ICIC Express Letters
Volume **16**, Number **9**, September **2022**
ICIC International ©2022 ISSN 1881-803X
pp. **923–932**

IMPROVING TEXT FEATURES IN TEXT SUMMARIZATION USING HARRIS HAWKS OPTIMIZATION

Alfonsus Wilson* and Amalia Zahra

Computer Science Department, BINUS Graduate Program – Master of Computer Science
Bina Nusantara University
Jl. K. H. Syahdan No. 9, Kemanggisan, Palmerah, Jakarta 11480, Indonesia
amalia.zahra@binus.edu
*Corresponding author: alfonsus.wilson@binus.ac.id

Received September 2021; accepted December 2021

Abstract. *This paper proposes an extractive text summary model with the aim of summarizing news documents with the ability to present the main information available to shorten the required reading time. This problem has become quite significant, especially in the information era, since the number of texts in documents, especially news documents, is growing quite rapidly. This paper explains the effectiveness of Harris Hawks Optimization (HHO) in finding the optimal weight of the 14 text features presented for text summarization. We used the IndoSum dataset which is the standard text summarization dataset for the case of Indonesian text. The HHO approaches were measured at several compression rates and evaluated using Recall-Oriented Understudy for Gisting Evaluation (ROUGE) measurements. We also evaluated the effectiveness of HHO in supporting the related weighting function. Our experimental results show that the proposed method is not only computationally more efficient compared to the neural model but also highly reliable especially for out-of-domain topics.*
Keywords: Text summarization, Harris Hawks optimization, Feature weighting, Text features, Text processing

1. **Introduction.** With the development of the information era lately, there are many sources of information or news that can be obtained from the Internet. This has led to the emergence of a large number of research focuses in the field of text summarization. By definition, text summarization is a process to automatically create a compressed text version of a given text, while still presenting useful information for the users [1]. This summary must reflect the entire summarized paragraph without losing the information.

In fact, according to the International Data Corporation (IDC), digital data circulating on the Internet every year around the world reached 4.4 zettabytes in 2013, and is predicted to reach 180 zettabytes by the end of 2025. The exceptionally large amount of data may contain less important or less representative data. With a large amount of data circulating in cyberspace, it is necessary to have a machine learning algorithm that can automatically summarize longer text into a shorter version with accurate information delivery.

Of course, to achieve this goal, there are many challenges faced in developing a text summarization algorithm. For example, as humans, to be able to make a summary narrative of information, we need to read the document as a whole to get an understanding both explicitly and implicitly before writing it down into a summary. Since different from humans, computers do not have an implicit and broad understanding, and only rely on the written text in the document, it makes text summarization a challenging and non-trivial study.

DOI: 10.24507/icicel.16.09.923

It should be noted that the text summarization approach consists of two main methods, namely extractive and abstractive. There are many methods that can be done with both extractive and abstractive approaches, one of which is Harris Hawk Optimization (HHO) algorithms that belong to extractive approach. The basic difference for the extractive and abstractive text summarization methods can be found in the summary results generated. In extractive text summarization, the summary results are a collection of sentences that are considered important, while abstractive text summarization attempts to create an abstract that may contain sentences that are not in the document or a paraphrased sentence.

The rest of the paper is summarized as follows. In Section 2, we discuss related work in text summarization, followed by Section 3 and Section 4 where we outlined the proposed text features optimization method using HHO along with the text features equation and the problems representation. The final results are analyzed and discussed in Section 5, with neural and non-neural model as comparison. Finally, conclusions are given in Section 6.

2. Related Work.

Text summarization was first proposed by Luhn in 1958 [2]. Then it was followed by several other approaches such as graphs, feature vectors, cluster-based, and Evolutionary Algorithms (EA) which employed a natural approach. Several studies have followed, including those which used the lexical chain method to obtain a representation of a text [3] and which used the LSA (Latent Semantic Analysis) approach to compile text summaries based on the semantic level [4].

Other studies were also carried out using the evolutionary algorithm as proposed by Silla et al. [5] who used genetic algorithms in attribute selection and supervised learning methods to make summaries of text. In Indonesian case study, Aristoteles et al. [6] used a genetic algorithm approach as a weighting algorithm in feature selection with additional features of semantic relations between sentences.

It is unfortunate that text summarization research, especially in Indonesia as reported by Slamet et al. [7] and Gunawan et al. [8], still did not use Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [9] as the evaluation method even though this method had become the standard metric for text summarization research. A research that employed ROUGE evaluation method can be found in [10] by Massandy and Khodra which could achieve a very high ROUGE-2 value. However, this study only used a dataset consisting of 56 articles and cannot be accessed publicly.

In the vast information era, many optimization approaches use evolutionary algorithms that can find a solution in a very large state space while at the same time overcoming the optimum local problems that are often found in the approach using gradient search as presented by Lee and Geem [11]. The problems found in this gradient approach become the basis for scientists to develop better optimization methods.

The Harris Hawks optimization algorithm proposed by Heidari et al. [12] was used in this study for weighting optimization. The HHO algorithm itself is one of many algorithms inspired by the behavior of the nature where the Harris Hawks hunts its prey in the wild. The HHO algorithm becomes superior among other metaheuristic algorithms after being recognized as a superior exploitation and exploration strategy, thereby enabling HHO to avoid Local Optimum (LO) and early convergence.

3. Proposed HHO Summarizer.

This study consists of three main parts, namely text preprocessing, text features, and optimization algorithm to support the weighting of text features. These components are interconnected and called HHO Summarizer.

3.1. Text preprocessing.

Text preprocessing is a stage to prepare text or documents that will be processed at a later stage [7]. Since this study employed the IndoSum as the dataset, the input for the process was in the form of a document that had been tokenized. However, for real case use, the preprocessing collections used are

- Sentence Fragmenting
  Sentence fragmenting is the first process for dividing a document into sentence segments.
- Case Folding
  In this process, all of the alphabets in the document are converted to lower-case.
- Tokenizing
  This process continues the process in stage 1 to segment each word from each existing sentence.
- Stopword Removal
  This process removes all words contained in the "stopword" list in Indonesian. These words include "di", "pada", "karena", etc.
- Stemming
  This process maps and decompress the form of a word into its basic word form. This study refers to stemming algorithm for the Indonesian language [13].

3.2. **Text features.** In the extractive summarization process, text features were used to assess the sentences and words in a document. For each function defined, the higher the score of a sentence indicates that the sentence is more important and most likely to be part of the resulting summary. Various features used for this paper are as follows.

- f_1: TF/ISF (Term Frequency/Inverse Sentence Frequency): This calculation served to limit the number of terms that appeared frequently but were not beneficial for summaries [14].

$$f(s) = tf \cdot isf \tag{1}$$

$$tf(t, s) = f_{t,s} \tag{2}$$

$$isf_i = \log\left(\frac{N}{1 + sf_i}\right) + 1 \tag{3}$$

  where $f_{t,s}$ is the raw count of a term in a sentence, i.e., the number of times that term t occurs in sentence s, and isf_i is the logarithmically scaled inverse fraction of the sentences that contain the term (obtained by dividing the total number of sentences by the number of sentences containing the term, and then taking the logarithm of that quotient).
- f_2: Sentence Location: The value of features was determined by the sentence's position. When compared to other sentences, the first five sentences in a paragraph usually have the most important meaning; hence, the first to the fifth sentence would get a score based on their position [15].

$$f(s_1) = 1 \tag{4}$$

$$\cdots \tag{5}$$

$$f(s_5) = 4/5 \tag{6}$$

$$f(s_n) = 0 \tag{7}$$

- f_3: Title Similarity: The ratio of the number of words in each sentence that intersects the words in the title.

$$f(s) = \frac{keywords\ in\ s \cap keywords\ in\ title}{keywords\ in\ s \cup keywords\ in\ title} \tag{8}$$

- f_4: Proper Noun: This calculation served to count the number of nouns contained in a sentence.

$$f(s) = \frac{proper\ noun\ in\ s}{length(s)} \tag{9}$$

- f_5: Pronoun: This calculation served to count the number of pronouns contained in a sentence.

$$f(s) = \frac{pronoun\ in\ s}{length(s)} \tag{10}$$

- f_6: Noun: This calculation served to count the number of nouns contained in a sentence.

$$f(s) = \frac{noun\ in\ s}{length(s)} \tag{11}$$

- f_7: Cohesion with all sentences: Cosine similarity was used in this computation to measure how similar it was to other sentences. The similarity value of the sentence was added to the similarity value of all the sentences in the news. The higher the similarity value, the more likely the sentence was to be a frequently used sentence that represents news content.

$$f(s) = \sum_{i=1}^{n} sim(s, s_i) \tag{12}$$

- f_8: Cohesion with centroid: The sentence was compared to the centroid vector produced from the average of all sentence vectors in the document, much like in the previous computation.

$$f(s) = sim(s, s_{centroid}) \tag{13}$$

- f_9: Numerical Value: This calculation served to weigh sentences with numbers. Numbered sentences were usually the main sentences that conveyed a summary.

$$f(s) = \frac{numerical\ data\ in\ s}{length(s)} \tag{14}$$

- f_{10}: Sentence Length: This calculation served to remove very short sentences since usually these sentences were not expected to appear in the summaries, namely author names, advertisements, etc.

$$f(s) = \frac{number\ of\ words\ in\ s}{unique\ word\ in\ the\ document} \tag{15}$$

- f_{11}: TextRank Algorithm: This feature uses the PageRank [16] algorithm to calculate the ranking of nodes (sentence) in a graph. The scores were used for the final calculation of sentence score, similar to what have done for web page ranking.

$$f(s) = textrank(s) \tag{16}$$

- f_{12}: Thematic Word: This feature served to calculate the sentences' value that contain frequently appearing words in the news. We take 15 most frequent words from the document as thematic words.

$$f(s) = \frac{keywords\ in\ s \cap keywords\ in\ thematic}{keywords\ in\ s \cup keywords\ in\ thematic} \tag{17}$$

- f_{13}: Bushy Path: The number of other phrases that were lexically related was calculated using the bushiness value of a node (sentence) [17]. Sentences that offended each other usually described an important theme.

$$f(s) = branch\ connected\ to\ the\ node \tag{18}$$

- f_{14}: Aggregate Similarity: Aggregate similarity values were obtained to calculate the importance of sentences in a paragraph. Aggregate similarity calculated the sum of the weights of each connected node, much like a bushy path.

$$f(s) = \sum weight\ connected\ to\ the\ node \tag{19}$$

From the features that will be used, the calculation of each function will be applied to each sentence which is denoted as s. Notation d used for a document or entire paragraph that represents a news, and n is the number of sentences contained in the news. It should be noted that to prevent the text feature from getting a value of 0, the final result of f_1-f_{14} will be added with a bias value of 0.05.

3.3. **HHO on weighting feature.** The role of HHO in the text summarization is to provide optimal weight for each feature to be used so that each feature has its own importance scale. HHO in weight determination can support several features that may or may not have an impact on the final summary. This is supported by the fitness function as the determinant.

Fitness functions play an important role in any metaheuristic algorithm, including the HHO used in this study. Fitness function can be described as a jury whose job is to assess the level of an individual's readiness to survive in the next generations. In this study, the fitness value of an individual HHO is the average ROUGE score from the documents contained in the training set. Therefore, each individual HHO which carries weight as a solution will go through the process of calculating the ROUGE score for the entire document in the training set which then becomes the individual fitness value. An example of a single Hawks vector representation for text summarization solution can be seen in Table 1.

Table 1. Hawks vector representation

w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}	w_{11}	w_{12}	w_{13}	w_{14}

It can be seen that the value of each cell in Hawks vector represents the weight for each text feature starting from f_1 to f_{14}. The resulting solution will pass SoftMax as an activation function so that every sentence s in document d where j is the feature is determined by the formula:

$$score(s) = \sum_{j}^{num\ of\ feature} softmax(w_j) \times f_j(i) \tag{20}$$

Each individual in the HHO population will bring their own solution and will pass the fitness formula to determine the level of goodness of the solution produced by the Hawk itself. The HHO individual fitness formula can be formulated into

$$fitness(i) = \frac{1}{n} \sum_{d}^{n} ROUGE_{using\ solution\ i}(d) \tag{21}$$

where i is each individual HHO, n is the number of documents in the training set, and d is each document in the training set.

All Hawk vectors in this population will start with random value ranging from -1 to 1, and will go through exploration and exploitation phase of 50 epochs. We used 100 individual population that represents 100 solution that will better each generation or epoch.

4. **HHO Summarizer Architecture.** The system architecture consists of three main phases, namely training phase, testing phase, and evaluation phase. The training phase includes the process of modeling using HHO to obtain optimal feature weight values, followed by the testing phase, and then the successfully formed models were tested with the test data. The structure of HHO Summarizer can be seen in Figure 1.

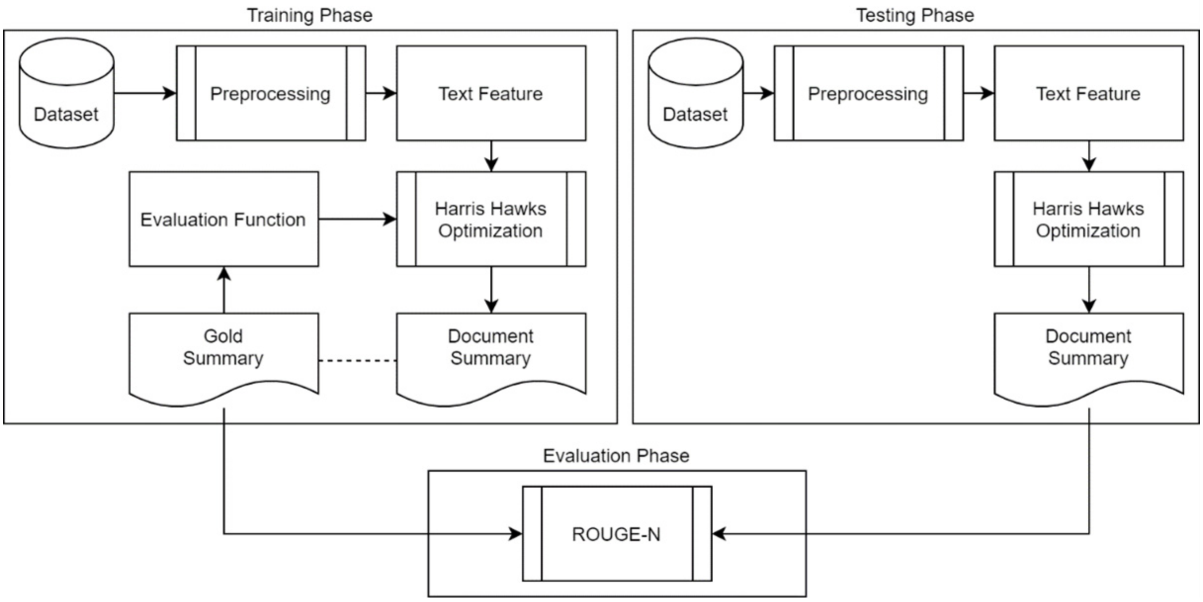

Figure 1. HHO Summarizer architecture

Table 2. IndoSum dataset summary example

Input article	Suara.com - Cerita sekuel terbaru James Bond bocor . Menurut sumber yang terlibat dalam produksi film ini , agen rahasia 007 berhenti menjadi mata-mata Inggris demi menikah dengan perempuan yang dicintainya . " Bond berhenti menjadi agen rahasia karena jatuh cinta dan menikah dengan perempuan yang dicintai , " tutur seorang sumber yang dekat dengan produksi seperti dikutip laman PageSix.com . Dalam film tersebut , Bond diduga menikahi Madeleine Swann yang diperankan oleh Lea Seydoux . Lea diketahui bermain sebagai gadis Bond di sekuel Spectre pada 2015 silam . Jika benar , ini merupakan satu-satunya sekuel yang bercerita pernikahan James Bond sejak 1969 . Sebelumnya , di sekuel On Her Majesty , James Bond menikahi Tracy Draco yang diperankan Diana Rigg . Namun , di film itu Draco terbunuh . Plot sekuel film James Bond ke - 25 bocor tak lama setelah Daniel Craig mengumumkan bakal kembali memerankan tokoh agen 007 . PageSix.com
Expected summary	Cerita sekuel terbaru James Bond bocor . Menurut sumber yang terlibat dalam produksi film ini , agen rahasia 007 berhenti menjadi mata-mata Inggris demi menikah dengan perempuan yang dicintainya . Jika benar , ini merupakan satu-satunya sekuel yang bercerita pernikahan James Bond sejak 1969 . Sebelumnya , di sekuel On Her Majesty , James Bond menikahi Tracy Draco . Namun , di film itu Draco terbunuh .
Source	Suara

This study employed the IndoSum dataset [18]. This dataset is a special dataset for benchmarking the text summarization algorithm in Indonesian language which has at least 20 thousand total news items with each gold summary. The IndoSum dataset was divided into 5 folds where each fold had training, development, and testing data. The distribution of this data allowed the model to be trained on estimated 100 thousand news items.

Describing Figure 1, there are 3 main parts in this research, namely training, testing, and evaluation. In the training process, the values obtained from the feature text will be calculated with random weights, to form a summary that will be compared with the gold summary. The ROUGE-N gold summary value with this summary will be a determinant in the weight calculation process. The HHO model that has been trained will be used in the testing process, which also uses ROUGE-N as a measurement value between documents.

In order to obtain similar results to the dataset, the same tokenizer was used, namely Natural Language Toolkit (NLTK) [19]. Meanwhile, stemming and stopword removal referred to well-known Indonesian stemming research [13]. The importance score of each sentence was obtained from the average score of the text features used. To optimize the importance value of each text feature, we used a solution from HHO that had been trained from the training set. In this case, after finding the weight of each sentence i, the sentences with the highest score were taken proportionally to the compression rate used, which were 10%, 20%, and 30%.

5. **Results and Discussion.** Again, we used the public dataset for text summarization published by IndoSum [18] so we can try with much larger dataset. This dataset has around 20K news articles (from several Indonesian news websites) with six categories formed in 5 folds.

To obtain maximum algorithm evaluation results and comparisons, we used the ROUGE [9] metric, a set of metrics called Recall-Oriented Understudy for Gisting Evaluation that has become standards of summary automatic evaluation. There are several types of ROUGE metrics depending on the text gram used. In order to balance the calculations with the IndoSum's calculations, we used ROUGE-1, ROUGE-2, and ROUGE-L as basic comparisons.

Experiments were carried out first by applying all possible features, each of which had a weight equal to 1. Subsequently, we proceeded with automatic weight calculations using HHO to determine the impact caused by HHO on text summarization. After 100 epochs, the average score with and without HHO (bracketed) was computed over the 5 folds as provided by the IndoSum dataset. This experiment was then repeated with different Compression Rates (CR) to determine the effect of compression rate on the resulting summary. The compression rates tested for comparison were 10%, 20%, and 30% with 2 and 5 acted as lower and upper bound. We also show the HHO Optimizer's best score where we took the maximum F1Score from the summary with 2 to 5 sentences. This score acted as our upper limit while further optimizing the compression rate algorithm. Table 3 shows the F1Score of ROUGE-1, ROUGE-2, and ROUGE-L according to the experimental design as previously described.

Table 3. HHO Summarizer F1Score over three variation compression rates

Scenario	**R-1**	**R-2**	**R-L**
Maximum	*72.87*	*67.57*	*70.66*
30%	**65.69 (57.77)**	**59.16 (49.53)**	**62.79 (54.32)**
20%	64.80 (57.75)	58.19 (49.28)	62.18 (54.04)
10%	59.71 (54.85)	53.55 (45.92)	57.33 (50.98)

Table 3 shows that the use of the HHO algorithm in the case of text summarization shows a significant increase compared to the model that does not use HHO. The most significant increase was shown in the ROUGE-2 value up to 19.5%. The increase in value of the three ROUGE metrics indicates that the HHO algorithm is able to solve the problem of weighting text features in text summarization. In addition to the increase in performance produced by HHO, we see that the value obtained by the HHO Summarizer can still be optimized to reach the Maximum HHO value.

As stated in the IndoSum, we used the ORACLE as a reference for comparison along with the best neural [20] and non-neural model [21]. ORACLE here acted as the upper limit of an extractive text summarization cases on the IndoSum dataset. We used a compression rate of 30% as the best HHO Summarizer model. The F1Score comparison between these three models can be seen in Table 4.

Table 4. HHO Summarizer F1Score among the best model for the IndoSum dataset

Model	R-1	R-2	R-L
ORACLE	79.27	72.52	78.82
Max. HHO Summarizer	*72.87*	*67.57*	*70.66*
HHO Summarizer	65.69	59.16	62.79
Neural model	**67.96**	**61.65**	**67.24**
Non-neural model	62.70	54.32	61.93

Even though the results look promising as the HHO can optimize the weight distribution quite significantly, those results are still considerably lower than the ORACLE. We can see that if we can utilize the compression rate effectively, we can obtain the highest F1Score outperforming the neural model.

Subsequent experiments demonstrate HHO's ability to solve out-of-domain problems. This experiment is needed considering that Indonesian is a low resource language, and collecting comprehensive dataset to summarize many categories can be difficult. Again, we used the F1Score from the best neural model as stated in the IndoSum as our base comparison with the HHO Summarizer. Since the IndoSum dataset has six categories (c1-c6), we trained HHO Summarizer using articles on c1 and evaluated the average performance of ROUGE-1 using articles on c2-c6, applicable for each category c1-c6. In the end, we obtained 36 categories to experiment with. Table 5 shows the out-of-domain HHO Summarizer results.

Table 5. HHO Summarizer F1Score for out-of-domain experiment

Model	Source category					
	Entmt	Inspiration	Sport	Showbiz	Headline	Tech
ORACLE	78.94					
Without HHO	52.62	52.01	60.66	57.32	59.30	52.17
HHO Summarizer	**61.08**	**61.82**	**61.99**	**62.00**	**62.01**	**60.69**
Neural model	60.07	60.74	60.79	60.99	61.68	54.85

Referring to Table 5, this experiment shows the ability of HHO to solve out-of-domain problems. As previously explained, ORACLE has the highest value that can be obtained by the extractive method. From this experiment, the HHO Summarizer shows superior results to the neural model, so it can be concluded that although the results obtained by the HHO Summarizer tend to be slightly lower than the neural model (the results can be seen in Table 4), the HHO Summarizer shows its capabilities if there are limitations on the training data used and available.

6. **Conclusions.** As mentioned earlier, the focus of this study is to investigate the effectiveness of HHO on weighting feature used in text summarization problems to find the optimal weights. We experimented with text summarization with and without HHO and found that HHO could increase the ROUGE score by at least 15%-20%.

According to the test results, we found that to date, the use of a compression rate of 30% with 2 and 5 acts as the lower-bound and upper-bound has the highest ROUGE score between the compression rates of 20% and 10%. We also found that summarizing text using a neural model, although it could have a slightly higher ROUGE score, had difficulty in dealing with few or no training set as written in the out-of-domain experiment.

Future work in this area may focus on increasing the number of features by employing newer model, such as neural summarizer [20] or employing SummaRuNNer [22] as one of the features which hopefully can be boosted by the HHO. Another focus can be directed to employing another supervised algorithm to the compression rate problem. As discussed earlier, the maximum F1Score that can be obtained with the HHO Summarizer is still below the maximum potential of the model itself.

REFERENCES

[1] M. A. Fattah and F. Ren, GA, MR, FFNN, PNN and GMM based models for automatic text summarization, *Computer Speech & Language*, vol.23, no.1, pp.126-144, 2009.
[2] H. P. Luhn, The automatic creation of literature abstracts, *IBM Journal of Research and Development*, vol.2, no.2, pp.159-165, 1958.
[3] R. Barzilay and M. Elhadad, Using lexical chains for text summarization, *Advances in Automatic Text Summarization*, pp.111-121, 1999.
[4] J.-Y. Yeh, H.-R. Ke, W.-P. Yang and I-H. Meng, Text summarization using a trainable summarizer and latent semantic analysis, *Information Processing & Management*, vol.41, no.1, pp.75-95, 2005.
[5] C. N. Silla, G. L. Pappa, A. A. Freitas and C. A. A. Kaestner, Automatic text summarization with genetic algorithm-based attribute selection, *Ibero-American Conference on Artificial Intelligence*, pp.305-314, 2004.
[6] A. Aristoteles, Y. Herdiyeni, A. Ridha and J. Adisantoso, Text feature weighting for summarization of document Bahasa Indonesia using genetic algorithm, *International Journal of Computer Science Issues (IJCSI)*, vol.9, no.1, pp.1-6, 2012.
[7] C. Slamet, A. R. Atmadja, D. S. Maylawati, R. S. Lestari, W. Darmalaksana and M. A. Ramdhani, Automated text summarization for Indonesian article using vector space model, *IOP Conference Series: Materials Science and Engineering*, vol.288, DOI: 10.1088/1757-899X/288/1/012037, 2018.
[8] D. Gunawan, A. Pasaribu, R. F. Rahmat and R. Budiarto, Automatic text summarization for Indonesian language using TextTeaser, *IOP Conference Series: Materials Science and Engineering*, vol.190, DOI: 10.1088/1757-899X/190/1/012048, 2017.
[9] C.-Y. Lin, ROUGE: A package for automatic evaluation of summaries, *Text Summarization Branches Out*, pp.74-81, 2004.
[10] D. T. Massandy and M. L. Khodra, Guided summarization for Indonesian news articles, *2014 International Conference of Advanced Informatics: Concept, Theory and Application (ICAICTA)*, pp.140-145, 2014.
[11] K. S. Lee and Z. W. Geem, A new structural optimization method based on the harmony search algorithm, *Computers & Structures*, vol.82, nos.9-10, pp.781-798, 2004.
[12] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja and H. Chen, Harris Hawks optimization: Algorithm and applications, *Future Generation Computer Systems*, vol.97, pp.849-872, 2019.
[13] F. Z Tala, *A Study of Stemming Effects on Information Retrieval in Bahasa Indonesia*, Master Thesis, Institute for Logic, Language and Computation, Universiteit van Amsterdam, The Netherlands, 2003.
[14] J. L. Neto, A. A. Freitas and C. A. A. Kaestner, Automatic text summarization using a machine learning approach, *Brazilian Symposium on Artificial Intelligence*, pp.205-215, 2002.
[15] Y. K. Meena and D. Gopalani, Evolutionary algorithms for extractive automatic text summarization, *Procedia Computer Science*, vol.48, pp.244-249, 2015.
[16] L. Page, S. Brin, R. Motwani and T. Winograd, *The Pagerank Citation Ranking: Bringing Order to the Web*, Technical Report, Stanford InfoLab, 1999.
[17] M. Mitra, A. Singhal and C. Buckley, Automatic text summarization by paragraph extraction, *Intelligent Scalable Text Summarization*, 1997.

[18] K. Kurniawan and S. Louvan, IndoSum: A new benchmark dataset for Indonesian text summarization, *2018 International Conference on Asian Language Processing (IALP)*, pp.215-220, 2018.

[19] S. Bird, E. Klein and E. Loper, *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*, O'Reilly Media, Inc., 2009.

[20] J. Cheng and M. Lapata, Neural summarization by extracting sentences and words, *arXiv Preprint*, arXiv: 1603.07252, 2016.

[21] C. Aone, M. E. Okurowski and J. Gorlinsky, Trainable, scalable summarization using robust NLP and machine learning, *The 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics*, pp.62-66, 1998.

[22] R. Nallapati, F. Zhai and B. Zhou, SummaRuNNer: A recurrent neural network based sequence model for extractive summarization of documents, *Proc. of the AAAI Conference on Artificial Intelligence*, 2017.