Sentiment Analysis of Indonesian Economics News Summary on The ICI using Long Short Term Memory

Muhammad Raihan Muhith[1], Imelda Atastina[2*]

[1,2]Faculty of Informatics, Informatics Department, Telkom University, Bandung, Indonesia
Email: [1]raihanmuhith@student.telkomuniversity.ac.id, [2*]imelda@telkomuniversity.ac.id
Correspondent Author Email: imelda@telkomuniversity.ac.id

Abstract− Many investors use news summary as a reference while deciding whether to purchase, sell, or hold shares as part of their investment activity. The information provided by the news, on the other hand, frequently fails to meet expectations. Therefore, this study aims to classify economic news summary using sentiment analysis method followed by analyzing the results to determine the correlation between Indonesian economic news summaries and Indonesia Composite Index (ICI) trends. The dataset for this study is a collection of news articles from the Kontan online news site. This study begins with the collection of a dataset that includes 1720 training data and 440 test data that have been automatically summarized using the cosine similarity approach and labeled (positive/negative) based on the ICI trend the next day after the data was crawled. We use Term Frequency – Inverse Document Frequency (TF-IDF) as the word weighting technique and Long Short Term Memory (LSTM) method as the classifier to build the model. From the model training process that has been carried out for eight trials by tuning hyperparameters, the training accuracy is 86% and the F1-Score is 89.3%. While the testing accuracy and F1-score are 68.1% and 75.8% consecutively.

Keywords: News Summary; Sentiment Analysis; ICI Trends; TF-IDF; LSTM

1. INTRODUCTION

Stocks are a type of investment that can deliver appealing benefits to investors, making them a viable alternative for investors. But in fact, stock investment is not a simple thing to do. Investors must consider several parameters to reduce stock investment risk, which has a significant impact on stock price fluctuation. The ICI is an indicator of stock price movements that shareholders can use as a guide to assess the trend of stock movements and decide whether to buy, hold, or sell their shares [1].

Many factors can influence stock price changes, including company events, company policy, economic situations, etc. In this modern era, these factors can now be known by the general public through mass media such as economic news websites. In making short-term stock investments, stock price trends tend to up and down due to positive and negative news [2]. Therefore, investors should keep an eye on the news in order to forecast ICI's future direction and make stock investing decisions. On the other side, it is important to note to find out the meaning contained in a piece of writing does not necessarily have to read the entire content. A summary is typically used to extract the key ideas from a piece of writing in order to make it more efficient. This also holds true for news; investors just need to consider the key information in a piece of news when making stock investment selections. With this background, we use the news summary rather than its entirety as the object of our research.

Nowadays, knowledge has certainly developed a lot. The use of text mining to predict a stock trend based on textual data is something that has been widely used in the investment world. Using the sentiment analysis process to overcome the inaccuracy of news on ICI trends is one of the options. Sentiment analysis is a technique in text mining fields that extract the opinions included in a sentence or document and categorizes their polarity [3][4].

Based on the background that has been described, we propose a classification model that can detect the sentiment polarity of opinions contained in news summaries based on the ICI trend. We use Indonesian economic news data crawled from online news sites, Kontan, and summarized using a sentence similarity approach. The data gathered, is split into data train and data test. Data train is used for the classification model while data test is used to evaluate the model results. This study aims to classify economic news summaries via sentiment analysis followed by analyzing the accuracy and F1-Score of the results to determine the correlation between Indonesian economic news summaries and ICI trend.

To complete sentiment analysis tasks, there are several methods used such as Naive Bayes, support vector machine, and random forest as performed by Joshi et al. on 2016 [5] and Gupta and Chen on 2020 [6]. [5] discussed the prediction of stock price trends on financial news articles with news sentiment classification. This research uses three machine learning methods (Random Forest, Naïve Bayes, and SVM) and two feature selection techniques (bag-of words and TF-IDF) tested against several scenarios using N fold-cross validation (N = 5, 10, and 15), 70% split data, 80% split data, and new test data. The results show that the Random Forest method produces very high accuracy for all test cases, which is 88% to 92%. Followed by SVM which has an average accuracy of 86%. While the accuracy of the Naive Bayes algorithm is around 83%. The research concluded that stock trends can be predicted using news articles and previous price history.

Meanwhile, [6] discussed sentiment analysis for stock price prediction. This research investigates the sentiment impact of a nine-month collection of tweets expressed on StockTwits for five companies, namely Apple, Amazon, General Electric, and Target. This research utilizes three machine learning methods (Naïve Bayes, Logistic Regression, and SVM) and five feature extraction techniques (bag-of-words, bigram, trigram, TF-IDF, and LSA). In this study, the combination of Logistic Regression and TF-IDF gives a fairly high average accuracy rate, which is between 75% and 85% for the five companies. On the other hand, this study compared the accuracy of stock price changes with sentimented and non-sentimented tweets. The accuracy obtained for the sentimented tweets is higher than the non-sentimented ones for all five companies. This provides reasonable evidence that sentiment data has a positive impact on the accuracy of predicting stock price changes.

Nonetheless, as shown in the following research [7], [8], and [9], the studies performed emotion detection from a text expression using the LSTM method. The object analyzed differs among the three studies: Goud and Garg on 2021 [7] analyses text expression from Amazon web service datasets in Microsoft Azure Jupyter notebooks, Miedima on 2018 [8] from IMDB movie reviews, and Li and Qian on 2016 [9] from JD.COM website comments and Ctrip reviews. Based on the three studies' test results, it was confirmed that the deep learning approach, LSTM, is better than the other methods and effective for sentiment recognition. By considering these three studies results, we use LSTM as a method for sentiment analysis of economics news summaries based on the ICI trends to get similar results.

Based on related studies that have been described, this topic has been discussed several times by various researchers, where these studies focus on emotion detection on English datasets, thus we conducted research with sentiment analysis topics that focus on Indonesian datasets, where the focus on this topic is still rarely found.

2. RESEARCH METHODOLOGY

2.1 Research Stages

In this study, sentiment analysis of news summaries based on ICI trends is performed using TF-IDF feature extraction and LSTM deep learning method. The stages we carried out during the research process are shown in Figure 1. There are eight steps that we apply in this study process, The research begins with retrieving data by the crawling method then labeled according to the ICI trend on the next day, then the dataset is preprocessed with the aim of providing quality data. After that, each data is summarized using the sentence similarity method and then proceeded to the feature extraction stage using TF-IDF. Before entering the modeling stage we separate the dataset into training data and test data in the data splitting stage, where the training data proceeds to the modeling stage while the test data is used for the last stage of this research, model testing.

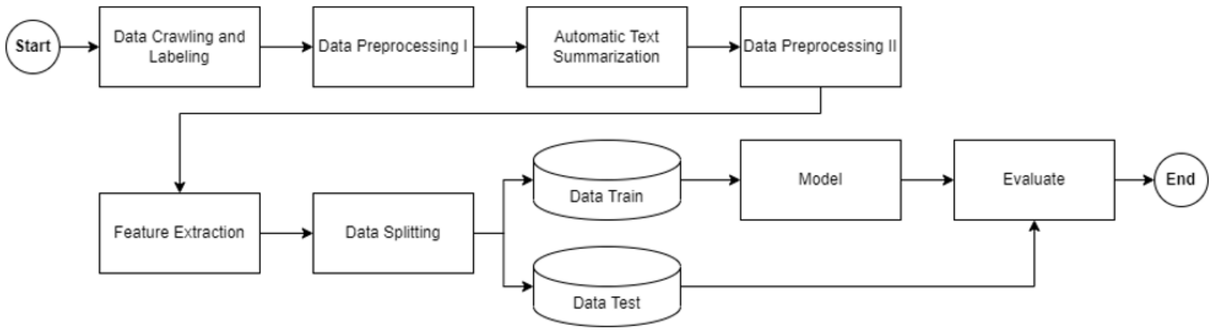

Figure 1. Research Stages

2.2 Data Crawling and Labeling

The data is taken from the economics news website, Kontan, through a Python-based crawling technique. News articles published between October 1, 2021, and May 31, 2022, are those that were taken from Kontan where only 10 news articles were taken per date. The data that has been collected from as many as 2160 news is then labeled based on the ICI trend on the following day. as demonstrated in Figure 2, all news retrieved on October 1, 2021, is labeled positive. this is due to the assumption that the news has a positive impact on the ICI trends. While all news taken on October 2, 2021, is negatively labeled due to the assumption that the news has a negative impact on the change in ICI trend. So if we look back at Figure 2, then the labels for the news taken on October 3 and 4 are positive, and so on.

Figure 2. Ilustration of news articles labeling based on ICI trend

2.3 Preprocessing

One of the objectives of this stage is to provide high-quality data. high-quality data will have a positive impact on model training results, especially on textual data. Textual input frequently contains noise and irregular forms, resulting in low-quality features that are used in the model training. There are two stages of preprocessing that we apply, preprocessing I and preprocessing II. In order to produce accurate summary results without reducing the original text's intent, the initial preprocessing stage cleans the data before the automatic text summarizing stage. The goal of the preprocessing II, which is carried out after the automatic text summarization stage, is to complete any preprocessing that was not completed in preprocessing I, such as stemming and removing frequent words. Figure 3 shows the preprocessing steps that we performed. The explanation of each stage is as follows.

a. Normalization: Normalization is a process that we use to remove noise. Noise is useless characters such as punctuation marks (dots, commas, question marks, etc.) and numbers. In addition, this stage also performs case folding, which is a process that converts all letters into lowercase letters.
b. Removing Stopwords: stopwords are a set of words that are inferred to have no meaning or impact on the sentiment polarity of a text. Some examples of Indonesian stopwords are 'aku', 'ke', 'yang', 'ini', etc. We apply this stage with the aim of removing meaningless words as well as reducing the number of features used for modeling.
c. Stemming: Stemming is a technique used to transform words with affixes into the proper basic words. This step groups words with similar base words and meanings but different forms due to affixes. For example, the words "menangis" and "tangisan" have similar base words, "tangis".
d. Removing frequent words: since an overused word will have an impact on the inconsistency of a sentiment model, therefore it is necessary to remove these words. Basically, this stage is a sub-stage of removing stopwords, as the goal of both stages is to remove unwanted words.

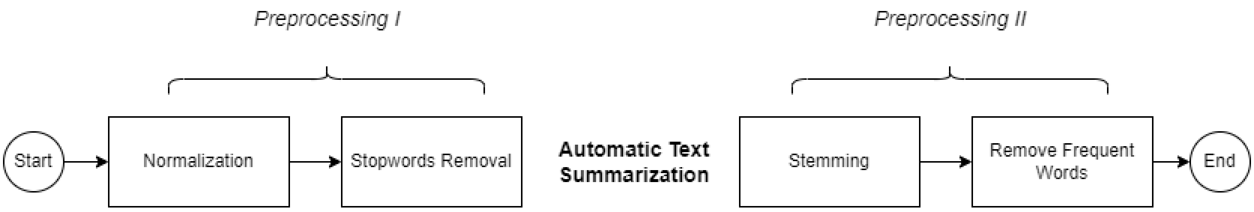

Figure 3. Preprocessing Stages

2.4 Automatic Text Summarization

This stage is the process of summarizing data by applying the sentence similarity approach. This approach calculates the degree of similarity between phrases or sentences [10], with the aim to determine the level of importance used to generate a summary [10]. The importance of a sentence is decided based on the statistics and linguistic features of the sentence [11]. we use cosine distance calculation to find the similarity of sentences in a document. With a value between [-1, 1], cosine similarity calculates the degree of similarity between two sentences or documents [12]. The cosine similarity diagram is shown in Figure 4 with v1 and v2 as two vectors that represent their corresponding sentences. The angle between the two vectors (θ) represents the similarity between sentences, the smaller the angle produced between the two vectors, the higher the level of similarity between the two sentences. Otherwise, the larger the angle produced between the two vectors, the lower the level of similarity between the two sentences [13].

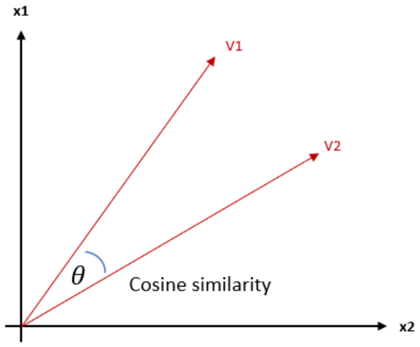

Figure 4. Cosine similarity between two vectors

In the application process (see Figure 5), the first thing is to segment each sentence in a document, next, each sentence is calculated to be similar to each other using the cosine similarity formula which is stored in a matrix. Then, the similarity matrix is ranked using nx.pagerank library in determining the overall similarity score of a sentence with its document and the top N (N: the number of sentences taken to be summarized) is used as the summary. We define N = 5 as the number of sentences taken to be summarized, based on the minimum length of an article in the dataset is 5.

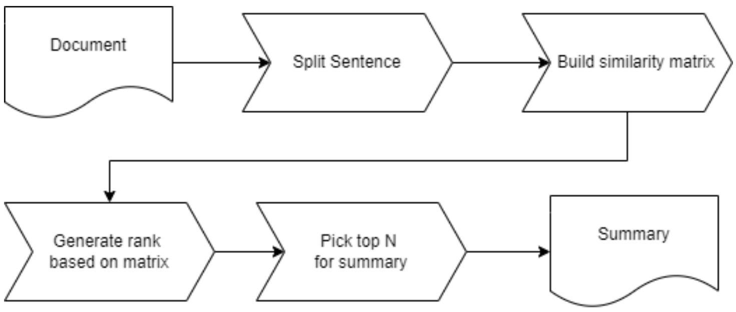

Figure 5. The Extractive Text Summarization Process

2.5 Feature Extraction

Feature extraction is used to convert textual data into numerical numbers. We use Term Frequency - Inverse Document Frequency (TF-IDF) where this method is a way to determine the weight of a word's relationship to a document based on the relevance of keywords to the corpus [14]. The TF and IDF calculations are the two key elements of the TF-IDF method [14]. The frequency of a word appearing in a document is measured by its term frequency (TF), and the frequency of documents containing that word is measured by its inverse document frequency (IDF). To put it simply, TF-IDF is used to find out how often a word appears in a document. Equation 1 shows the IDF calculation formula where idf_i represent the IDF score of word i, D is the total documents in the corpus and df_i is the total documents containing word i. Equation 2 shows the word weighting formula with the TF-IDF method where W_{ij} is the weight of word i in document j and tf_{ij} is the frequency of occurrence of word i in document j [15]. While Equation 3 calculates the average weight of a word i (W_i) on a set of documents.

$$idf_i = \log(\frac{D}{df_i}) \quad (1)$$

$$W_{ij} = tf_{i_j} \times idf_i \quad (2)$$

$$avg\ W_i = \frac{\sum_{j=0}^{D} W_{ij}}{D} \quad (3)$$

2.6 Data Splitting

The dataset is split into training and test data before classification, creating two subsets. Classification models are trained using training data and then evaluated using test data. In determining the optimal ratio, we tested the model with a splitting ratio of training data and test data of 70:30, 80:20, and 90:10. The best results were seen at a splitting ratio of 80:20, thus the number of training data and test data were 1720 and 440, respectively. In order to maintain the original ratio of the number of positive and negative classes, the distribution of classes in data train and data test is done randomly. The distribution of positive and negative class comparisons in the training and test data is shown in Figure 6.

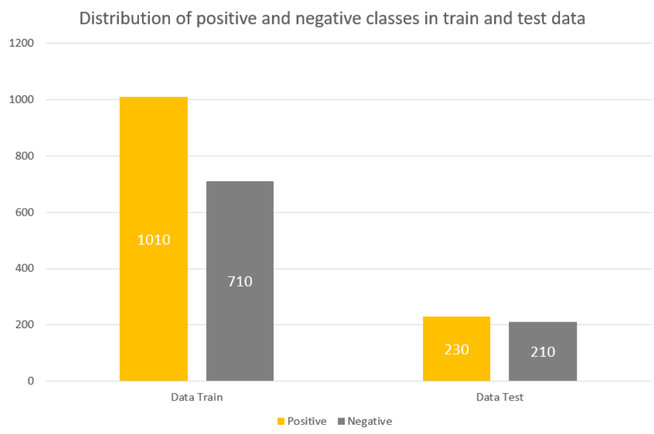

Figure 6. Distribution of positive and negative classes in train and test data

2.7 Modelling

The purpose of this stage is to classify economic news that has been labeled by training and learning the characteristics contained in the news and whether the news has a positive or negative impact on the ICI through sentiment analysis techniques. To fulfill this purpose, we apply the Long Short Term Memory (LSTM) method to build the classifier model. LSTM is a deep learning algorithm modified from Recurrent Neural Networks (RNNs) by adding a memory cell (hidden layer) that can store information for a long period of time [16]. LSTMs are formed to overcome the problem of vanishing gradients by adding a set of gates that are used to control when information enters memory [17]. It also overcomes the problem of training on long sequential data (various features) [18].

Essentially, this method implements a looping neural network where each loop returns results that have gone through four stages, the first of which decides how much information needs to be removed. This decision is operated by a gate called the forget gate layer (f_t). The second step is to decide how much information needs to be retained and used. This decision is operated by the input gate layer (i_t) and also passes through the Hyperbolic tangent activation function (*tanh*). The third stage is to remove the information obtained from the forgate layer and then combined it with the results of the second stage to add new information into a new memory cell (C_t). Then the last is the stage in determining what information will be generated. This decision is operated by a gate called output gate (o_t). All these stages are used to calculate the output of the hidden layer (h_t). The calculations used are shown in equations 4, 5, 6, 7, and 8 [19], where σ is a sigmoid activation function, then W_f, W_i, W_c, and W_o are weight matrices, then b_f, b_i, b_c, and b_o are bias vectors, x_t is the present term, and h_{t-1} is the output of the previous hidden layer.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f \qquad (4)$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i \qquad (5)$$
$$C_t = f_t * C_{t-1} + i_t * \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \qquad (6)$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o \qquad (7)$$
$$h_t = o_t * \tanh C_t \qquad (8)$$

We did some transformation to the clean data by transforming the dataset into 3-dimensional time series data. The procedure is to group the data by date, where each date has 10 articles that have been summarized. After that, each summary is represented as a vector with a length that has been determined at the feature extraction stage.

2.8 Model Evaluation

To be able to measure the performance and feasibility of the model that has been built, we use the confusion matrix table as shown in Table 1. From Table 1, we know that True Positive (TP) is the number of positive data predicted as positive data, False Negative (FN) is the number of positive data predicted as negative data, False Positive (FP) is the number of negative data predicted as positive data, and True Negative (TN) is the number of negative data predicted as negative data [20].

Table 1. Confusion Matrix

	Actually Positive	Actually Negative
Predicted Positive	TP	FP
Predicted Negative	FN	TN

In addition, the Confusion matrix can also be used to calculate various performance metrics to measure the performance of the model that has been created. The performance metrics used are accuracy and F1-Score. We use accuracy metrics to see the performance of the model in classifying each class, while F1-Score is used because the distribution of classes in the training data tends to be unbalanced (see Figure 5). However, in determining F1-Score, auxiliary metrics such as precision and recall are needed. Precision shows the level of accuracy between the information requested and the answer given by the model, while recall measures the success rate of the model in detecting information. The calculation of accuracy, precision, recall, and F1-Score is shown in equations 9, 10, 11, and 12.

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{N data test}} \tag{9}$$

$$\text{Precision} = \frac{\text{TP}}{\text{FP+TP}} \tag{10}$$

$$\text{Recall} = \frac{\text{TP}}{\text{FN+TP}} \tag{11}$$

$$\text{F1} - \text{Score} = \frac{2 \times (\text{recall} \times \text{precision})}{(\text{recall} + \text{precision})} \tag{12}$$

3. RESULTS AND DISCUSSION

3.1 Data Preprocess Results

It has been explained earlier that in this research the data preprocessing stage is divided into two, before and after text summarization. The first preprocessing produces clean data for text summarization, while the second preprocessing produces high-quality data for modeling. The results of each preprocessing stage (see Figure 3): normalization, stopwords removal, stemming, and remove frequent words are shown in Table 2, Table 3, Table 4, and Table 5.

Table 2. Normalization result

Input	Output
"Harga emas batangan bersertifikat Antam keluaran Logam Mulia PT Aneka Tambang Tbk (ANTM) naik pada Kamis (14/10) mengutip situs Logam Mulia."	"harga emas Batangan bersertifikat antam keluaran logam mulia pt aneka tambang tbk naik pada kamis mengutip situs logam mulia."

From Table 2, the result of normalization is data that has been done case folding, removal of punctuation marks (except periods) including numbers, and removal of words within parentheses.

Table 3. Stopwords Removal result

Input	Output
"harga emas Batangan bersertifikat antam keluaran logam mulia pt aneka tambang tbk naik pada kamis mengutip situs logam mulia."	"harga emas batangan bersertifikat antam keluaran logam mulia aneka tambang naik mengutip situs logam mulia."

From Table 3, the stopwords removal stage produced data free of words with no context or meaning. In the output section of Table 3, we know that the words "pt," "tbk," and "kamis" are removed from the text. To facilitate this stage, we use a list of Indonesian stopwords, which is combined with a list of stock codes listed on the Indonesia Stock Exchange (IDX).

Table 4. Stemming result

Input	Output
"harga emas batangan bersertifikat antam keluaran logam mulia aneka tambang naik mengutip situs logam mulia."	"harga emas batang sertifikat antam keluar logam mulia aneka tambang naik kutip situs logam mulia."

After normalization and removal of stopwords, we found that the data was ready for text summarization and then the second preprocessing stage was carried out. Table 4 shows the results of the stemming, where the output after stemming is a text that is reverted to its base word such as "batang", "sertifikat", "keluar", and "kutip" is the base words of "batangan", "bersertifikat", "keluaran", and "mengutip" respectively. The last step is to remove the most frequently used words in each news summary. We count the number of occurrences of each word in the corpus to define a list of words to be removed based on the most used words. Figure 6 shows the 10 most frequently used words with their number of appearances in the corpus.

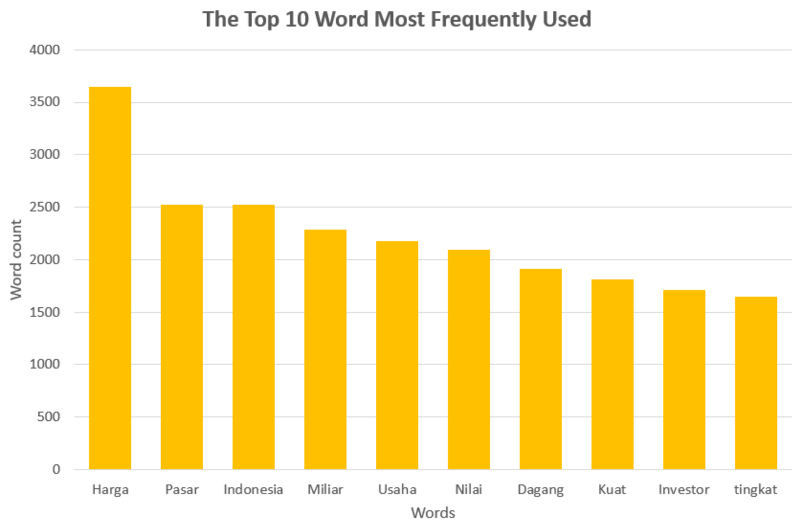

Figure 7. The top 10 words most frequently used

We then proceeded to remove the words that had been defined in the list that has been showed in Figure 7, therefore as can be seen from the output section of table 5, we know that the word "harga" has been removed.

Table 5. Remove Frequent Words result

Input	Output
"harga emas batangan bersertifikat antam keluaran logam mulia aneka tambang naik mengutip situs logam mulia."	"emas batang sertifikat antam keluar logam mulia aneka tambang naik kutip situs logam mulia."

3.2 Text Summarization

The output of the automatic text summarization stage is a summarized news article, where the summarization process is done by taking sentences that have a high number of similarities (see Figure 4). As explained earlier, the similarity score is calculated based on the cosine similarity between sentences and combine the scores using nx.pagerank library to determine the overall similarity value of a sentence with its document. Table 6 shows the similarity score from each sentence in a document.

Table 6. Sentence similarity from a document

No	Sentence	Similarity Score
1.	membuka rekening phintraco sekuritas calon investor dana.	0.0917
2.	menyambut inklusi keuangan phintraco sekuritas modal investasi calon investor membuka rekening aplikasi android profits anywhere.	0.0912
3.	direktur utama phintraco sekuritas jeffrey hendrik phintraco sekuritas berinovasi kemudahan calon investor berinvestasi pasar modal indonesia.	0.0909
4.	pemberian modal menyemarakkan inklusi keuangan dicanangkan otoritas jasa keuangan kesempatan masyarakat indonesia investor pasar modal indonesia.	0.0898
5.	modal investasi calon investor phintraco sekuritas berinovasi kemudahan calon investor membuka rekening kemudahan pendaftaran full online mengirimkan dokumen fisik.	0.0897
…	.....	….
11.	masyarakat diajak memanfaatkan kesempatan berlaku desember.	0.0796
12.	langkah mudah tanda tangan full digital pendaftaran menit.	0.0316

Based on our analysis, the shortest article length has 5 sentences while the average article length has at least 22 sentences. Taking this into consideration, we used the five sentences with the highest similarity score to be used as the summary. The summary results are shown in Table 7.

Table 7. Summary result

Output
membuka rekening phintraco sekuritas calon investor dana. menyambut inklusi keuangan phintraco sekuritas modal investasi calon investor membuka rekening aplikasi android profits anywhere. direktur utama phintraco sekuritas jeffrey hendrik phintraco sekuritas berinovasi kemudahan calon investor berinvestasi pasar modal indonesia. pemberian modal menyemarakkan inklusi keuangan dicanangkan otoritas jasa keuangan kesempatan masyarakat indonesia investor pasar modal indonesia. modal investasi calon investor phintraco sekuritas berinovasi kemudahan calon investor membuka rekening kemudahan pendaftaran full online mengirimkan dokumen fisik.

3.3 Feature Extraction

In order to transform textual data into numerical data, as explained in section 2—particularly in the feature extraction method, we used the TF-IDF technique to carry out feature extraction and word weighting. Equations 2 and 3 are used in the weighting method to determine a word's importance to a document. According to the results of this feature extraction, the corpus contains 9337 unique features. In this study, the maximum number of features used is 1000 features with the assumption of improving the training model results. Table 8 shows the five features with the highest importance in the corpus using Equation 3. Based on equation 3, then a term has an average TF-IDF score of the total number of $W_{\text{"term"}\ in\ document\ 1,2,3,\ldots,2160}$ divided by the number of documents in the corpus which is 2160.

Table 8. Most important word based on TF-IDF calculation results

Words	TF-IDF Score
total	0.0301
capai	0.0282
transaki	0.0271
triliun	0.0267
usaha	0.0259

3.4 Model Training

Modeling is carried out by classifying news summary data into positive and negative sentiment polarities. Therefore the classification we conduct is a binary classification which categorize the data into positive and negative. The data used is training data with 1720 records that have successfully passed feature extraction and preprocessing. We use the Keras Machine Learning API in implementing the LSTM method along with other functions needed. The architecture of the model we built is shown in Table 9.

Table 9. Model Architecture

Layer	Type	Neurons
Hidden Layer	LSTM	64
	Dropout	0.1
Output Layer	Dense	256, 128, 1

We use five layers to build the classifier model, which are two hidden layers, and three output layers. From Table 10, we use LSTM, and Dropout as the hidden layer, and three Dense as the output layer. In Addition, there are some parameters chosen that are default from Keras API such as tanh and sigmoid activation functions, while the optimization algorithm used is Adam. The training model results show excellent training accuracy, with an accuracy value of 97.6% and F1-Score of 98%.

3.5 Model Evaluation

This section is the testing stage of the pre-trained model. The model has been trained for 8 training epochs and the batch size is 32. We tested the feasibility of the model using 440 records of test data containing 210 negative sentiment data and 230 positive sentiment data (see figure 5). Based on the test results, there are 110 positive data predicted positive (TP), 90 negative data predicted positive (FP), 120 negative data predicted negative (TN), and 130 positive data predicted negative (FN). Table 10 shows the testing accuracy, precision, recall, and F1-Score of the model by using formulas 9, 10, 11, and 12 based on the results of TP, TN, FP, and FN. The results of testing the model appear to produce testing accuracy of 50%. This means that the model can only predict an ICI trend based on the news with only a 50% chance of being correct.

Table 10. Model Testing result

Accuracy	Precision	Recall	F1-Score
50%	53%	48%	48%

To improve the results, by tuning the hyperparameters, we intend to retrain the model and directly test with the test data. This was performed to get a good model on both the training and testing sides. The hyperparameter tuning experiment was conducted eight times, wherein in each experiment we tune the hyperparameters by changing the epochs and batch size. The results of each experiment can be seen in Table 11.

Table 11. Experiment results

	Training		Testing	
Batch Size and Epochs	Accuracy	F1-Score	Accuracy	F1-Score
16 and 8	98.9%	99.0%	45.4%	57.1%
16 and 10	100%	100%	45.4%	57.1%
16 and 15	100%	100%	40.9%	43.4%
32 and 10	100%	100%	50.0%	47.6%
32 and 15	99.4%	99.5%	40.9%	13.3%
64 and 8	57.2%	72.7%	59.0%	74.2%
64 and 10	86.0%	89.3 %	68.1%	75.8%
64 and 15	95.5%	96.4%	54.5%	68.7%

From Table 12, in each experiment, we compared the accuracy metrics of training with testing and the F1-Score of training with testing. If we consider the F1-Score value of the testing side of the model, the experimental results show that the best results are in the model using a batch size of 64 with the number of epochs is 10. This model provides quite satisfying results, where the percentage of testing F1-Score can reach 75.8%. In addition, this model also turns out to have the best test accuracy value among other experiments, which is 68.1%.

However, when considering the F1-Score value from the model training side, there are three models with the best F1-Score value. Which are the model using a batch size of 16 with 10 epochs, the model using a batch size of 16 with 15 epochs, and finally the model using a batch size of 32 with 10 epoch. These three models provide an excellent training F1 Score value of 100%, and it emerges that these models' training accuracy is also 100%.

Overall, if we consider the model from both sides. we choose the model that uses a batch size of 64 with 10 epochs. There are two reasons, first because this model gives the best testing results despite the relatively low training results compared to other experiments. Second, the difference between the training and testing numbers of this model is not too far (standard overfitting). From the experimental result shown in table 11, we know that the difference between training and the testing accuracy is 17.9%, and the difference between training and testing F1-Score is 13.5%. Table 12 shows the performance of the model in the form of a confusion matrix.

Table 12. Confusion Matrix classification result

	Actually Positive	Actually Negative
Predicted Positive	50.00%	29.55%
Predicted Negative	2.27%	18.18%

From Table 12, the number of positive sentiment data that is predicted positive (TP) is 220 or 50%. The number of positive sentiment data that is predicted negative (FN) is 10 or 2.27%. The number of negative sentiment data that is predicted negative (TN) is 80 or 18.18%. The number of negative sentiment data that is predicted positive (FP) is 130 or 29.55%. In other words, the model classifies 350 positive data (TP + FP) and 90 negative data (TN + FN).

Hyperparameter tuning experiments turned out to provide better results than before, the model's testing accuracy increased from 50% to 68,1% and its testing F1-Score increased from 48% to 75,8%. Although if we check the accuracy and F1-Score from the result of the model training side, this does not hold true. The training accuracy dropped from 97.6% to 86% while the training F1-Score dropped from 98% to 89,3%. Considering the model testing accuracy and F1-Score, we will continue the research in the future to get better testing result for predicting ICI trend based on Indonesian economic news summary. We will consider to pay more attention to the words that have the highest weight according to the type of sentiment and word weighting technique in the sentence.

4. CONCLUSION

Based on the results and discussion of this research, there are two things that can be concluded. First, the sentiment classification model on Indonesian economic news summaries on ICI trends produces accuracy and F1-scores that can reach 100% from the training model. This proves that the Indonesian economic news summary and ICI trend have a very good correlation value. In other words, the Indonesian economic news summary influences the ICI trend, so that investors can consider to use economic news summary as a tools to help them to make investation decision. Furthermore, in this study we tune the batch size and epochs hyperparameters where the best result from all eight experiments lies in the model with batch size of 64 and epochs 10. This model produces accuracy and F1-Score of 86% and 89.3% from training side, while 68.1% and 75.8% from testing side. Based on the evaluation results, this model has a positive class distribution of 350 and a negative class distribution of 90. Thus, what can be concluded from the discussion is that this model is more likely to classify data into positive classes than negative classes due to unbalanced classes in the training phase. For future research in terms of improving or enhancing the quality of the model built, we can tune the hyperparameters of the layer/architecture combination used as well as other term weighting technique to the sentiment class. In addition, the economic news summary topics used will be focused on economic news that only talks about ICI.

REFERENCES

[1] G. N. Arviana, “Sebelum Investasi, Pahami Dulu Apa Itu IHSG dan Berbagai Fungsinya,” *8 March*, 2021. https://glints.com/id/lowongan/apa-itu-ihsg/#.YsBDAWhBxPZ (accessed Jul. 02, 2022).

[2] R. Budiman, “Strategi Membeli dan Menjual,” in *Investing is Easy Teknik Analisa dan Strategi Investasi Saham untuk Pemula*, Fathur, Ed. Jakarta: PT Elex Media Komputindo, 2017, p. 84.

[3] W. Medhat, A. Hassan, and H. Korashy, “Sentiment analysis algorithms and applications: A survey,” *Ain Shams Eng. J.*, vol. 5, no. 4, 2014, doi: 10.1016/j.asej.2014.04.011.

[4] S. Mohan, S. Mullapudi, S. Sammeta, P. Vijayvergia, and D. C. Anastasiu, “Stock price prediction using news sentiment analysis,” 2019, doi: 10.1109/BigDataService.2019.00035.

[5] K. Joshi, B. H. N, and J. Rao, “Stock Trend Prediction Using News Sentiment Analysis,” *Int. J. Comput. Sci. Inf. Technol.*, vol. 8, no. 3, 2016, doi: 10.5121/ijcsit.2016.8306.

[6] R. Gupta and M. Chen, “Sentiment Analysis for Stock Price Prediction,” 2020, doi: 10.1109/MIPR49039.2020.00051.

[7] A. Goud and B. Garg, “Sentiment Analysis Using Long Short-Term Memory Model in Deep Learning,” 2021, doi: 10.1007/978-3-030-47560-4_2.

[8] F. Miedema, “Sentiment analysis with long short-term memory networks,” *Vrije Univ. Amsterdam*, vol. 1, 2018.

[9] D. Li and J. Qian, “Text sentiment analysis based on long short-term memory,” 2016, doi: 10.1109/CCI.2016.7778967.

[10] R. Ferreira, R. D. Lins, F. Freitas, S. J. Simske, and M. Riss, “A new sentence similarity assessment measure based on a three-layer sentence representation,” 2014, doi: 10.1145/2644866.2644881.

[11] N. Moratanch and S. Chitrakala, “A survey on extractive text summarization,” 2017, doi: 10.1109/ICCCSP.2017.7944061.

[12] M. S. Pawar and S. Rathod, “Text summarization using cosine similarity and clustering approach,” EasyChair, 2020.

[13] R. H. Singh, S. Maurya, T. Tripathi, T. Narula, and G. Srivastav, “Movie recommendation system using cosine similarity and KNN,” *Int. J. Eng. Adv. Technol.*, vol. 9, no. 5, pp. 556–559, 2020.

[14] S. Qaiser and R. Ali, “Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents,” *Int. J. Comput. Appl.*, vol. 181, no. 1, 2018, doi: 10.5120/ijca2018917395.

[15] M. R. Pratama, F. A. G. Soerawinata, R. R. Zhafari, and H. N. Irmanda, “Sentiment Analysis of Beauty Product E-Commerce Using Support Vector Machine Method,” *J. RESTI (Rekayasa Sist. Dan Teknol. Informasi)*, vol. 6, no. 2, pp. 269–274, 2022.

[16] Z. Zhao, W. Chen, X. Wu, P. C. Y. Chen, and J. Liu, “LSTM network: A deep learning approach for Short-term traffic forecast,” *IET Intell. Transp. Syst.*, vol. 11, no. 2, 2017, doi: 10.1049/iet-its.2016.0208.

[17] C. C. Aggarwal, *Neural Networks and Deep Learning: A Textbook*. 2018.

[18] R. C. Staudemeyer and E. R. Morris, “Understanding LSTM - a tutorial into Long Short-Term Memory Recurrent Neural NetworksNo Title,” *CoRR*, vol. abs/1909.0, 2019, [Online]. Available: http://arxiv.org/abs/1909.09586.

[19] T. B. Shahi, A. Shrestha, A. Neupane, and W. Guo, “Stock price forecasting with deep learning: A comparative study,” *Mathematics*, vol. 8, no. 9, 2020, doi: 10.3390/math8091441.

[20] A. Nurkholis, D. Alita, and A. Munandar, “Comparison of Kernel Support Vector Machine Multi-Class in PPKM Sentiment Analysis on Twitter,” *J. RESTI (Rekayasa Sist. Dan Teknol. Informasi)*, vol. 6, no. 2, pp. 227–233, 2022.