

# Text Summarization on Verdicts of Industrial Relations Disputes Using the Cross-Latent Semantic Analysis and Long Short-Term Memory

*by Galih Wasis Wicaksono*

---

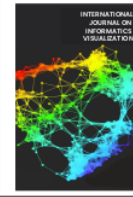
**Submission date:** 25-Sep-2023 11:32AM (UTC+0700)

**Submission ID:** 2176042007

**File name:** zation\_cross\_latent\_semantic\_analysis\_long\_short-term\_memory.pdf (3.48M)

**Word count:** 5120

**Character count:** 25443



## Text Summarization on Verdicts of Industrial Relations Disputes Using the Cross-Latent Semantic Analysis and Long Short-Term Memory

Galih Wasis Wicaksono <sup>a</sup>, Muhammad Nafi<sup>†</sup> Maula Hakim <sup>a,b</sup>, Nur Hayatin <sup>a</sup>, Nur Putri Hidayah <sup>a,\*</sup>,  
Tiara Intana Sari <sup>a</sup>

<sup>a</sup> Informatics, <sup>5</sup> University of Muhammadiyah Malang, Malang, Indonesia

<sup>b</sup> Law, University of Muhammadiyah Malang, Malang, Indonesia

Corresponding author: \*[nurputri@umm.ac.id](mailto:nurputri@umm.ac.id)

**Abstract**—The information presented in the documents regarding industrial relations disputes constitutes four legal disputes. However, too much information leads to difficulty for readers to find essential points highlighted in industrial relations dispute documents. This research aims to summarize automated documents of court decisions over industrial relations disputes with permanent legal force. This research involved 35 documents of court decisions obtained from Indonesia's official Supreme Court website and employed an extractive summarization approach to summarize the documents by utilizing Cross Latent Semantic Analysis (CLSA) and Long Short-Term Memory (LSTM) methods. The two methods are compared to obtain the best results CLSA was employed to analyze the connection between phrases, requiring the ordering of related words before they were converted into a complete summary. Then, the use of LSTM is combined with the Attention module to decoder and encoder the information entered so that it becomes a form that can be understood by the system and provides a variety of splitting of documents to be trained and tested to see the highest performance that the system can generate. The research has found out that the CLSA method gave a precision of 79.1%, recall score of 39.7%, and ROUGE-1 score of 50.9%, and the use of LSTM was able to improve the performance of the CLSA method with the results obtained 93.6%, recall score of 94.5 %, and ROUGE-1 score of 93.9% on the variation of splitting 95% training and 5% testing.

**Keywords**—Extractive summarization; cross latent semantic analysis; long short-term memory; legal document.

Manuscript received 15 Apr. 2022; revised 15 Sep. 2022; accepted 22 Dec. 2022. Date of publication 10 Sep. 2023.  
International Journal on Informatics Visualization is licensed under a Creative Commons Attribution-Share Alike 4.0 International License.



### I. INTRODUCTION

Artificial Intelligence (AI) is a system that can learn from the user's experience. AI can also form a pattern out of data prepared as exercise materials. The pattern allows automatization resembling those made by humans, commonly dubbed human-centered Artificial Intelligence [1], [2]. The implementation of AI has reached other sectors, such as education [3], [4], health [5], and social & business sectors [6]. Departing from this wider scope, AI also plays its role in automated decision-making within a legal purview. In automated decision-making, AI can consider what comes after the decision-making according to training data previously collected and processed [7].

The documents on court decisions were sourced from industrial relations disputes (henceforth referred to as PHI). The documents consist of judicial decisions with abundant

information. The substance of the documents covers four disputes: 1) right disputes, 2) conflict of interest, 3) layoffs, and 4) labor union-related disputes. However, information ranges from significant to insignificant. The problem is making it hard for the readers to spot the essence in the PHI documents. The documents on industrial relations disputes require an analysis recalling that they take the scope between society and corporate.

Text summarization is categorized into two, constituting abstractive and extractive summarization. Abstractive summarization functions to compose new words identical to the words existing in original documents. Abstractive summarization refers to creating new phrases without omitting the substance in the original documents [8]. On the other hand, extractive summarization requires the selection of important words existing in the original documents. This type of summarization cannot produce new phrases but can create

new sentences to be incorporated into a complete summary of the original documents. Extractive summarization is preferred because it is simpler and easier to implement in several cases of automated text summarization [9].

Previous studies reviewed and surveyed text summarization of court decisions. The first study was focused on the summarization of single court decisions sorted out with Latent Semantic Analysis (LSA), Maximum Marginal Relevance (MMR), conditional Random Field (CRF), and Matrix Factorization (NMF), but for multi-documents, the methods such as Dependency Word Pair (DWP), Cosine Similarity and Latent Dirichlet Allocation (LDA) are more recommended [10].

Other studies summarized article documents in Bahasa by employing the MMR method [11], while the use of LSA was proven in article summarization presented in Bahasa [12]. The study using Cross Latent Semantic Analysis (CLSA) in document summarization was presented in Bahasa [13]. The comparison between CLSA and LSA sourced from 240 news documents in Bahasa resulted in an F-Measure score of 70% and CLSA accounting for 72%. The method of CLSA often gives shorter summary results and can extract essential points compared to LSA, but the LSA is more reliable in language use [14].

Several studies using Long Short-Term Memory (LSTM) to summarize documents have been conducted, including either extractive [15], [16] or abstractive summarization [17]–[19], which have proven the performance of LSTM in text summarization. In the text summarization of court decision documents, several methods such as LSA [20], and the merging of several methods such as LSA, LUHN, LEXRANK, and SUMBASIC [21], were employed.

This research required the summarization of single documents in an automated mode on the documents of court decisions by employing CLSA and LSTM methods. The PHI decisions used in this research also held permanent legal force. The summarization performed did not spoil the messages contained in the documents, and it could help readers spot important information contained in the decisions.

## II. MATERIALS AND METHOD

Research stages in the summarization of the documents of PHI employed CLSA and LSTM, requiring data collection, data pre-processing, document segmentation, document ranking, and evaluation.

### A. Collecting Data

The PHI data were sourced from text documents manually downloaded from the official website of the directory of Indonesia's Supreme Court (<https://putusan3.mahkamahagung.go.id/>). This research involved 35 documents of the PHI court decisions with permanent legal force. The data were collected based on keywords/types of cases of industrial relations disputes. The documents obtained were downloaded in Portable Document Format (PDF), and the documents of the PHI decisions were further converted into document.txt.

Each sheet of the document of PHI decision contained a watermark and information on both the header and footer. To improve the data quality, we cleaned and normalized

documents, and the documents were further normalized to erase ASCII from the documents.

### B. Pre-processing

Pre-processing was performed to enhance the quality of testing data for further processing. Moreover, the file format conversion was also performed to adjust the document input to the methods used. The methods used in text pre-processing were examined according to the characteristics of the methods under the type and model of the dataset used and to enhance accuracy [22], [23]. The pre-processing stage employed case folding, tokenizing, and StopWord Removal [24], [25]. The omission of less important words was performed using a modified StopWord list, recalling that not all words were to be omitted. This approach was intended to gain results relevant to the characteristics of PHI documents. The process taken in setting the StopWord list took a consultation with a legal expert.

### C. Document Segmentation

One PHI document has five core parts, constituting decision header (title and decision number), the identity of both plaintiff and defendant, case (lawsuit), judge's consideration, indictment, and decision footer (the time and the names of the panel of judges deciding a case). The segmentation of the PHI decision documents into several decisions could result in independent information separate from the other parts of a decision, and this setting degrades the quality of the summary result.

### D. Cross Latent Semantic Analysis (CLSA)

CLSA is a method aiming to summarize a text through extraction developed based on an LSA method. CLSA was employed to analyze the connection between phrases, requiring the ordering of related words before they were converted into a complete summary [13][14]. The process of summarization with the method of CLSA began with word weighing by using Term-Frequency Method-Inverse Document Frequency (TF-IDF) [12].

$$idf = \log \frac{N}{df(t)} \quad (1)$$

$$w = tf * idf \quad (2)$$

Legend:

- W : Weight of document towards words
- Tf : The number of words searched in one document
- idf : Inverse Document Frequency
- N : The number of documents
- df : The number of documents towards words searched

The matrix process represented the results of word weighing previously performed using TF-IDF. The weight matrix of words was decomposed to reduce the data dimension. This research employed Singular Value Decomposition (SVD) method to perform matrix decomposition resulting from the previous matrix input. SVD reduces the number of dimensions of the matrix of the weight of words from the documents of PHI decisions. The decomposition and the normalization results are presented in equation 3. [13], [26].

$$A = USV^T \quad (3)$$

Legend:

- A : Matrix of word weighing results
- U : Singular vector of matrix A
- V : The diagonal vector from the corresponding singular vector
- S : Diagonal matrix with positive and null matrix

Following the matrix decomposition stage, a word list forming each sentence was obtained, and this narrowed down the number of sentences [27] so that the system was able to perform extraction of sentences of the total weight previously obtained from SVD, as shown in Table 4. Furthermore, the decomposition result would be re-selected according to the weight adjusted to the average score [26].

#### E. Long Short-Term Memory

Long Short-Term Memory (LSTM) can fix the drawbacks spotted in the conventional Recurrent Neural Network (RNN). In equation 4,  $x_t$  is a hidden state of the input of the period of  $-t$ , and  $h_{t-1}$  is a hidden state of the previous period, while  $f$  represents activation function (non-linear) [28], where  $f$  is replaceable by LSTM. The stages of LSTM are presented in Fig. 1.

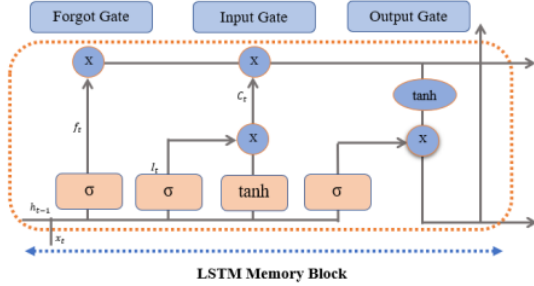


Fig. 1 Detail of LSTM

$$h_t = f(x_t, h_{t-1}, b) \quad (4)$$

Fig. 1 indicates that LSTM has three gates: forget gate, the input gate, and output gate [29]. The computation process in LSTM [30] began by filtering information with forget gate ( $f$ ), where less significant information was omitted, which is represented by the sigmoid function ( $\sigma$ ) as in equation 5.

$$f_t = \sigma(W_f [h_{t-1}, X_t] + b_t) \quad (5)$$

As seen in the equation,  $X_t$  represents a hidden state from the input during the period of  $t$ , and  $h_{t-1}$  represents the hidden state of the previous period. The second stage constitutes the input gate, representing the process where the system was intended to sort out particular information renewed into cell state with the function of tanh as shown in Equations 6 and 7. Furthermore, in the following stage, cell state received information determined in an earlier process, as shown in equation 8.

$$i_t = \sigma(W_f [h_{t-1}, X_t] + b_t) \quad (6)$$

$$C_t = \sigma(W_e [h_{t-1}, X_t] + b_c) \quad (7)$$

$$c_t = f_t \times c_{t-1} + i_t \times C_t \quad (8)$$

Output gate was the final stage, representing a process to give output score in a hidden state and to put the cell state at tanh with the implementation of sigmoid, as shown in Equations 9 and 10. The implementation of LSTM in document summarization is presented in Fig. 2.

$$O_t = \sigma(W_o [h_{t-1}, X_t] + b_o) \quad (9)$$

$$h_t = O_t \times \tanh(C_t) \quad (10)$$

In the modeling process, the training data is entered into LSTM summarization models as a data input that has been created in Fig. 2. First, the input layer is defined according to the vocab size and maximum word length, then followed by an embedding layer. This layer's vocabulary retrieval process is coded with an array of integers and embedding vectors for each word index.

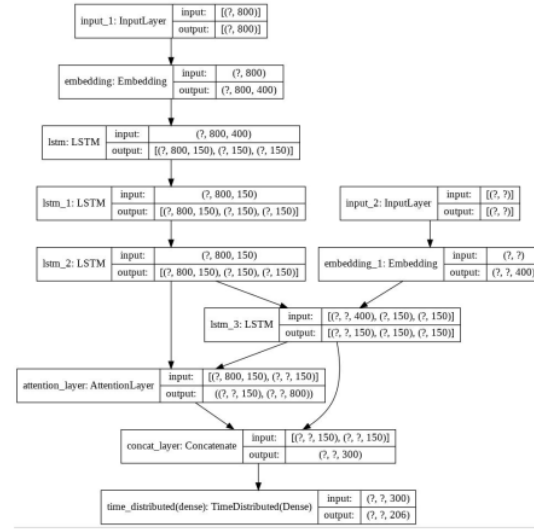


Fig. 2 Structure of LSTM for document summarization

The input layer was given based on the maximum length of 800 words. The embedding layer referred to the score of the maximum length of the original text and the text from an expert with 800 and 400 words, respectively. Furthermore, the implementation of stacked LSTM with several layers of LSTM stacked over one another with the number of neurons based on a latent dimension with the value of 150. The use of layer attention was involved in the model-making to maximize text encoding and decoding processes. Thus, combining the layers of LSTM and Attention requires the definition of combining layers from the input resulting from the two layers. The final stage involved defining the output layer using layer dense with softmax activation.

The compiling model process utilized the optimizer the Rmsprop referring to loss sparse categorical cross-entropy with its function to immediately change integers into one-hot vector. The iteration score/epoch was defined by 500 using callback early stopping with patience ten so that if the loss value does not change in 10 iterations, the lowest value is used.

### F. Testing and Evaluation

This section aims to find out the effectiveness and accuracy of the method used: recall-oriented understudy for gusting evaluation (ROUGE), which tests text summary results [31] by matching the summary results generated by the system and those of the summarization performed by an expert [32]. To find out the accuracy level of the summary generated by the system compared to the summary given by an expert, ROUGE<sub>Precision</sub> was used, while ROUGE<sub>Recall</sub> was used to find out the level of success of the system in regaining the information generated, as presented in equation 12-13:

$$R - N = \frac{\sum_{s \in \{RefSum\}} \sum_{N-g \in SC_{match}(N-g)}}{\sum_{s \in \{RefSum\}} \sum_{N-g \in SC_{(N-g)}}} \quad (11)$$

For:

$$R_{Recall} = \frac{bigram \ value \ sum.system}{sum.expert \ bigram \ numbers} \quad (12)$$

$$R_{Precision} = \frac{unigram \ value \ sum.expert}{sum.system \ unigram \ numbers} \quad (13)$$

### III. RESULTS AND DISCUSSION

The summary results involving 35 documents of PHI court decisions were tested using ROUGE-1 score on each document, as presented in Table 1. For the result, the maximum result of the ROUGE score was taken from the method of LSTM with a maximum score of 100% and the minimum score of 70%. In addition to showing the matching scores of the summary results, this research also tested the summary results using the average precision and recall for the three methods.

These two testing methods were intended to determine the system's success in finding solutions to generate a summary resembling the original summary results. The comparison of the number of words between the original documents and the summary given by an expert and system is presented in Table 4.

TABLE I  
ROUGE SCORE RESULTS FOR EACH METHOD (LSA, CLSA, AND LSTM)

Document	ROUGE SCORE		
	LSA	CLSA	LSTM
Doc_1	0.674	0.621	0.925
Doc_2	0.603	0.560	1.000
Doc_3	0.209	0.120	0.754
Doc_4	0.538	0.548	0.777
Doc_5	0.639	0.618	1.000
Doc_6	0.639	0.618	1.000
Doc_7	0.406	0.395	1.000
Doc_8	0.505	0.502	0.701
Doc_9	0.505	0.502	1.000
Doc_10	0.574	0.542	1.000
Doc_11	0.570	0.519	0.961
Doc_12	0.631	0.663	1.000
Doc_13	0.481	0.435	1.000
Doc_14	0.576	0.616	1.000
Doc_15	0.576	0.616	1.000
Doc_16	0.772	0.715	1.000
Doc_17	0.564	0.557	1.000
Doc_18	0.564	0.557	1.000
Doc_19	0.487	0.471	1.000
Doc_20	0.393	0.414	1.000
Doc_21	0.216	0.192	1.000

Document	ROUGE SCORE		
	LSA	CLSA	LSTM
Doc_22	0.397	0.364	1.000
Doc_23	0.389	0.364	1.000
Doc_24	0.603	0.560	0.995
Doc_25	0.570	0.519	0.930
Doc_26	0.674	0.621	1.000
Doc_27	0.574	0.542	0.925
Doc_28	0.564	0.622	1.000
Doc_29	0.547	0.532	0.754
Doc_30	0.551	0.660	0.777
Doc_31	0.606	0.657	1.000
Doc_32	0.209	0.120	1.000
Doc_33	0.603	0.560	1.000
Doc_34	0.570	0.519	0.701
Doc_35	0.674	0.621	1.000

In the LSTM method, the separation of data into training and test data also affects the results of each document, different proportions in the splitting process also provide variations on each performance obtained so that providing various variations of splitting is very necessary as a way to obtain The best results and total of data for each splitting can be seen in Table 2.

TABLE II  
SPLITTING PROPORTION

	Data Splitting Category		
	80:20	90:10	95:5
Training Data	35	38	41
Testing Data	9	6	3

Each category of data splitting could implement epoch sharing a similar indicating number of 500. However, a good fit condition could be identified with the help of callbacks early topping functioning to retain the most minimum loss validation score. The data splitting mode accounted for 80:20, the iteration accounted for 441, and the lowest loss validation score accounted for 25%. The second data splitting model represented 90:10 with the iteration of 391, and the lowest loss validation score accounted for 23%. Then, the last splitting 95:5 with the iteration of 500 and the lowest loss validation score accounted for 2%. The training process of those models is presented in Fig. 4, 5, and 6.

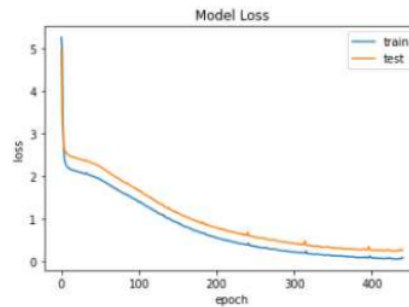


Fig. 1 Graph Model Loss with Splitting 80:20



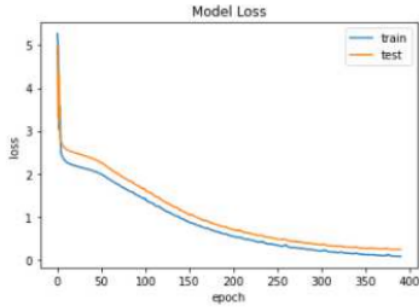


Fig. 2 Graph Model Loss with Splitting 90:10

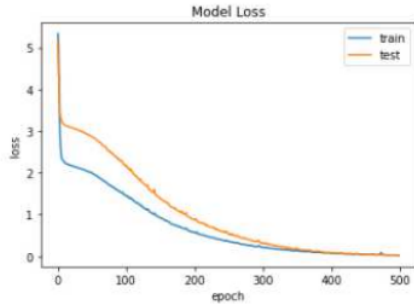


Fig. 3 Graph Model Loss with Splitting 95:5

Each category of data splitting had a different ROUGE-1 score, with its best results presented with a data proportion of 95% of training data in comparison to 5% of testing data as shown in Table 3. This can happen because in the training process the data provided is much more so that it can affect the learning process. The system becomes more stable and performs much better than other splitting proportions.

TABLE III  
ROUGE-1 SCORE FOR EACH SPLITTING ON LSTM

Document	LSTM		
	80:20	90:10	95:5
Doc_1	1.000	0.986	0.925
Doc_2	0.638	0.670	1.000
Doc_3	1.000	0.482	0.754
Doc_4	0.737	0.657	0.777
Doc_5	0.615	1.000	1.000
Doc_6	0.729	1.000	1.000
Doc_7	0.775	1.000	1.000
Doc_8	0.995	0.640	0.701
Doc_9	1.000	0.613	1.000
Doc_10	0.995	0.549	1.000
Doc_11	0.696	0.683	0.961
Doc_12	0.686	0.646	1.000
Doc_13	0.995	1.000	1.000
Doc_14	0.961	0.646	1.000
Doc_15	0.960	0.596	1.000
Doc_16	1.000	0.659	1.000

Document	LSTM		
	80:20	90:10	95:5
Doc_17	0.676	1.000	1.000
Doc_18	1.000	0.577	1.000
Doc_19	1.000	0.650	1.000
Doc_20	0.581	0.659	1.000
Doc_21	1.000	0.679	1.000
Doc_22	1.000	1.000	1.000
Doc_23	0.995	1.000	1.000
Doc_24	0.686	0.986	0.995
Doc_25	1.000	1.000	0.930
Doc_26	1.000	1.000	1.000
Doc_27	0.995	1.000	0.925
Doc_28	0.877	0.506	1.000
Doc_29	1.000	0.659	0.754
Doc_30	0.970	1.000	0.777
Doc_31	1.000	1.000	1.000
Doc_32	1.000	1.000	1.000
Doc_33	0.542	1.000	1.000
Doc_34	0.650	0.995	0.701
Doc_35	1.000	0.634	1.000

From Table 3 know that each of splitting have minimum ROUGE-1 score. But, for each splitting have the same maximum value which is in 100% score. First in 80:20 splitting proportion the minimum value is 58% and the maximum value is 100%. Then, in 90:10 splitting proportion the minimum value is 48% and the maximum value is 100%. Last, in 95:5 splitting proportion the minimum value is 70% and the maximum value is 100%.

Furthermore, Table 4 shows the comparison of words in 35 documents after going through the summary process with various methods, in the table it can be seen that the LSTM has a much lower word count than the CLSA method. So from the results of the comparison of the number of words it can be concluded that LSTM is able to summarize documents optimally.

TABLE VI  
COMPARISON OF THE NUMBER OF WORDS FROM SUMMARY RESULTS

No Doc	Word Number			
	Doc Asli	Expert's Summary	CLSA Summary	LSTM Summary
1	1604	385	701	187
2	1383	315	601	186
3	5392	270	3577	154
4	1783	363	903	158
5	1628	350	783	150
6	1628	350	783	173
7	1653	288	728	186
8	1638	289	619	186
9	1638	289	619	145
10	1556	289	605	186
11	2075	370	913	113

No Doc	Word Number			
	Doc Asli	Expert's Summary	CLSA Summary	LSTM Summary
12	1429	390	648	122
13	1638	268	885	197
14	1957	487	955	145
15	1957	487	955	186
16	1634	513	785	158
17	1366	267	615	122
18	1366	267	615	243
19	1464	270	536	145
20	1853	292	1046	114
21	2141	189	1160	145
22	1656	234	834	243
23	1652	234	841	197
24	1383	315	601	122
25	2074	370	913	158
26	1604	385	701	145
27	1555	289	605	187
28	1537	383	832	216
29	1941	405	1034	154
30	1934	569	984	114
31	1292	416	564	187
32	5392	270	3577	237
33	1383	315	601	121
34	2075	370	913	186
35	1604	385	701	158

All testing results performed in this research were summarized in Table 5, giving summary results with better LSTM method than LSA and CLSA results. With the data splitting proportion of 95:5, LSTM was capable of generating the ROUGE score of 0.939, and Recall and Precision accounted for 0.936 and 0.949, respectively.

TABLE V  
AVERAGE RESULTS OF PRECISION, RECALL AND ROUGE SCORE

Method	Evaluation Score		
	Precision	Recall	ROUGE-1
LSA	0.459	0.463	0.411
CLSA	0.791	0.397	0.509
LSTM (80:20)	0.865	0.907	0.879
LSTM (90:10)	0.813	0.807	0.805
LSTM (95:05)	0.936	0.949	0.939

Departing from testing result analysis, this research has found there were several contributing factors resulting in the varied scores of the evaluation of the summarization of the documents on PHI court decisions with LSA and CLSA methods. The variety of words in the sentences extracted affected the quality of the summarization results. Numerous numbers and symbols in PHI court decisions have hampered the system of extracting the connection between phrases, affecting the relevance of the summary results given by the expert during the evaluation process. Numerous words in the documents of the decisions were also present as one of

contributing factors causing the decrease in the evaluation score generated because the summary result generated by the system was not as systematic as that generated by the expert. The characteristics of the summary result produced by the expert were capable of setting the restriction of each summary result, while the summary result given by the system failed to restrict the number of words in the documents. The discrepancy between the number of words in the summary result given by the expert and that of the system would decrease the ROUGE score, but the factors affecting the variety of summarization results could be sorted out using an LSTM method.

#### IV. CONCLUSION

The research results have found that the LSA, CLSA, and LSTM can be used in the summarization of documents on PHI court decisions based on the testing that examines the relevance and resemblance level of the manual summary given by an expert. The most reliable result was obtained from the LSTM method. In terms of the potential model made, the summarization model based on the CLSA method can still be improved by selecting and considering pertinent pre-processing methods. In the time to come, the improvement of summary results may require the use of synonyms for each part of the document of PHI court decisions.

#### REFERENCES

- [1] M. O. Riedl, "Human-centered artificial intelligence and machine learning," *Hum. Behav. Emerg. Technol.*, vol. 1, no. 1, pp. 33–36, 2019, doi: 10.1002/hbe2.117.
- [2] D. C. Brock, "Learning from artificial intelligence's previous awakenings: The history of expert systems," *AI Magazine*, vol. 39, no. 3, pp. 3–15, 2018.
- [3] I. Roll and R. Wylie, "Evolution and Revolution in Artificial Intelligence in Education," *Int. J. Artif. Intell. Educ.*, vol. 26, no. 2, pp. 582–599, 2016, doi: 10.1007/s40593-016-0110-3.
- [4] G. Malik, D. K. Tayal, and S. Vij, "An Analysis of the Role of Artificial Intelligence in Education and Teaching," *Adv. Intell. Syst. Comput.*, vol. 707, pp. 407–417, 2019, doi: 10.1007/978-981-10-8639-7\_42.
- [5] K. Y. Lee and J. Kim, "Artificial Intelligence Technology Trends and IBM Watson References in the Medical Field," *Korean Med. Educ. Rev.*, vol. 18, no. 2, pp. 51–57, 2016, doi: 10.17496/kmer.2016.18.2.51.
- [6] L. Cui, S. Huang, F. Wei, C. Tan, C. Duan, and M. Zhou, "Superagent: A customer service chatbot for E-commerce websites," *ACL 2017 - 55th Annu. Meet. Assoc. Comput. Linguist. Proc. Syst. Demonstr.*, pp. 97–102, 2017, doi: 10.18653/v1/P17-4017.
- [7] B. Alarie, A. Niblett, and A. H. Yoon, "How artificial intelligence will affect the practice of law," *University of Toronto Law Journal*, vol. 68, no. March 2017, pp. 106–124, 2018, doi: 10.3138/utlj.2017-0052.
- [8] K. Yoko, V. C. Mawardi, and J. Hendryli, "Sistem Peringkat Otomatis Abstraktif Dengan Menggunakan Recurrent Neural Network," *Comput. J. Comput. Sci. Inf. Syst.*, vol. 2, no. 1, p. 65, 2018, doi: 10.24912/computatio.v2i1.1481.
- [9] X. Duan *et al.*, "Legal summarization for multi-role debate dialogue via controversy focus mining and multi-task learning," *Int. Conf. Inf. Knowl. Manag. Proc.*, pp. 1361–1370, 2019, doi: 10.1145/3357384.3357940.
- [10] A. Kanapala, S. Pal, and R. Pamula, "Text summarization from legal documents: a survey," *Artif. Intell. Rev.*, vol. 51, no. 3, pp. 371–402, 2019, doi: 10.1007/s10462-017-9566-2.
- [11] N. F. Saraswati, Indriati, and R. S. Perdana, "Peringkasan Teks Otomatis Menggunakan Metode Maximum Marginal Relevance Pada Hasil Pencarian Sistem Temu Kembali Informasi Untuk Artikel Berbahasa Indonesia," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 11, pp. 5494–5502, 2018, doi: 10.1016/s1010-6030(01)00380-x.

- [12] N. Savanti, W. Gotami, and R. K. Dewi, "Peringkasan Teks Otomatis Secara Ekstraktif Pada Artikel Berita Kesehatan Berbahasa Indonesia Dengan Menggunakan Metode Latent Semantic Analysis," *J. Pengemb. Teknol. Inf. dan Ilmu Komput. Univ. Brawijaya*, vol. 2, no. 9, pp. 2821–2828, 2018.
- [13] Y. M. Sari and N. S. Fatolah, "Automatic Text Summarization in Indonesian Language Learning Module Using Cross Latent Semantic Analysis (CLSA) Method," *J. Edukasi dan Penelit. Inform.*, vol. 7, no. 2, p. 153, 2021, doi: 10.26418/jp.v7i2.47768.
- [14] G. Mandar and G. Gunawan, "Peringkasan dokumen berita bahasa indonesia menggunakan metode cross latent semantic analysis," *Regist. J. Ilm. Teknol. Sist. Inf.*, vol. 3, no. 2, pp. 94–104, 2017, doi: 10.26594/register.v3i2.1161.
- [15] D. Suleiman and A. A. Awajan, "Deep Learning Based Extractive Text Summarization: Approaches, Datasets and Evaluation Measures," in *2019 6th International Conference on Social Networks Analysis, Management and Security, SNAMS 2019, Oct. 2019*, pp. 204–210, doi: 10.1109/SNAMS.2019.8931813.
- [16] W. Jiang, Y. Zou, T. Zhao, Q. Zhang, and Y. Ma, "A hierarchical bidirectional LSTM sequence model for extractive text summarization in electric power systems," in *Proceedings - 2020 13th International Symposium on Computational Intelligence and Design, ISCID 2020*, Dec. 2020, pp. 290–294, doi: 10.1109/ISCID51228.2020.00071.
- [17] R. M. Patel and A. J. Goswami, "Abstractive Text Summarization with LSTM using Beam Search Inference Phase Decoder and Attention Mechanism," Jun. 2021, doi: 10.1109/ICCI52257.2021.9484880.
- [18] R. Karmakar, K. Nirantar, P. Kurunkar, P. Hiremath, and D. Chaudhari, "Indian Regional Language Abstractive Text Summarization using Attention-based LSTM Neural Network," Jun. 2021, doi: 10.1109/CONIT51480.2021.9498309.
- [19] D. Patel, N. Shah, V. Shah, and V. Hole, "Abstractive Text Summarization on Google Search Results," *Proc. Int. Conf. Intell. Comput. Control Syst. ICICCS 2020*, pp. 538–543, May 2020, doi: 10.1109/ICICCS48265.2020.9120998.
- [20] K. Merchant and Y. Pande, "NLP Based Latent Semantic Analysis for Legal Text Summarization," in *2018 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2018*, Nov. 2018, pp. 1803–1807, doi: 10.1109/ICACCI.2018.8554831.
- [21] K. Agrawal, "Legal case summarization: An application for text summarization," Jan. 2020, doi: 10.1109/ICCC48352.2020.9104093.
- [22] A. W. Pradana and M. Hayaty, "The Effect of Stemming and Removal of Stopwords on the Accuracy of Sentiment Analysis on Indonesian-language Texts," *Kinet. Game Technol. Inf. Syst. Comput. Network, Comput. Electron. Control*, vol. 4, no. 3, pp. 375–380, 2019, doi: 10.22219/kinetik.v4i4.912.
- [23] A. Tabassum and R. R. Patil, "A Survey on Text Pre-Processing & Feature Extraction Techniques in Natural Language Processing," *Int. Res. J. Eng. Technol.*, no. June, pp. 4864–4867, 2020.
- [24] S. Alam and N. Yao, "The impact of preprocessing steps on the accuracy of machine learning algorithms in sentiment analysis," *Comput. Math. Organ. Theory*, vol. 25, no. 3, pp. 319–335, 2019, doi: 10.1007/s10588-018-9266-8.
- [25] S. Vijayarani, M. J. Ilamathi, M. Nithya, A. Professor, and M. P. Research Scholar, "Preprocessing Techniques for Text Mining -An Overview," vol. 5, no. 1, pp. 7–16.
- [26] A. Amalia, D. Gunawan, Y. Fithri, and I. Aulia, "Automated Bahasa Indonesia essay evaluation with latent semantic analysis," *J. Phys. Conf. Ser.*, vol. 1235, no. 1, 2019, doi: 10.1088/1742-396/1235/1/012100.
- [27] K. Al-Sabahi, Z. Zhang, J. Long, and K. Alwasabi, "An Enhanced Latent Semantic Analysis Approach for Arabic Document Summarization," *Arab. J. Sci. Eng.*, vol. 43, no. 12, pp. 8079–8094, 2018, doi: 10.1007/s13369-018-3286-z.
- [28] J. W. G. Putra, "Pengenalan Konsep Pembelajaran Mesin dan Deep Learning," vol. 4, no. August, pp. 1–235, 2019.
- [29] G. Van Houdt, C. Mosquera, and G. Nápoles, "A review on the long short-term memory model," *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5929–5955, 2020, doi: 10.1007/s10462-020-09838-1.
- [30] H. Chung and K. S. Shin, "Genetic algorithm-optimized long short-term memory network for stock market prediction," *Sustain.*, vol. 10, no. 10, 2018, doi: 10.3390/su10103765.
- [31] K. Kurniawan and S. Louvan, "INDOSUM: A New Benchmark Dataset for Indonesian Text Summarization," *2018 Int. Conf. Asian Lang. Process.*, pp. 215–220, 2018.
- [32] S. Ghodrathnama, A. Beheshti, M. Zakershaharak, and F. Sobhanmanesh, "Extractive Document Summarization Based on Dynamic Feature Space Mapping," *IEEE Access*, vol. 8, pp. 139084–139095, 2020, doi: 10.1109/ACCESS.2020.3012539.



# Text Summarization on Verdicts of Industrial Relations Disputes Using the Cross-Latent Semantic Analysis and Long Short-Term Memory

## ORIGINALITY REPORT

6%

SIMILARITY INDEX

7%

INTERNET SOURCES

5%

PUBLICATIONS

3%

STUDENT PAPERS

## PRIMARY SOURCES

1	<a href="http://stmik-budidarma.ac.id">stmik-budidarma.ac.id</a> Internet Source	1%
2	<a href="http://ejurnal.seminar-id.com">ejurnal.seminar-id.com</a> Internet Source	1%
3	<a href="http://thesai.org">thesai.org</a> Internet Source	1%
4	<a href="http://www.jatit.org">www.jatit.org</a> Internet Source	1%
5	Nur Hayatin, Gita Indah Marthasari, Syadza Anggraini. "Improvement of Cluster Importance Algorithm with Sentence Position for News Summarization", 2018 5th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), 2018 Publication	1%
6	Stephen H. Fairclough. "Designing human-computer interaction with neuroadaptive	1%

# technology", Elsevier BV, 2022

Publication

---

7	<a href="http://ia800805.us.archive.org">ia800805.us.archive.org</a>	1 %
<hr/>		
8	<a href="http://jurnal.iaii.or.id">jurnal.iaii.or.id</a>	1 %
<hr/>		
9	<a href="http://vdocuments.site">vdocuments.site</a>	1 %
<hr/>		
10	<a href="http://ijeecs.iaescore.com">ijeecs.iaescore.com</a>	1 %
<hr/>		
11	<a href="http://repository.unsri.ac.id">repository.unsri.ac.id</a>	1 %

---

Exclude quotes  On

Exclude matches  < 1%

Exclude bibliography  Off