Journal of information and communication convergence engineering
http://jicce.org

J. Inf. Commun. Converg. Eng. 20(4): 309-316, Dec. 2022 **Regular paper**

Text Summarization on Large-scale Vietnamese Datasets

Ti-Hon Nguyen[1] and Thanh-Nghi Do[1]*, *Member*, *KIICE*

[1]Department of Computer Networks, Can Tho University, Can Tho 94000, Viet Nam

Abstract

This investigation is aimed at automatic text summarization on large-scale Vietnamese datasets. Vietnamese articles were collected from newspaper websites and plain text was extracted to build the dataset, that included 1,101,101 documents. Next, a new single-document extractive text summarization model was proposed to evaluate this dataset. In this summary model, the *k*-means algorithm is used to cluster the sentences of the input document using different text representations, such as BoW (bag-of-words), TF-IDF (term frequency – inverse document frequency), Word2Vec (Word-to-vector), Glove, and FastText. The summary algorithm then uses the trained *k*-means model to rank the candidate sentences and create a summary with the highest-ranked sentences. The empirical results of the F1-score achieved 51.91% ROUGE-1, 18.77% ROUGE-2 and 29.72% ROUGE-L, compared to 52.33% ROUGE-1, 16.17% ROUGE-2, and 33.09% ROUGE-L performed using a competitive abstractive model. The advantage of the proposed model is that it can perform well with $O(n,k,p) = O(n^{(k+2/p)}) + O(nlog_2n) + O(np) + O(nk^2) + O(k)$ time complexity.

Index Terms: Extractive text summarization, Abstractive text summarization, Cluster-based, Sequence-to-Sequence

I. INTRODUCTION

Automatic text summarization is an exciting research field in computer science. There are various publications on both extractive and abstractive methods [1]. In addition, there are some large-scale datasets for evaluating summary models in English, such as GigaWord [2,3] and CNN/Daily Mail [4]. However, there is little summarization research on large-scale Vietnamese datasets [5,6,7] for a single document.

Therefore, this study focuses on the automatic text summarization of a single document for the Vietnamese dataset. The three main contributions of this work are as follows. First, we introduce a new Vietnamese large-scale dataset for automated text summarization research. Second, we propose a new robust and straightforward extractive text-summarization algorithm with $O(n,k,p) = O(n^{(k+2/p)}) + O(nlog_2n) + O(np) + O(nk^2) + O(k)$ time complexity. Moreover, the output summary of our model is a grammatical document owing to the extractive technique. The third is we trained three word-embedding models: Word-to-Vector, Glove, and FastText, which can be used for other word-representation tasks.

The numerical test results showed that our extractive model performed better than the state-of-the-art abstractive summary model, not only in ROUGE-2, but also in training time, summarizing time, and the minimum required resources for experimentation.

The remainder of this paper is organized as follows: Section II describes the work related to our extractive summary model. The methodology is presented in Section III. Section IV discusses the experiment and the results of the summary models. Finally, the conclusions and future work are presented in Section V.

Received 23 April 2022, **Revised** 15 November 2022, **Accepted** 16 November 2022
***Corresponding Author** Thanh-Nghi Do (E-mail: dtnghi@ctu.edu.vn, Tel: +84-38-399-0932)
Department of Computer Networks, Can Tho University, Can Tho 94000, Viet Nam

Open Access **https://doi.org/10.56977/jicce.2022.20.4.309** print ISSN: 2234-8255 online ISSN: 2234-8883

II. RELATED WORK

Text summarization was first introduced by Luhn [8], the task of creating a smaller version of a document or a set of documents while retaining the primary information. Two summarization methods are based on the output technique [1]: extractive and abstract. Abstractive methods attempt to rewrite a summary with a new vocabulary and sentence structure. In contrast, extractive techniques attempt to find the highest-ranking sentences in the original text to produce a summary.

However, based on the number of input documents used to produce a summary, there are two types of text summarization models: single-document and multi-document summary [1]. The single-document model uses one document as the input and generates a summary of the document. The multi-document summary model uses a set of documents as the input and produces a summary of the contents of these documents.

The proposed summarization model is extractive for summarizing a single document. This approach attempts to find sentences that are most similar to the main content of the input document in order to create a summary.

Our extractive summary model is similar to the extractive model proposed by Radev et al. 2004 [9] in using centroids to extract summary sentences. However, the Radev model and our model differ in terms of constructing the centroids and the number of input documents. Thus, Radev used the TF-IDF (**t**erm **f**requency – **i**nverse **d**ocument **f**requency) score of words in the corpus to determine the centroids, but we used a clustering algorithm. The next difference is that Radev proposed a model for summarizing multiple documents, whereas we proposed a model for summarizing a single document.

Our model is similar to Rossiello's model [10] in that it uses word embedding to represent the input text as the vector feature and extracts the highest-ranking sentence for the summary based on the centroids. However, our model differs from Rossiello's in creating centroids and the number of input documents. At that point, Rossiello's model is similar to Radev's model because both are multi-document summarization models that use TF-IDF to assign centroids.

III. METHODOLOGY

For comparison, we evaluated our dataset on our extractive model and pointer generator network, which is a state-of-the-art abstractive text summarization model.

A. Extractive summarization

1) Summary Model

A summary of the tasks is shown in Fig. 1. In the extractive model, the training set was used to train the word embedding. In the abstractive model, the training set was used to train the summary model. Finally, the testing set was used to evaluate both summary models.

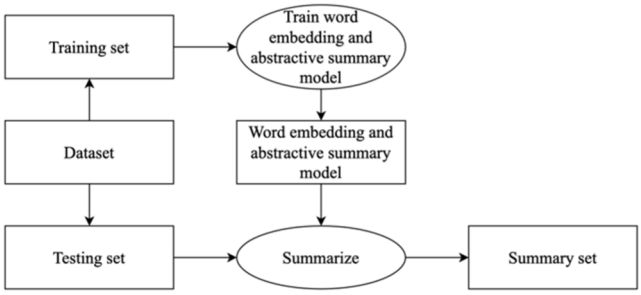

Fig. 1. The process of the summary models.

In our extractive model, we use a clustering model to specify the clusters of sentences in the input document, and then pick the nearest sentence with each cluster's centroid to produce a summary. We used five types of word representations as the input models to find the best representation for our dataset. These are the (bag-of-words) BoW, TF-IDF, Word2Vec, Glove, and FastText models. The extractive summary, Algorithm 1, processes a single-document to produce the summary.

Algorithm 1: Summary function for a single-document

```
Given: document d, number of output sentences k
Initialize S is an empty set of sentences, X is an empty set of vectors,
V is an empty set of vectors, summary is a zero-length text
S ← s sentences is split from d
For s ∈ S do
    X ← x vector of s
end for
kmeans_model ← kmeans(X, k)        // Clustering X into k clusters
                                      by k-means algorithm
C ← c cluster centroid in kmeans_model
For c ∈ C do
x ← f(c, X)                        // f(c, X) is the function for finding closest
                                      x ∈ X with c
V ← (x_idx, x_dis)                 // x_idx is index of x ∈ X, x_dis is distance
                                      of c and x
end for
sort V by increasing x_dis
For x_idx ∈ V do
    summary = summary + S[x_idx] + "."
end for
return summary
```

In Algorithm 1, $|C| = k$, and vectors x of s depend on the type of input model: BoW, TF-IDF, Word2Vec, Glove, and

FastText. When using the BoW and TF-IDF input models, the vocabulary included all words in document *d*. However, the vocabulary was based on the training set when using Word2Vec, Glove, and FastText input models. The *f(c, X)* function is implemented using cosine similarity.

Given the number of sentences in the input text *n*, number of clustering centers *k*, and dimension of the feature vector *p*, the time complexity of Algorithm 1 is $O(n,k,p) = O(n^{(k + 2/p)}) + O(nlog_2n) + O(np) + O(nk^2) + O(k)$. where the time complexity of *k*-means clustering is $n^{(k + 2/p)}$ [11] and the time complexity of Timsort [12] is $nlog_2n$. The step calculation of the time complexity for each line in Algorithm 1 is

$Line\ 1 = C_1 * 1$
$Line\ 2 = C_2 * 1$
$Line\ 3 = C_3 * 1$
$Line\ 4, 5, 6 = C_4 * n * p$
$Line\ 7 = C_5 * n^{(k + 2/p)}$
$Line\ 8 = C_6 * 1$
$Line\ 9, 10, 11, 12 = k * (C_7 * k * n + C_8 * 1)$
$Line\ 13 = C_9 * nlog_2n$
$Line\ 14, 15, 16 = C_{10} * k$
$Line\ 17 = C_{11} * 1$
$Total\ run\ time = C_1 * 1 + C_2 * 1 + C_3 * 1 + C_4 * n * p + C_5 * n^{(k + 2/p)} + C_6 * 1 + k * (C_7 * k * n + C_8 * 1) + C_9 * nlog_2n + C_{10} * k + C_{11} * 1$
$= C + C + C + C * np + C * n^{(k + 2/p)} + C + C * n * k^2 + C * k + C * nlog_2n + C * k + C$
$= 5C + C * np + C * n^{(k + 2/p)} + C * n * k^2 + 2C * k + C * nlog_2n$
$= O(n^{(k + 2/p)}) + O(nlog_2n) + O(np) + O(n * k^2) + O(k) + 0$
$= O(n^{(k + 2/p)}) + O(nlog_2n) + O(np) + O(nk^2) + O(k)$

In the evaluated dataset of this study, the average value of *n* is 21.27 (Sents/Art value of test set in Table 1), and the maximum value of *k* is 5. The value of *p* is 100 when using Word-to-Vector and FastText, and *p* is 300 when using Glove; the average value of p is 24.15 (Words/Sent value of test set in Table 1) when using BoW and TF-IDF.

2) The k-means Model

k-means [13,14] is one of the most useful algorithms in clustering. In this work, we use mini-batch *k*-means [15] (Algorithm 2), a variant of *k*-means that improves performance when working on large-scale datasets to find clusters of sentences in the input document. These clusters were then used to infer a summary.

In Algorithm 2, C is the set of cluster centers, $c \in R^m$ is the cluster center, $|C| = k$, X is the collection of vectors x, *f(C, x)* returns the nearest cluster center $c \in C$ to x by using the Euclidean distance.

Algorithm 2: Mini-batch *k*-means [15]

```
Given: k, mini-batch size b, iterations t, data set X
Initialize each c ∈ C with an x selected randomly from X
v ← 0
For i = 1 to t do
    M ← b examples selected randomly from X
    for x ∈ M do
        d[x] ← f(C, x)          // Cache the center nearest to x
    end for
    for x ∈ M do
        c ← d[x]                // Get cached center for x
        v[c] ← v[c] + 1         // Update per-center counts
        η 1/ v[c]               // Get per-center learning rate
        c ← (1 - η) c + ηx      // Take gradient step
    end for
end for
```

3) Input Model

The inputs of the *k*-means model and the *f(c, X)* function in Algorithm 1 are vector types. Therefore, we must use the word representation model to transform the text document into a vector to feed the model.

a) Bag-of-words:

BoW [16] is one of the first methods of text representation based on the study of word probability in language, which was introduced in 1954 by Harris in the study "Distributional Structure". In a traditional BoW, vocabulary is built using all words in the corpus. A document is typically represented by a vector. The vector element is the word frequency in that document, and the vector length is the number of vocabularies. In our study, to extract the sentences of the input document to construct a summary, the vocabulary includes all words in the input document. Each sentence is then represented by a vector BoW.

b) TF-IDF:

In BoW, the importance of a term (word) is based on its frequency. However, some terms, such as rare words, have a low frequency in the document, but that word is essential for expressing the content of that document. Therefore, in 1972, Jones introduced the TF-IDF [17], which uses weight to present the importance of a word of a record in the corpus to solve this problem. In our study, we treat the sentence as a document and the document as a corpus to calculate the TF-IDF value for all words in the input document.

c) Word-to-vector:

Instead of presenting the term frequency as an element of a vector, as in the BoW model, or presenting the weight of a term as an element of a vector, as in the TF-IDF model, word embedding is the technique used to learn the relation of a word in the corpus to present a term as a vector with short dimension but high quality for natural language processing tasks. The word-to-vector (Word2Vec) [18] was introduced in 2013 by Mikolov and can be used to learn the similarity between words from massive datasets with billions of terms and millions of words in the vocabulary. As it is built on top of the DistBelief architecture [19], Word2Vec can utilize computing clusters with thousands of machines to train large models. In our study, we used the training set as the corpus for training the Word2Vec model. Then, in the summary process, each sentence in the input document is a vector that is the sum of the word vectors according to the word (term) in that sentence.

d) Glove:

Word2Vec learns the relation between word terms using the local window through the content of the training text; this approach does not utilize the statistics of the corpus. In 2014, Pennington introduced the Glove model [20], which proposes a specific weighted least squares model, trains on global word-word co-occurrence counts, and thus makes efficient use of statistics. In our study, similar to Word2Vec, we train a Glove model with the training set; then, in the summary process, we sum the word vectors of all words in the sentence to construct a sentence vector.

e) FastText:

In the Word2Vec and Glove models, each vocabulary word is presented by a distinct vector without parameter sharing. Therefore, in 2016, Joulin introduced FastText [21], a word representation model that focuses on the morphology of words. In Fast Text, each word is represented as a bag of character n-grams. Each vector representation is associated with a character based on the n-gram model, and a word vector is expressed as the sum of all character representations in that word. In our study, FastText is used in the same way as other word representation methods to improve results in the case of rare words.

B. Abstractive Summarization

The Pointer-Generator network [4] is a state-of-the-art abstractive summarization model. This model is a sequence-to-sequence model [22] consisting of an encoder and decoder. The encoder is a single-layer bidirectional LSTM [23], and the decoder is a single-layer unidirectional LSTM.

Bahdanau's attention [24] was applied to the decoder to calculate the probability distribution over source words. In addition, the author added a pointer network [25] to the decoder, which allows both copying words by pointing and generating words from a fixed vocabulary. In addition, a coverage vector is maintained to solve the problem of repetition, which is a common problem in sequence-to-sequence models.

Therefore, we chose this model as the standard for comparing our extractive summarization model and evaluating our large-scale Vietnamese dataset.

IV. EXPERIMENTATION AND RESULTS

A. Experimentation

We evaluated our dataset with both our summary model and the Pointer-Generator network summary model, and the results are based on ROUGE [26] metrics.

1) Dataset

We name our new dataset the **VNText** dataset. Similar to the method used to build the **CNN/Daily Mail** dataset [4, 27], our dataset was built by collecting articles from information websites in Vietnam. These articles were collected in HTML form and then cleaned by eliminating HTML tags and unrelated information, such as links and advertising. After preprocessing, the VNText dataset contained 1,101,101 plain text articles, and every article included the title, subtitle, and main content. The subtitle of each article is used as a reference summary, and its content is used as the input document for the summary models.

Next, VNText was split into three subsets: the training set with 880,895 records, the validation set with 110,103 records, and the testing set with 110,103 records, the details of which are shown in Table 1. The training set was used for training Word2Vec, Glove, FastText, and the Pointer-generator network model, the validation set was used for tuning parameters, and the testing set was used for evaluation.

Table 1. Dataset information for VNText

Information	Train	Test	Test - ref
Articles	880,895.00	110,103.00	110,103.00
Sentences	18,738,333.00	2,342,296.00	155,573.00
Words	452,686,377.00	56,563,630.00	4,163,723.00
Words/Sent	24.16	24.15	26.76
Words/Art	513.89	513.73	37.82
Sents/Art	21.27	21.27	1.41

In Table 1, *Words/Sent* represents the average number of words per sentence, *Words/Art* represents the average number of words per article, and *Sents/Art* represents the average number of sentences per article. In addition, the *Test – ref*

column contains information on the reference summary in the testing set.

2) Extractive Model Training

Table 2 shows the parameters used in our extractive model and the training word embedding, which represent the vector features in our extractive summary model. In this table, the prefix "w2v" stands for the parameters for training Word2Vec and FastText models; the prefix "glove" parameters are used to train the Glove model.

Table 2. Parameters of our extractive model

Name	Value/Range	Use for/Meaning
k	1, 2, 3, 4, 5	Number of *k*-means clusters, and number of sentences in the output summary
w2v_embedding_len	100	Embedding length
w2v_epoch	5	Number of epochs
w2v_window_size	10	Context windows size
w2v_related	skip_gram	Related word model
glove_embedding_len	300	Embedding length
glove_epoch	15	Number of epochs
glove_window_size	15	Context window size
glove_related	skip_gram	Related word model

In addition, all the processes in our extractive model were run on one computer. Its CPU is an ARM Neoverse-N1 with four cores, 2.8 GHz, one thread per core, 64 GB RAM, and 150.34 MB/sec read and write disk speed.

3) Training Pointer-generator Network Model

Similar to the training strategy of the author of the Pointer-generator networks, we first trained this model on our training set in point-gen mode. We then used the last checkpoint to continue the training in the coverage mode.

However, because the VNText has approximately three times more records than the CNN/Daily Mail, we trained this model 10 epochs in point-gen mode and continued training 2,000 iterations for the coverage mode.

In addition, we trained this model on a V100 GPU using default parameters.

A. Results

1) Summary Length

Table 3 presents the average number of words in the summary produced by the summary model. This result provides a general overview of the length of the output summary.

In this table, *k* is the number of sentences in the summary generated by the extractive model, and we use the value of *k* in {1, 2, 3, 4, 5}, as mentioned in the parameters section. In more detail, we can see that the average number of words per summary when using the TF-IDF input model is less than that of the other input models.

Point-gen is the point-gen mode, and coverage is the coverage mode of the abstractive model. The maximum length of the output of the abstractive summary model was 100 words.

Table 3. Number of words per summary

Input model/Mode	k = 1	k = 2	k = 3	k = 4	k = 5	
BoW	**46.81**	83.75	116.31	146.47	174.98	
TF-IDF	36.88	**58.31**	**81.71**	**101.58**	**124.14**	
Word2Vec	45.95	81.54	115.05	142.24	171.96	
Glove	**47.26**	83.02	116.51	143.59	173.57	
FastText	46.05	81.75	115.21	142.11	171.81	
Point-gen						41.38
Coverage						**41.69**

As shown in Table 3, the average output length of our extractive summary model is longer than that of the abstractive summarization model, except when TF-IDF is used at *k = 1*.

2) ROUGE Score

a) ROUGE score of our extractive model:

Tables 4, 5, and 6 show the F1-score of our extractive summary model based on ROUGE-1, ROUGE-2, and ROUGE-L when using various types of input models.

Table 4. F1-score (%) of extractive model based on ROUGE-1

Input model	k = 1	k = 2	k = 3	k = 4	k = 5
BoW	50.97	47.09	40.97	36.06	32.31
TF-IDF	49.36	**50.53**	**47.44**	**43.77**	**39.64**
Word2Vec	51.61	46.60	40.87	36.72	32.81
Glove	**51.91**	46.23	40.49	36.46	32.59
FastText	51.64	46.50	40.77	36.70	32.82

In ROUGE-1, with Glove, the highest F1-score was **51.91%**, with Word2Vec and FastText, and the F1-score was slightly lower at 51.61% and 51.64% when *k = 1*. The TF-IDF F1-score was only 49.36% when *k = 1* and increased to 50.53% when *k = 2*. If we look back to Table 3, we can see that the average length of the summary at the peak point of the F1-score based on ROUGE-1 is not high. In addition, the F1-score exhibits a downward trend when *k* increases to 3, 4, and 5.

In ROUGE-2 and ROUGE-L, with TF-IDF, the best F1-score is **18.77%** when *k = 3* and remains a highly stable trend when the *k* is changing.

Table 5. F1-score (%) of extractive model based on ROUGE-2

Input model	k = 1	k = 2	k = 3	k = 4	k = 5
BoW	**17.88**	18.45	17.92	17.18	16.45
TF-IDF	17.52	**18.62**	**18.77**	**18.53**	**18.09**
Word2Vec	17.22	17.08	16.66	16.28	15.80
Glove	17.45	16.97	16.47	16.09	15.60
FastText	17.27	17.03	16.56	16.17	15.70

Table 6. F1-score (%) of extractive model based on ROUGE-L

Input model	k = 1	k = 2	k = 3	k = 4	k = 5
BoW	**29.72**	27.68	25.31	23.27	21.62
TF-IDF	29.39	**29.21**	**27.75**	**26.27**	**24.66**
Word2Vec	29.55	27.62	25.31	23.54	21.84
Glove	29.65	27.58	25.27	23.51	21.79
FastText	29.57	27.61	25.31	23.57	21.87

Moreover, in ROUGE-L, with BoW input, the F1-score was the highest at **29.72%** but decreased when k was changed. In ROUGE-2 and ROUGE-L, with Word2Vec, Glove, and FastText, the F1-score was only slightly different for various values of *k*.

b) ROUGE score of Pointer-generator networks model:

Table 7 presents the results of the Pointer-generator networks model, which shows that the coverage mode had the highest F1-score in both ROUGE-1 and ROUGE-L with **52.33** and **33.09**%, respectively, which are higher than the F1-score of our extractive summary model.

However, the F1-score in ROUGE-2 was only 16.17%, which was not higher than that in our extractive model.

Table 7. F1-score (%) of the Pointer-generator networks

Model	ROUGE-1	ROUGE-2	ROUGE-L
Point-gen	49.60	15.36	32.59
Coverage	52.33	16.17	33.09

3) Time

Table 8 shows the time spent in training the Word2Vec, Glove, FastText, and Pointer-generator networks on the training set and the time to evaluate the testing set with both summary models.

With our extractive summary model, the evaluation time was not significantly different when using the difference in the word representation model for presenting the input. It required no more than 1.3 hours to summarize our testing set five times and calculate the ROUGE score of the output summaries. In particular, in the training step, we did not spend time using the BoW and TF-IDF. In addition, Glove requires less training time than Word2Vec and FastText.

With the Pointer-generator network model, the training time was not too high, and the evaluation time was also acceptable in both modes. The evaluation time of this model is the time to summarize our testing set at one time and the time to calculate the ROUGE score.

Table 8. Computing time (in hour)

Input model/Mode	Training	Evaluating
BoW	0.00	1.11
TF-IDF	0.00	1.12
Word2Vec	4.73	1.28
Glove	1.32	1.16
FastText	7.24	1.30
Point-gen	129.63	41.02
Coverage	134.30	35.78

V. CONCLUSION AND FUTURE WORKS

A new and efficient extractive text summarization model was proposed to summarize a single document. The approach uses BoW, TF-IDF, Word2Vec, Glove, and FastText to represent the sentences of the input document, and then the *k*-means algorithm is applied to these sentences to create the clusters for extracting the highest-ranked sentences in the summary. The test results on our Vietnamese large-scale dataset show that our extractive model achieved better performance than the state-of-the-art abstractive text summarization model in ROUGE-2, while saving computational time and resources. Further experimental investigations are recommended to combine the word representation model with the input of our extractive summary model to improve the ROUGE score.

APPENDIX

This appendix provides output summaries of an example from the test set with the ROUGE score. It includes the input text, reference summary, output summary of the extractive model with BoW, TF-IDF, Word2Vec, Glove, and FastText word representation, and output summary of the abstractive model with point-gen mode and coverage mode. It also provides the F1 score of each output summary based on the ROUGE metrics.

Content: *theo tiến sĩ phạm văn bình khoa điện tử viễn thông đại học bách khoa hà nội nhiều bà nội trợ vẫn còn băn khoăn về chất lượng bếp từ bếp hồng ngoại mang thương hiệu việt. tuy nhiên chị em vẫn có thể chọn được chiếc bếp ưng ý chất lượng tốt giá cả phải chăng. bếp từ bếp hồng ngoại ngày càng được sử dụng phổ biến nhờ đặc tính an toàn tiết kiệm. tiến sĩ bình tạm chia bếp từ bếp hồng*

ngoại trên thị trường thành 2 loại. một loại là những thương hiệu nổi tiếng thế giới nhập khẩu nguyên chiếc có chất lượng tốt nhưng giá thành cao 30 40 triệu đồng. loại thứ 2 là bếp của doanh nghiệp việt thương hiệu không mạnh chưa được nhiều người tiêu dùng biết đến nhưng giá cả khá rẻ. nếu thuê các tập đoàn công nghệ cao hàng đầu thế giới sản xuất theo đơn đặt hàng những loại bếp thương hiệu việt thường có chất lượng tốt. thông thường bếp sử dụng mặt kính chịu nhiệt schott ceran của hãng schott đức dày tới 4 mm có khả năng chịu lực chịu nhiệt lên đến 1 000 độ c và chống sốc nhiệt. loại kính này được cấu tạo bởi thủy tinh hữu cơ độ cứng lớn đảm bảo không bị trầy xước trong quá trình sử dụng. khi gặp nhiệt độ cao biến đổi đột ngột kính không bị biến dạng hay vỡ nứt. với độ trong suốt và đồng nhất cao hầu như mọi tia hồng ngoại và điện từ đều đi qua được hiệu suất sử dụng năng lượng của kính là rất lớn tiết kiệm điện và giảm thời gian đun nấu. bà nội trợ nên tìm hiểu các linh kiện quan trọng của bếp. ngoài ra nên chọn bếp có mâm nhiệt bếp hồng ngoại và cuộn từ bếp từ do các hãng cao cấp sản xuất chẳng hạn như ego đức. đây là 2 bộ phận biến điện năng thành nhiệt năng. hiệu suất càng cao càng làm giảm tiêu hao điện giảm thời gian nấu chín thức ăn tiết kiệm thời gian và tiền bạc. tiến sĩ bình nhấn mạnh sự chênh lệch giá bán giữa bếp nội và bếp ngoại là do thương hiệu chứ không phải chất lượng. vì vậy người tiêu dùng thông thái có thể chọn mua chiếc bếp thương hiệu việt có chất lượng tốt mà giá cả lại hợp túi tiền. minh tân chef's là thương hiệu bếp của công ty cổ phần thiết bị gia dụng châu âu với các sản phẩm bếp từ bếp điện sử dụng linh kiện của 3 đối tác schott ceran ego đức và copreci tây ban nha. các sản phẩm bếp cao cấp của chef's có giá bán từ 17 triệu đến 20 triệu đồng mỗi chiếc. bếp được bảo hành 3 năm đổi trả hoặc hoàn tiền trong 7 ngày đầu bảo dưỡng theo định kỳ. từ ngày 1 12 công ty bảo hành điện tử với tem bảo hành được dán trên bếp chef's để chống hàng giả hàng nhái. khách hàng nhắn tin theo cú pháp đến tổng đài để xác thực là hàng chính hãng và kích hoạt bảo hành bảo hiểm trách nhiệm sản phẩm lên đến 1 tỷ đồng. liên hệ số 122 168 đường kim giang đại kim hoàng mai hà nội.

Reference: *mặt kính chịu nhiệt mâm nhiệt cuộn từ là những linh kiện mà bà nội trợ nên chú ý khi chọn mua bếp thương hiệu việt.*

BoW: *tiến sĩ bình nhấn mạnh sự chênh lệch giá bán giữa bếp nội và bếp ngoại là do thương hiệu chứ không phải chất lượng. minh tân chef's là thương hiệu bếp của công ty cổ phần thiết bị gia dụng châu âu với các sản phẩm bếp từ bếp điện sử dụng linh kiện của 3 đối tác schott ceran ego đức và copreci tây ban nha.* **F1-score:** 41.33% ROUGE-1, 17.57% ROUGE-2, 22.67% ROUGE-L.

TF-IDF: *một loại là những thương hiệu nổi tiếng thế giới nhập khẩu nguyên chiếc có chất lượng tốt nhưng giá thành cao 30 40 triệu đồng. minh tân chef's là thương hiệu bếp của công ty cổ phần thiết bị gia dụng châu âu với các sản phẩm bếp từ bếp điện sử dụng linh kiện của 3 đối tác schott ceran ego đức và copreci tây ban nha.* **F1-score:** 39.74% ROUGE-1, 18.79% ROUGE-2, 23.84% ROUGE-L.

Word2Vec: *tiến sĩ bình nhấn mạnh sự chênh lệch giá bán giữa bếp nội và bếp ngoại là do thương hiệu chứ không phải chất lượng. với độ trong suốt và đồng nhất cao hầu như mọi tia hồng ngoại và điện từ đều đi qua được hiệu suất sử dụng năng lượng của kính là rất lớn tiết kiệm điện và giảm thời gian đun nấu.* **F1-score:** 44.90% ROUGE-1, 13.79% ROUGE-2, 21.77% ROUGE-L.

Glove: *với độ trong suốt và đồng nhất cao hầu như mọi tia hồng ngoại và điện từ đều đi qua được hiệu suất sử dụng năng lượng của kính là rất lớn tiết kiệm điện và giảm thời gian đun nấu.* **F1-score:** 48.08% ROUGE-1, 7.84% ROUGE-2, 26.92% ROUGE-L.

FastText: *tiến sĩ bình nhấn mạnh sự chênh lệch giá bán giữa bếp nội và bếp ngoại là do thương hiệu chứ không phải chất lượng. nếu thuê các tập đoàn công nghệ cao hàng đầu thế giới sản xuất theo đơn đặt hàng những loại bếp thương hiệu việt thường có chất lượng tốt.* **F1-score:** 43.60% ROUGE-1, 19.85% ROUGE-2, 30.08% ROUGE-L.

Point-gen: *với tiến sĩ phạm văn bình khoa điện tử bếp hồng ngoại ngày càng băn khoăn về chất lượng bếp từ bếp hồng ngoại trên thị trường thành 2 loại bếp từ bếp hồng ngoại mang thương hiệu việt.* **F1-score:** 49.09% ROUGE-1, 20.37% ROUGE-2, 34.55% ROUGE-L.

Coverage: *tiến sĩ dụng phổ biến thông thường bếp hồng ngoại ngoại ngoại trên thị trường thành 2 loại bếp thương hiệu việt đã sử dụng mặt kính chịu lực chịu nhiệt lên đến 1 000 độ cứng nhiệt.* **F1-score:** 57.14% ROUGE-1, 36.89% ROUGE-2, 28.57% ROUGE-L.

REFERENCES

[1] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, and K. Kochut, "Text summarization techniques: A brief survey," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 8, no. 10, pp. 397-405, 2017. DOI: 10.14569/IJACSA.2017.081052.

[2] D. Graff and C. Cieri, "English gigaword," *Linguistic Data Consortium, Philadelphia*, vol. 4, no. 1, pp. 34, Jan. 2003.

[3] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Lisbon, Portugal, 2015. DOI: 10.18653/v1/d15-1044.

[4] A. See, P. J. Liu, and C. D. Manning, "Get to the point: summarization with pointer-generator networks," *arXiv preprint arXiv:1704.04368*, 2017. DOI: arXiv preprint arXiv:1704.04368.

[5] H. Q. To, K. V. Nguyen, N. L. -T. Nguyen, and A. G. T. Nguyen, "Monolingual vs multilingual BERTology for Vietnamese extractive multi-document summarization," in *Proceedings of the 35th Pacific Asia Conference on Language, Information and Computation*, Shanghai: China, pp. 692-699, 2021.

[6] N. Van-Hau, N. Thanh-Chinh, N. Minh-Tien, and H. Nguyen, "VNDS: A Vietnamese dataset for summarization," in *Proceedings*

of 2019 6th NAFOSTED Conference on Information and Computer Science (NICS), Hanoi, Vietnam, pp. 375-380, 2019. DOI: 10.1109/NICS48868.2019.9023886.

[7] T. A. Nguyen-Hoang, K. Nguyen, and Q. V. Tran, "TSGVi: A graph-based summarization system for Vietnamese documents," *Journal of Ambient Intelligence and Humanized Computing*, vol. 3, no. 4, pp. 305-313, Jun. 2012. DOI: 10.1007/s12652-012-0143-x.

[8] H. P. Luhn, "The automatic creation of literature abstracts," *IBM Journal of research and development*, vol. 2, no. 2, pp. 159-165, Apr. 1958. DOI: 10.1147/rd.22.0159.

[9] D. R. Radev, H. Jing, M. Styś, and D. Tam, "Centroid-based summarization of multiple documents," *Information Processing & Management*, vol. 40, no. 6, pp. 919-938, Nov. 2004. DOI: 10.1016/j.ipm.2003.10.006.

[10] G. Rossiello, P. Basile, and G. Semeraro, "Centroid-based text summarization through compositionality of word embeddings," in *Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres*, Valencia, Spain, pp. 12-21, 2017. DOI: 10.18653/v1/W17-1003.

[11] D. Arthur and S. Vassilvitskii, "How slow is the k-means method?," in *Proceedings of the twenty-second annual symposium on Computational geometry*, Sedona: AZ, USA, pp. 144-153, Jun. 2006. DOI: 10.1145/1137856.1137880.

[12] P. McIlroy, "Optimistic sorting and information theoretic complexity," in *Proceedings of the fourth annual ACM-SIAM symposium on Discrete algorithms*, Austin: TX, USA, pp. 467-474, 1993.

[13] S. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, IEEE, vol. 28, no. 2, pp. 129-137, Mar. 1982. DOI: 10.1109/TIT.1982.1056489.

[14] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Oakland: CA, USA, vol. 1, no. 14, pp. 281-297, 1967.

[15] D. Sculley, "Web-scale k-means clustering", in *Proceedings of the 19th International Conference on World wide Web*, Raleigh: NC, USA, pp. 1177-1178, 2010. DOI: 10.1145/1772690.1772862.

[16] Z. S. Harris, "Distributional structure," *Word*, vol. 10, no. 146-162, pp. 146-162, 1954. DOI: 10.1080/00437956.1954.11659520.

[17] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval," *Journal of Documentation*, vol. 28, no. 1, pp. 11-21, Jan. 1972.

[18] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of the 26th International Conference on Neural Information Processing Systems*, Lake Tahoe: NV, USA, vol. 2, pp. 3111-3119, Dec. 2013.

[19] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, Q. Le, and A. Ng, "Large scale distributed deep networks," in *Proceedings of Advances in Neural Information Processing Systems*, Lake Tahoe: NV, USA, vol. 1, pp. 1223-1231, 2012.

[20] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, Doha, Qatar, pp. 1532-1543, 2014. DOI: 10.3115/v1/D14-1162.

[21] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, Valencia, Spain, vol. 2 , pp. 427-431, 2016.

[22] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of the 27th International Conference on Neural Information Processing Systems*, vol. 2, pp. 3104-3112, 2014.

[23] S. Hochreiter and C. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735-1780, Nov. 1997. DOI: 10.1162/neco.1997.9.8.1735.

[24] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014. DOI: arXiv preprint arXiv:1409.0473.

[25] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks", *Advances in Neural Information Processing Systems*, vol. 28, pp. 1-9, 2015.

[26] C. Y. Lin, "ROUGE: A package for automatic evaluation of summaries," *Text summarization branches out*, Barcelona, Spain, pp. 74-81, 2004.

[27] K. M. Hermann, T. Kočiský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," in *Proceedings of the 28th International Conference on Neural Information Processing Systems*, Cambridge: MA, USA, vol. 1, pp. 1693-1701, 2015.

Ti-Hon Nguyen

was born in Camau in 1988. He received his MSc. degree in Computer Science from Cantho University 2018. He is currently a lecturer at the College of Rural Development, Cantho University, Vietnam. His research interests include text summarization clustering.

Thanh-Nghi Do

was born in Cantho in 1974. He received his PhD. degree in Informatics from the University of Nantes in 2004. He is currently an associate professor at the College of Information Technology, Cantho University, Vietnam. He is also an associate researcher at UMI UMMISCO 209 (IRD/UPMC), Sorbonne University, and the Pierre and Marie Curie University, France. His research interests include data mining with support vector machines, kernel-based methods, decision tree algorithms, ensemble-based learning, and information visualization. He has served on the program committees of international conferences and is a reviewer for journals in his fields of expertise.