf (inverse document frequency) is the inverse frequency of that word in the DUC2007 dataset. The new model is called BoW_tf.idf model.

The method of using a bag of words to represent input sentences does not contain the semantic meaning of the sentence. Therefore, this representation cannot help in computing the semantic relation among sentences. Word embedding is integrated into our system for that purpose.

Word embedding. Word embedding is a vector representation of a particular word, constructed by learning from a large text corpus. It is capable of capturing semantic relation with other words in the vocabulary. Google's pre-trained Word2vec model[1] is used for that purpose.

There are two common architectures of word embedding: Continuous Bag-of-Words (CBoW) and Skip-gram.

- CBoW model [1]: This model takes the context of each word as input data and tries to predict the word corresponding to this context. CBoW model learns word embedding by predicting the current word based on its context. More specifically, we use the one-hot vector of the input word and calculate the output errors by comparing to the one-hot vector of the target word. During the target word prediction, the model will learn to represent the vector of the target word.

- Skip-gram model [1]: Learning word embedding by predicting the words around for the current word. At the skip-gram architecture, there is only one input word for a training case, and there are multiple output contexts for each input word.

4.2 Centroid-based method

The centroid-based method is often used in text summarization to determine salient sentences in a document set. A sentence vector is represented based on the *tf.idf* of words in the sentence. A word is a centroid if its *tf.idf* value is greater than a given threshold θ_{sent}. The sentences containing multiple centroid words will be extracted to the summary. Our centroid-based approach to multi-document summarization is presented as follows:

Algorithm 1. Centroid-based algorithm

Input: A set of sentences

Output: A summary from the input set of sentences

Algorithm:

1. Represent the input set of sentences using the BoW_tf.idf model.
2. Define the centroid vector c: The size of the centroid vector equals to the vocabulary size. Each element a_w of the centroid vector c represents for a word w in the vocabulary. The value of a_w is calculated by the following formula:

$$a_w = \sum_{s \in S} tf.idf_{w,s} \quad (1)$$

where S is the set of sentences from the input documents; $tf.idf_{w,s}$ is the tf.idf of the word w in the sentence s.

3. Calculating the similarity among the sentence vector s and the centroid vector c. If a sentence whose similarity with the centroid is smaller than a given threshold θ_{sent}, this measure will be set to 0. The formula to calculate the similarity between a sentence s and the centroid c is computed as follows:

$$sim(s,c) = \frac{(1 - cosine(s,c)) + 1}{2} \quad (2)$$

4. Sorting the set of sentences in descending order of their similarity with the centroid.
5. The summary is generated by getting sentences in the above set by its sorting orders. Also, this sentence must have minimum overlapping information with sentences already in the summary. To prevent overlapping, a sentence s is selected if it satisfies the following condition:

$$\forall v \in V, \frac{(1 - cosine(s,v)) + 1}{2} \le \theta_{sim} \quad (3)$$

where V is the set of sentences that have been included in the summary, θ_{sim} is the threshold for the similarity between two sentences, and

$$cosine(s,v) = 1 - \frac{s.v}{\| s \|_2 \| v \|_2} \quad (4)$$

The centroid-based method for the document summarization problem eliminates the information overlap in the summary by using the cosine measure. However, the summary quality depends on choosing the thresholds θ_{sent}, θ_{sim}, and has not considered the semantic similarity among sentences yet.

[1] Available at https://code.google.com/archive/p/word2vec/

4.3 Maximal Marginal Relevance

The original Maximal Marginal Relevance (MMR) [11] was proposed to solve information retrieval (IR) problem to measure the relevance between the user query Q and sentences in the document.

This measure is calculated by the formula:

$$MMR \overset{def}{=} Arg \max_{D_i \in C \setminus S} \left[\lambda \left(Sim_1\left(D_i, Q\right) - \left(1-\lambda\right) \max_{D_j \in S} Sim_2\left(D_i, D_j\right) \right) \right] \quad (5)$$

where C is the set of sentences from the input documents, S is the existing sentences in the summary, Sim_1 is the similarity between the considering sentence and the query, Sim_2 is the similarity between the considering sentence and the existing sentences in the summary (Sim_2 can be equal to Sim_1), and λ is a parameter ($\lambda \in [0, 1]$). The parameter value λ is chosen depending on each problem. If it is necessary to return information around the query, the parameter λ is adjusted with a smaller value. If the result needs to be diverse, the parameter λ is adjusted with a greater value. Higher MMR means the considered item is both relevant to the query and contains minimal similarity to previously selected items.

To apply MMR to the document summarization, we redefine the formula to calculate MMR as follows:

$$MMR \overset{def}{=} Arg \max_{D_i \in C \setminus \{S, Q\}} \left[\lambda \left(Sim_1\left(D_i, Q\right) - \left(1-\lambda\right) \max_{D_j \in S} Sim_2\left(D_i, D_j\right) \right) \right] \quad (6)$$

where *C* is the sentential set that was selected from the previous algorithm, *Q* is selected from the set *C* that is the sentence that best described the main idea of input documents, *S* is the sentential set that included in the summary, Sim_1, Sim_2 are the similarities between the two sentences that are calculated to the formula:

$$Sim_1\left(u, v\right) = Sim_2\left(u, v\right) = \frac{\sum_{w \in v} tf_{w,u} tf_{w,v} \left(idf_w\right)^2}{\sqrt{\sum_{w \in u} \left(tf_{w,u} idf_w\right)^2}} \quad (7)$$

where *u*, *v* are two sentences that we need to calculate the similarity, $tf_{w,u}$ is the term frequency of the word *w* in the sentence *u*, idf_w is the importance of the word *w*, and *λ* is the chosen parameter.

The main point of applying MMR is to eliminate redundant information in the summary. In order to do that, three steps needed to be carried out are:

(i) Determining the main topics of the input documents;
(ii) Finding sentences relevant to the main topics;
(iii) Eliminating redundant sentences whose similarity with existing sentences in the summary is larger than a certain threshold.

5 Implementing scenarios

We propose several summarization models from the basic k-means clustering algorithm, including:

1. K-means with relative sentence positions
2. K-means with sentence positions
3. K-means with MMR and sentence positions
4. K-means with centroid-based method, MMR and sentence positions

Since the centroid-based method and the MMR do not concern the semantic meaning of text, we apply word embedding in representing input documents and use it as the input of the k-means algorithm.

Scenario 1: K-means with relative positions. In this implementation, each word is represented as a word embedding vector. Each sentence is also represented as an embedding vector, which is the total embedding vectors of words in that sentence. The k-means algorithm takes as input embedding vectors of sentences from input documents and groups these sentences into clusters.

Sentences that are closest to the center of each cluster will be put in the summary. Since each cluster represents a different topic of the input documents, sentences extracted by this way rarely have information overlap. To guarantee the summary length, the number of clusters should be equal to the number of the desired sentences in the summary.

Each extracted sentence is then put in the summary based on its position's score. This score is computed as the average of all sentence positions in their original documents belonging to the cluster. The final summary is generated by selecting extracted sentences from smallest to largest sentence position's score.

Scenario 2: K-means with sentence positions

Generating a summary using scenario 1 has some limitations because the relative position of the sentence does not reflect exactly that sentence position in the document. Therefore, we carry out another experiment using sentence positions in the document instead.

Scenario 3: K-means with MMR and sentence position.

Choosing the number of clusters equal to the number of desired sentences in the summary may reduce the summary quality when the number of clusters is small. When the number of clusters is greater than the number of desired sentences, the system can have more options to choose from. Here we have a new task, which is to determine which sentences will be chosen among sentences representing each cluster. To deal with that, we eliminate the most redundant sentences comparing to the sentences that are already included in the summary, using the MMR measure. Sentence positions will be used after that to put sentences in the summary in the right order.

Scenario 4: K-means with the centroid-based method, MMR and sentence positions.

An important factor in the k-means algorithm is to find an optimal number of clusters (k value). If k is large, some clusters may contain poor information. In that case, the sentence representing that cluster should not include in the summary. To deal with that problem, after selecting sentences that represent for all clusters, we apply the centroid-based method to get the most important sentences among the output sentences of k-means. Then the MMR is used to remove redundancy sentences from the output of the centroid-based module. Finally, information about sentence positions is used to put selected sentences in correct order in the summary. Our summarization model following scenario 4 is shown in Figure 1.

The next section introduces in detail our experimental results for each of the above-mentioned scenarios.

6 Experimental Results

6.1 Dataset

The DUC 2007 dataset [4] is used for our multi-document summarization task. The dataset consists of 45 topics with 25 related documents per topic. Each document cluster is processed by 4 different assessors to create a summary of approximately 250 words. These summaries are used to evaluate our models.

Length's distributions of sentences in the DUC 2007 dataset are represented in Figure 2 below.

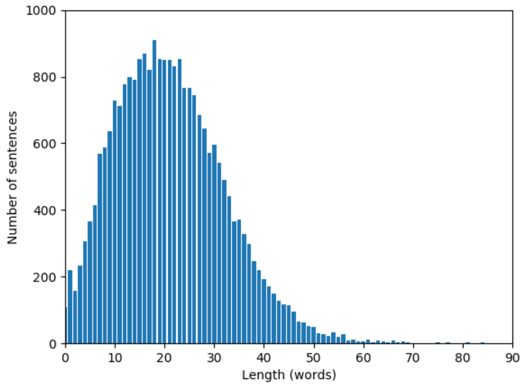

Figure 2. The length's distribution of sentences in the DUC2007 dataset

As it is shown in Figure 2, most of the sentences contain around 20 words. Therefore, a summary of 250 words is equivalent to 12 or 13 sentences.

The documents on the DUC 2007 dataset are preprocessed for extracting its content, eliminating special characters, stemming, and eliminating stop words.

The standard ROUGE measures [23] including Rouge-1, Rouge-2, and Rouge-L are used to evaluate our system's quality.

6.2 Parameter assignments

We first carried out experiments with four scenarios mentioned in Section 4. There are five parameters in these scenarios, including:

- n_clusters: the number of clusters in the k-means algorithm
- n_{dim}: the dimension of a word embedding vector
- λ: the parameter used in the MMR measure
- θ_{sent}: the threshold for the similarity between a sentence and with the centroid vector. This threshold is used in the centroid-based method.
- θ_{sim}: the threshold for the similarity between two sentences. This threshold is used in the centroid-based method.

An important factor in the clustering algorithm is to determine the number of clusters. In scenarios 1 and 2, the number of clusters is supposed to be equal to the number of sentences in the summary. Since the summary's length in the DUC 2007 dataset is approximately 250 words, which is equivalent to 12 to 13 sentences, the number of clusters is assigned with the value 13. In scenario 3, we choose a larger value for the number of clusters, as centroid sentences of some clusters will be eliminated by the MMR measure. In scenario 4, this value is chosen as 50 since we want to have more sentences to be chosen and we have several methods to pick the best sentences to add in the summary.

Parameters used in our scenarios are assigned as follow:

Scenario 1: $n\_clusters = 16$, $n_{\dim} = 256$

Scenario 2: $n\_clusters = 16$, $n_{\dim} = 256$

Scenario 3: $n\_clusters = 21$, $n_{\dim} = 256$, $\lambda = 0.6$

Scenario 4: $n\_clusters = 50$, $n_{\dim} = 256$, $\lambda = 0.6$, $\theta_{sent} = 0.3$, $\theta_{sim} = 0.95$

6.3 Experimental results

Our experimental results with the above four scenarios are shown in Table 1 below.

Table 1: Experimental results with four scenarios

Scenario	Rouge-1	Rouge-2	Rouge-L
K-means + relative position	37.81	7.30	34.61
K-means + position	38.11	7.87	34.86
K-means + MMR + position	38.82	8.15	35.53
K-means + centroid-based method + MMR + position	**40.39**	**9.53**	**37.05**

Table 1 shows that using sentence positions (scenario 2) is better than using the relative sentence position (scenario 1). In addition, it is important to reduce information redundancy from the output summary. The MMR measure is a good solution for that purpose. The Rouge-1 score of the system increases by 0.71% when applying this method.

The results in Table 1 proved that methods of processing information overlap and eliminating sentences represented for clusters containing poor information (scenario 4) are efficient in improving the summary's quality. The Rouge-1 measure in the experiment with scenario 4 is 2.59% higher than that in the experiment with scenario 1. The Rouge-2 and Rouge-L measures of this model are also better than the other models. Because of that, this model is chosen to compare with other multi-document summarization systems.

To evaluate the effectiveness of k-means comparing to other clustering techniques, we implement Latent Semantic Analysis (LSA) [25], Latent Dirichlet Allocation (LDA) [17] for the summarization task. We also compare our proposed approach with LexRank and Centroid-based methods. These two methods are good at ranking sentences from a sentence set, thus they are suitable for text summarization. Table 2 presents our experimental results using the DUC 2007 dataset.

Table 2: Experimental results with basic methods

Method	**Rouge-1**	**Rouge-2**	**Rouge-L**
LSA	37. 92	7.74	35.02
LDA	35.69	6.26	32.71
K-means + position	38.11	7.87	34.86
LexRank	37.52	8.14	34.18
Centroid-based method	38.95	9.08	35.50
LSA + Centroid-based + MMR + Position	36.369	6.895	33.503
LDA + Centroid-based + MMR + Position	36.727	7.224	33.578
K-means + Centroid-based method + MMR + Position (Our model)	**40.385**	**9.532**	**37.051**

Table 2 indicates that the LSA and LDA techniques are not as good as k-means in this summarization task. LexRank is also worse than k-means. However, the centroid-based method is quite good at solving this problem with 38.95% Rouge-1 score. A combination of K-means, centroid-based method, MMR, and sentence positions provides the best result compared to other methods.

We also compare our system with other state-of-the-art research in this field. DSDR-non [26] represents for Document Summarization based on Data Reconstruction with the nonnegative reconstruction. In this method, important sentences are selected and reconstructed by learning a reconstruction function for the sentence. Then DSDR finds an optimal set of representative sentences to approximate the entire set of documents by minimizing the reconstruction error. PV-DM [12] uses distributed memory to represent documents and selects sentences using a document level reconstruction framework.

Two baseline models, Random [26] and Lead [26] are also used to compare with our system. The Random technique selects randomly sentences from input documents to put in the summary. Instead, Lead sorts input documents chronologically and selects the leading sentences from each document one by one. The comparison results are presented in Table 3 below.

Table 3: Comparison with other researches

Method	**Rouge-1**	**Rouge-2**	**Rouge-L**
Random [26]	32.028	5.432	29.127
Lead [26]	31.446	6.151	26.575
DSDR-non [26]	39.573	7.439	35.335
PV-DM [12]	39.826	8.514	
K-means + Centroid-based method + MMR + Position (Our model)	**40.385**	**9.532**	**37.051**

Table 3 shows that our proposed approach provides better results than several modern methods that have been published. It is proved that our proposed method is efficient for the extractive multi-document summarization problem.

An example of our system's output is presented in Table 4 below. As can be seen from Table 4, the system's output shares major points with the reference summary. In other words, it contains main information of the input documents. However, sentences' order should be considered to improve the coherence of the output text.

Table 4 – Summaries of the cluster D0716D in DUC 2007 dataset from our system and the reference

Our system's summary
The Australian federal government Thursday rejected a UNESCO report which called for Kakadu National Park in northwest Australia to be placed on the endangered list because of the threat posed by the Jabiluka uranium mine. CANBERRA, Australia (AP)A United Nations World Heritage committee called Wednesday for the scrapping of the proposed Jabiluka uranium mine in Australia's Northern Territory. The Australian: -- The Australian government's environmental report on the Jabiluka uranium mine (located in Kakadu Natural Park), to be released Thursday, found the area is not under threat and attacked a UNESCO report that said Kakadu Natural Park was in danger. In a major embarrassment to the Howard government, the Bureau of the U.N. World Heritage Committee found Kakadu was under threat, raising the prospect that the committee will this week make Kakadu only the 26th of the world's 552 World Heritage Sites to be placed on its endangered list. The Age -- Australian conservationists and traditional aboriginal owners threatened to blockade development of the huge Jabiluka uranium mine in the country's vast Kakadu National Park, which is on the World Heritage List, after the federal government approved the mining plan for the Jabilika mine yesterday.

"The mission has concluded that Kakadu National Park is exposed to a number of serious threats which are placing it under both ascertained and potential danger," the bureau said in a report after it sent a mission to Australia to examine claims by conservation groups that Kakadu (National Park in Northern Territory) was under threat from Jabiluka.
Reference summary
In October 1997, the Australian government gave permission to Energy Resources of Australia (ERA) to open the Jabiluka uranium mine on the edge of the Kakadu National Park which is on the World Heritage List, in Australia's Northern Territory. The mine is expected to produce 19.5 million tons of ore and generate 4.46 billion U.S. dollars to Australia's GNP over 28 years. Jabiluka is considered a litmus test for up to 12 other uranium mines in Australia. Conservationists and the Aboriginal "Mirrar" owners of the land oppose the mine while ERA insists that its environmental record has been proven by the 16-year operation of the Ranger mine, also located in the Kakadu Park. Opposition leader Kim Beazley said the Labor Party would stop Jabiluka if it won the government in the October national election. Shortly after construction began in mid June 1998, there were a series of public protests. An ERA office in Darwin was firebombed. A team from the United Nations World Heritage Bureau visited the site, then called for closing the Jabiluka mine because it poses a danger to the cultural and natural values of the Kakadu Park. In November 1998, the U.N. World Heritage Bureau, after intense lobbying by the Australian government, decided not to put the Kakadu National Park on its endangered list, but asked for a detailed report by April 15th 1999 on what has been done to prevent further damage and mitigate all threats to the Kakadu park by the Jabiluka mine.

7 Conclusions

In this paper, we have proposed an efficient extractive multi-document system using the K-means clustering algorithm combining with the centroid-based method, MMR, and sentence positions. Experiments with DUC 2007 dataset show that our system is better than several modern systems in this field. It proves that our method of combining several techniques is efficient in the multi-document summarization problem. In the future, we will investigate other techniques to learn vectors representing sentences to improve further the system performance. A deep learning model [2] is a good choice for this goal.

REFERENCES

[1] Abhijit Mondal. *Understanding Word Vectors and Word2Vec*. URL: http://www.stokastik.in/understanding-word-vectors-and-word2vec/. (Last updated: 02 May 2019).

[2] Christopher O. 2015. Understanding LSTM Networks. Retrieved from http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[3] Dragomir R. Radev; Hongyan Jing; Malgorzata Stys; and Daniel Tam. 2004. Centroid-based summarization of multiple documents. In: *Inf. Process. Manage.*, 40(6):919–938

[4] DUC 2007: Task, Documents, and Measures. url: https://duc.nist.gov/duc2007/tasks.html. (Last updated: 02 May 2019).

[5] E. Hovy and C.-Y. Lin. 1996. Automated text summarization and the SUMMARIST system. In *Proc. of a workshop on held at Baltimore*, Maryland, pages 197–214, Baltimore, Maryland.

[6] Gaetano Rossiello, Pierpaolo Basile, Giovanni Semeraro. 2017. Centroid-based Text Summarization through Compositionality of Word Embeddings. *Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres*.

[7] G. Erken and D. R. Radev. 2004. LexPageRank: Prestige in multi-document text summarization. In *Proc. of EMNLP'04*, Barcelona, Spain.

[8] George A. Miller. 1995. *Wordnet: a lexical database for English*. In: *Communications of the ACM* 38(11):39–41.

[9] Gunes Erkan and Dragomir R. Radev. 2004. LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. In *Journal of Artificial Intelligence Research* 22 (2004) 457-479.

[10] Harshal J. Jain, M. S. Bewoor and S. H. Patil. 2012. Context Sensitive Text Summarization Using K Means Clustering Algorithm. In *International Journal of Soft Computing and Engineering.*

[11] Jaime Carbonell and Jade Goldstein. 1998. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. In *Research and Development in Information Retrieval.*

[12] Kaustubh Mani, Ishan Verma, Hardik Meisheri, and Lipika Dey. 2018. Multi-Document Summarization using Distributed Bag-of-Words Model. In: *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI).*

[13] Meishan Hu; Aixin Sun; and Ee-Peng Lim. 2008. Comments-oriented document summarization: understanding documents with readers' feedback. In *Proc. of the 31st ACM SIGIR*, 291–298. ACM.

[14] M R Prathima, H R Divakar. 2018. Automatic Extractive Text Summarization Using K-Means Clustering. In *International Journal of Computer Sciences and Engineering.*

[15] Quoc Le; and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In: In *Proceedings of the 31st International Conference on Machine Learning (ICML-14),* pages 1188–1196.

[16] Q. Zhou, L. Sun, and J.-Y. Nie. 2005. IS SUM: A multi-document summarizer based on document index graphic and lexical chains. In *Proceeding of DUC2005.*

[17] Rachit Arora; and Balaraman Ravindran. 2008. Latent Dirichlet Allocation Based MultiDocument Summarization. In *Conference on Information and Knowledge Management.*

[18] R. Barzilay and M. Elhadad. 1997. Using lexical chains for text summarization. In *Proc. of the Intelligent Scalable Text Summarization Workshop,* Madrid, Spain, 1997.

[19] R. Mihalcea and P. Tarau. 2004. TextRank: Bringing order into text. In *Proc. of EMNLP'04*, pages 404–411, Barcelona, Spain.

[20] Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual Web search engine. In: *Computer Networks and ISDN systems* 30(1-7):107–117. 1998.

[21] Phạm Hoàng Anh. 2019. *Xây dựng chương trình tóm tắt văn bản (tiếng Việt) đơn giản với Machine Learning*. URL: https://viblo.asia/p/xay-dung-chuong-trinh-tom-tat-van-ban-tieng-viet-don-gian-voi-machine-learning-YWOZrgAwlQ0. (Last updated 09/09/2019).

[22] Radev, Dragomir R. and Jing, Hongyan and Budzikowska, Malgorzata. 2000. Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In *NAACL-ANLP 2000 Workshop: Automatic Summarization.*

[23] *What Is ROUGE And How It Works For Evaluation Of Summarization Tasks?* url: https://rxnlp.com/how-rouge-works-for-evaluation-of-summarizationtasks/#.XOO5Z8j7TIW. (Last updated: 02 May 2019).

[24] X. Wan and J. Yang. 2007. CollabSum: exploiting multiple document clustering for collaborative single document summarizations. In *Proc. of SIGIR'07*, pages 143–150, Amsterdam, The Netherlands.

[25] Yihong Gong; and Xin Liu. 2001. Generic text summarization using relevance measure and latent semantic analysis. In *Research and Development in Information Retrieval..*

[26] Zhanying He; Chun Chen; Jiajun Bu; Can Wang; and Lijun Zhang. 2012. Document summarization based on data reconstruction. *In Proceedings of AAAI.*