Aspect-Based Sentiment Analysis of Indonesian-Language Hotel Reviews Using Long Short-Term Memory with an Attention Mechanism

Linggar Maretva Cendani, Retno Kusumaningrum(✉), and Sukmawati Nur Endah

Department of Informatics, Diponegoro University, Semarang, Indonesia
linggarmc@students.undip.ac.id, retno@live.undip.ac.id, sukmane@lecturer.undip.ac.id

Abstract. The development of tourism and technology has given rise to many online hotel booking services that allow users to leave reviews on hotels. Therefore, an analytical model that can comprehensively present the aspects and sentiments in user reviews is required. This study proposes the use of a long short-term memory (LSTM) model with an attention mechanism to perform aspect-based sentiment analysis. The architecture used also implements double fully-connected layers to improve performance. The architecture is used simultaneously for aspect extraction and sentiment polarity detection. Using 5200 Indonesian-language hotel-review data points with labels of five aspects and three sentiments, the model is trained with the configuration of hidden units, dropout, and recurrent dropout parameters in the LSTM layer. The best model performance resulted in a micro-averaged F1-measure value of 0.7628 using a hidden units parameter of 128, dropout parameter of 0.3, and recurrent dropout parameter of 0.3. Results show that the attention mechanism can improve the performance of the LSTM model in performing aspect-based sentiment analysis.

Keywords: Aspect-based sentiment analysis · Long short-term memory · Attention mechanism · Hotel reviews

1 Introduction

Tourism is one of the major supporting sectors of the Indonesian economy, which is rapidly developing. From January to October 2021, the number of foreign tourists coming to Indonesia reached 1.33 million [1]. Developments in the tourism sector are accompanied by developments in the field of technology in Indonesia. Indonesia has the fifth highest number of startups in the world with 2300 startups [2]. The combination of the tourism and technology developments in Indonesia has given rise to many digital startups

M. Ben Ahmed et al. (Eds.): NISS 2022, LNDECT 147, pp. 106–122, 2023.
https://doi.org/10.1007/978-3-031-15191-0_11

that provide online hotel booking services. These services allow users to review hotels by discussing various aspects of the hotel and their sentiments, namely positive, negative, and neutral. As these reviews are large in number, an analytical model is required that can thoroughly and comprehensively present the aspects and sentiments in user reviews, such as aspect-based sentiment analysis (ABSA).

Sentiment analysis, or opinion mining, comprises three levels: document, sentence, and aspect [3]. ABSA is a sentiment analysis model that is conducted at the aspect level, assuming that each document or sentence has various aspects or targets with their respective sentiment classes [4]. Research on the ABSA of Indonesian-language reviews has been conducted previously.

The first study proposed the use of latent Dirichlet allocation, semantic similarity, and long short-term memory (LSTM) for performing ABSA with GloVe word embedding [5]. The F1-score obtained was 85% for aspect detection and 93% for sentiment analysis. A subsequent study used coupled multi-layer attentions and a bi-LSTM architecture with a double-embedding mechanism in FastText to extract aspects and opinions from the Indonesian-language hotel review domain [6]. This study produced the best results with a combination of double embedding and the attention mechanism, with F1-scores of 0.914 and 0.90, respectively, for the extraction of aspects and opinions. Another study proposed the use of bidirectional encoder representations from transformers (BERT) to perform aspect extraction on the Indonesian-language review domain from the TripAdvisor website [7]. This model produced an F1-score of 0.738. By adding the preprocessing stage, the precision value increased to 0.824. Subsequent research proposed the use of the bidirectional GRU model with multi-level attention in the Indonesian-language health service review domain [8]. This resulted in an F1-score of 88% for sentiment classification and 86% for aspect detection.

Another study proposed the use of the convolutional neural network (CNN) model with Word2vec word embedding [9]. The F1-score obtained from this research was 92.02%. However, this study had several shortcomings. It failed to provide a method to determine the target opinion and to deal with the problem of negation in sentiment classification. A later study proposed the use of the BERT or m-BERT multilingual language representation model on Indonesian-language hotel review data and produced an F1-score of 0.9765 [10]. However, misclassification and vocabulary problems persisted because it used multilingual BERT, which is not specific to the Indonesian-language domain. Subsequent research proposed the use of LSTM with the addition of double fully connected layers on top [11]. This study used Word2vec word embedding and produced an F1-score of 75.28%. In [9] and [11], the aspect and sentiment extraction processes were conducted in one process.

Some previous ABSA studies used different methods for sentiment classification and aspect detection [5, 8]. Study [9] can perform sentiment classification and aspect detection simultaneously with the same method by implementing deep learning using CNN architecture. Based on research [12], the LSTM architecture outperforms the CNN architecture in processing text data such as hotel review data on sentiment analysis tasks, because the CNN architecture does not have a memory cell so it can store information about sequence data such as text. Furthermore, study [11] used the LSTM architecture to perform ABSA. The LSTM architecture can remember information in long data sequences using memory cells; thus, it can store the context information contained within a sentence. In addition, LSTM can overcome the vanishing gradient problem that occurs in the vanilla recurrent neural network (RNN) architecture during backpropagation [13].

The LSTM model thoroughly processes text data and assigns the same priority to all words. However, there are only a few words that contain the aspect and sentiment information necessary to obtain the aspect and sentiment output. Because the LSTM model does not prioritize words that contain aspects and sentiments, it sometimes fails to recognize the aspects and sentiments in sentences. This study proposes the use of the LSTM model with the attention mechanism proposed in [14] and [15] to perform ABSA.

In contrast to research [6] which used fasText, this study used Word2vec as word embedding. In research [6], the data used are labeled with annotations, while this study uses data with separate labeling from the review text, so that the text review is in the form of raw data and is not annotated. Labeling in this way is considered faster. In addition, this study uses the attention mechanism proposed by Raffel and Ellis [13], which requires lower computing power than other types of attention mechanisms. This study also applies double fully-connected layers to the proposed architecture, which is proven to improve performance as in the study [11].

One thing that distinguishes this research from other ABSA studies is that most of the other ABSA studies provide different methods or networks for performing aspect/entity extraction and sentiment polarity detection on the architecture, while this study uses the same network that is simultaneously used for aspect classification and sentiment polarity. In addition, in other ABSA studies, most of them still use the SemEval dataset which uses annotations as labeling, this study tries to use a novel dataset with separate labeling with the text review data, so there is no annotation on the text data.

2 Methods

Research was conducted in two main stages: data preparation and model building. Data preparation comprised four stages: data collection, data cleaning, data preprocessing, and Word2vec model training. Model building comprised four stages: data splitting, model training, testing, and evaluation. The entire process of this study is shown in Fig. 1.

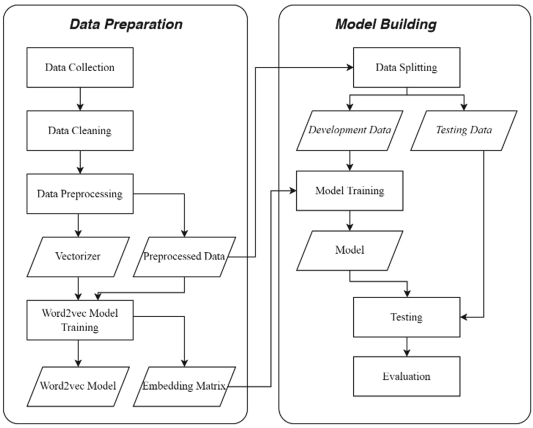

Fig. 1. Research methods

2.1 Data Collection

The data used in this study were hotel user review data obtained from the Traveloka website by crawling the data using the Selenium library. In total, there were 5200 data points comprising 1–5 star hotel reviews. A total of 2500 data points were obtained from studies [12] and [16], and 2700 data points were obtained from study [11]. Data labeling on the dataset had been previously performed. Each data point was labeled in the form of five aspects, namely "makanan" (foods and beverages), "kamar" (rooms), "layanan" (services), "lokasi" (location), and "lain" (others). The label "lain" (others) is a label given to data point that contains aspects but is not included in the first 4 aspects. Each aspect could have a sentiment polarity value in the form of positive, negative, or neutral. The "neutral" sentiment polarity is given to data point that contains opinions but does not include positive or negative opinions, or in other cases the review provides both positive and negative opinions on the same aspect. Because there were five aspects and each aspect had three possible sentiment polarities, 15 labels were used for each review, where each label had a binary representation of 1 or 0. The representation of the label on each review in the form of sentiment polarity for each aspect is shown in Fig. 2, while the data distribution for each aspect are shown in Fig. 3.

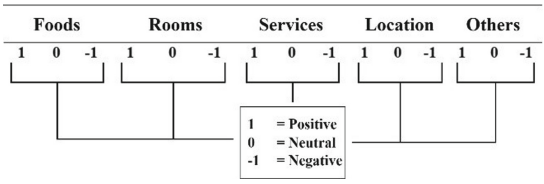

Fig. 2. Representation of the sentiment and aspect polarity labels on reviews

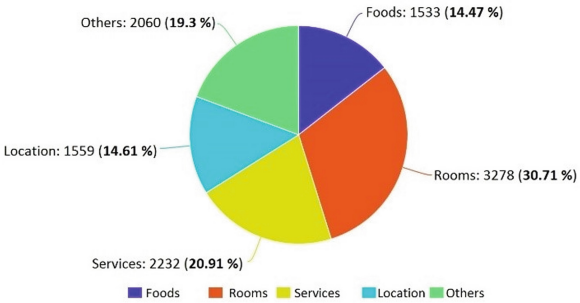

Fig. 3. Data distribution for each aspect

The review data primarily discussed the "rooms" aspect, with a total of 3278 reviews. However, the distribution of the data as a whole was balanced. The data were divided into two parts: 200 testing and 5000 development data points comprising training and validation data.

Examples of hotel review data and their labels can be seen in Table 1. and Table 2., respectively.

Table 1. The examples of hotel review data

No	Review data	Review data (Eng.)
1	Kamar ok tempat strategis apalagi deket dengan stasiun	The room is ok, the location is strategic, especially close to the station
2	Kamar kotor ada banyak rambut belum disapu tissue toilet basah	The room is dirty, there is a lot of hair that hasn't been swept, the toilet tissue is wet
3	Lantai kotor seperti tidak disapu. Kamar mandi bau. Suasana kamarnya agak serem mungkin karena warna catnya gelap jadi kesan pencahayaannya kurang	The floor is dirty like hasn't been swept. The bathroom is smelly. The atmosphere of the room is a bit scary maybe because the paint color is dark so the impression of lighting is lacking

Table 2. The examples of hotel review data labels

No	Foods			Rooms			Services			Location			Others		
	+	=	−	+	=	−	+	=	−	+	=	−	+	=	−
1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0
2	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0
3	0	0	0	0	0	1	0	0	1	0	0	0	0	0	1

In Table 2., the sign "+" means positive sentiment polarity, the sign "=" is neutral sentiment polarity, and the sign "−" means negative sentiment polarity.

2.2 Data Cleaning

The data obtained at the data collection stage were still in the form of noisy data. Data cleaning was performed to improve data quality and eliminate the noise in the data that occurred from missing values and incorrect labels, so that the data could be used in the next process [17].

2.3 Data Preprocessing

Data preprocessing was conducted to prepare the data for use in training and testing [18]. The data preprocessing conducted in this study had six stages.

- Case folding: converting all letters into lowercase, so that the data were uniform [19].
- Filtering: eliminating words, letters, characters, or other parts contained in the text data that appeared often, were considered unimportant, and could affect the performance of the resulting model [20]. The filtering process in this study had three main stages: stopword, punctuation, and number removal. The deletion of these elements was done because they do not have a significant effect on the classification of sentiment.
- Stemming: Changing each word to its basic form. Words in documents typically have affixes in the form of prefixes, suffixes, infexes, and confixes [20]. Stemming was performed by removing affixes.
- Tokenization: Converting document data in the form of a collection of sentences into word parts called tokens [21]. The main process of tokenization was to separate words that were combined in a document into single words that existed alone, so that each document would be in the form of a list of tokens. The tokenization stage was divided into two processes: sentence and word tokenization. The documents in the form of a collection of sentences were first broken down into a list of sentences; thereafter, each sentence comprising a collection of words was broken down into a list of tokens, where each token was one word.
- Padding: equating the size or length of each document so that the length of all data was the same [11]. The padding size used the longest data size. Other data that had a length less than the padding size would be added with a "<pad>" token until the length was equal to the padding size.
- Vectorization: converting text data into numeric data. Vectorization was performed to reduce the amount of memory required, to make it easier for machines to process the data [11]. Vectorization produced two outputs: (a) vectorized review data and (b) a vectorizer, which is a word dictionary with contents in the form of pairs of words and numbers that serve as numerical representations of the vectorized review data.

The output of the data preprocessing stage was the output of the vectorization stage, i.e., preprocessed data and the vectorizer.

2.4 Word2vec Model Training

Word2vec is a word-embedding technique that can represent words in vector form with adjustable length and store the context information of words [22]. The Word2vec model

training parameters utilized in this study were the parameters with the best results in study [12]. Four parameters were used: Word2vec architecture, evaluation method, vector dimensions, and window size. The parameters and their values used in the Word2vec model training are listed in Table 3..

Table 3. Word2vec model training parameters

No	Parameters	Values
1	Word2vec architecture	Skip-gram
2	Evaluation method	Hierarchical Softmax
3	Vector dimensions	300
4	Window size	3

Word2vec model training was conducted unsupervised on preprocessed data and produced two outputs, namely the trained Word2vec model and embedding_matrix, which is a vector dictionary for each word with dimensions of 300.

2.5 Data Splitting

Before training was conducted, the data were first divided via data splitting. Data splitting is a process for dividing data into training, validation, and testing data, so that the data can be used for training and testing. The 5200 data points from the preprocessing stage were divided into two categories: development data, which amounted to 5000 data points, and testing data, which amounted to 200 data points. Development data were further divided into two categories, namely training data with a total of 4500 data points and validation data with a total of 500 data points.

2.6 Model Training

The architecture of the LSTM model with an attention mechanism comprised an input layer, five hidden layers, and an output layer. The input layer accepted input in the form of preprocessed hotel review data. The output layer had 15 neurons with a sigmoid activation function to predict aspects and sentiments. The five hidden layers used in this study were the embedding, LSTM, attention, and double fully connected layers. The proposed LSTM architecture with the attention mechanism is illustrated in Fig. 4.

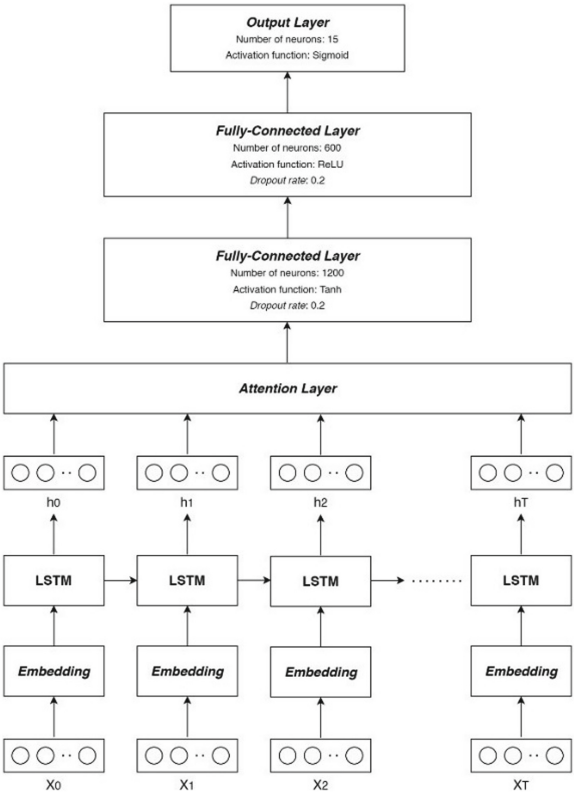

Fig. 4. Proposed architecture

The embedding layer used the embedding matrix generated via Word2vec model training to translate the input data into Word2vec word vectors with a word dimension of 300. The output from this embedding layer was then processed at the LSTM layer, and hidden state values for all timesteps or LSTM cells were generated.

The LSTM model used in this study was proposed by Hochreiter and Schmidhuber [23] with the use of memory cells with three gate units: input, forget, and output. Each LSTM memory cell received three inputs: previous memory (C_{t-1}) or memory from the previous cell, the previous hidden state (h_{t-1}) or hidden state generated from the previous memory cell, and the input vector at the current timestep (x_t). Each memory cell then produced two outputs: the cell state (C_t) or current memory and the hidden state (h_t). The LSTM memory cell architecture is illustrated in Fig. 5.

The hidden states output from the LSTM layer was processed at the attention layer for word weighting by prioritizing words that better represent the sentiments and aspects contained in the sentence.

The attention mechanism for text classification cases proposed by Raffel and Ellis [14] was used in this study. The attention mechanism used is different from the attention mechanism used in Transformers architecture such as BERT which is self-attention [24]. The attention mechanism used in this study requires smaller computing power. However, the attention mechanism used works like other attention mechanisms by prioritizing inputs (in this case words) that have a greater weight to obtain classification outputs, more specifically words that contain information on aspects and their sentiment polarity.

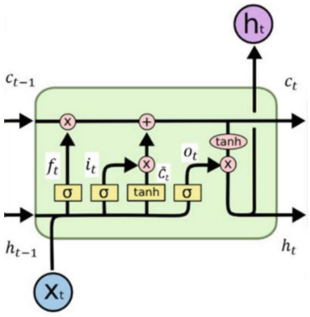

Fig. 5. LSTM memory cell architecture [24]

The attention mechanism used in this study has been tested previously on datasets with long sequences, and can be applied with the LSTM architecture and fully-connected layer. The architecture of the attention mechanism is shown in Fig. 6.

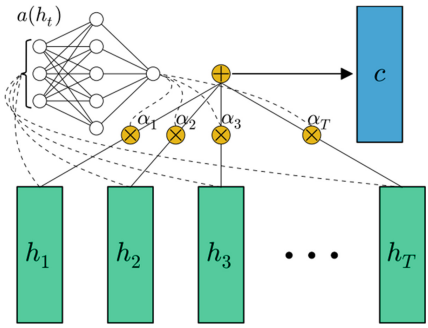

Fig. 6. Attention mechanism architecture [14]

The hidden states of the LSTM layer were first processed to obtain the alignment score (et). The formula for the alignment score (et) is given in Eq. (1).

$$e_t = a(h_t) = tanh(W_a h_t + b_a) \tag{1}$$

The results of the calculation of the alignment score were then entered into the softmax activation function to obtain the final value of the context vector (α_t). The formula is given in Eq. (2).

$$\alpha_t = \frac{\exp(e_t)}{\sum_{k=1}^{T} \exp(e_k)} \tag{2}$$

The result of context vector calculation was then multiplied by the hidden states (h_t) at the corresponding time step. Thereafter, the value of the attention vector (c) was obtained by calculating the weighted average of the multiplication results. The formula for the attention vector is given in Eq. (3).

$$c = \sum_{t=1}^{T} \alpha_t h_t \tag{3}$$

The dimensions of the attention vector were the same as those of the hidden units. This attention vector was then processed on double fully connected layers, referring to research [11], which has been proven to increase the F1-score by 10.16%. The first fully connected layer had 1200 neurons with the TanH activation function, while the second fully connected layer had 600 neurons with the ReLU activation function. The two fully connected layers had a dropout rate of 0.2.

The output from the fully connected layer was then processed by the output layer to perform loss function calculation on the aspect and sentiment label values with a size of 15 binary spaces. The loss function used in this study was binary cross-entropy [25]. The formula for binary cross-entropy is given in Eq. (4).

$$L = -\left(y log\left(\hat{y}\right) + (1 - y) \log\left(1 - \hat{y}\right)\right) \tag{4}$$

The proposed architecture with the combination of LSTM, attention mechanism, and fully-connected layer in this research work in a complementary way. LSTM captures the context and word order in sentences, then the attention mechanism gives word priority to words that contain the most important information in the sentence, then double fully-connected layers are added to improve the performance resulting from the first two layers, as well as classifying aspects and their sentiments, so that accurate output is obtained in the form of aspects and sentiments contained in the input.

2.7 Testing

Testing was conducted to measure the performance of the resulting model. Testing was conducted using the model generated in the model training process and using the testing data from the previous data splitting stage. The results of testing were predictive data in the form of a vector along a length of 15 with decimal numbers according to the number of labels. For comparison with the label on the testing data, the predicted data needed to be thresholded by changing the decimal number to a binary number in the form of 0 or 1. Thresholding was done by rounding numbers greater than or equal to 0.5 up to 1, and rounding numbers less than 0.5 down to 0.

The predicted data can provide more than one sentiment prediction on an aspect. For example, the “foods” aspect can have a sentiment polarity result of “positive” and

"negative" simultaneously. Because one aspect can only have one sentiment polarity, it was necessary to determine the sentiment classification for each polarity combination that could occur. The sentiment classification for all possible prediction results is shown in Table 4..

Table 4. Sentiment classification for each polarity combination

No	Polarity combination			Classification
	Positive	Neutral	Negative	
1	1	0	0	Positive
2	1	0	1	Neutral
3	1	1	0	
4	1	1	1	
5	0	1	1	
6	0	1	0	
7	0	0	1	Negative
8	0	0	0	None

2.8 Evaluation

Evaluation was conducted using the micro-averaged F1-measure method, commonly referred to as the F1-score. The F1-Score metric, specifically the micro-averaged F1-measure is used as a metric in this study because it is very commonly used in multi-label classification tasks, especially in text data. The F1-Score gives combined performance for all classes with the same importance, with more emphasis on the most common labels in the data. Therefore, this metric is preferred for use in multi-label classification tasks. F1-score calculation was performed by calculating the precision and recall values using the confusion matrix table. The confusion matrix was calculated for each aspect because each review could be classified into more than one aspect.

In the micro-averaged F1-measure, the total calculation of TP (true positive), FP (false positive), and FN (false negative) in the confusion matrix table does not consider each class individually; it considers them as a whole [26]. The formulas for precision and recall for the calculation of the micro-averaged F1-measure are given in Eqs. (5) and (6).

$$Precision = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N} TP_i + FP_i} \tag{5}$$

$$Recall = \frac{\sum_{i=1}^{N} TP_i}{\sum_{i=1}^{N} TP_i + FN_i} \tag{6}$$

The formula for the micro-averaged F1-measure or F1-score itself is given in Eq. (7).

$$F1 - score = \frac{2 \times Precision \times Recall}{(Precision + Recall)} \tag{7}$$

3 Results and Discussion

Training was conducted using three combinations of parameters to perform hyperparameter tuning or optimization. The parameters used were hidden units, dropout, and recurrent dropout, all of which were in the LSTM layer. The LSTM layer provided an output that was used as the input for the attention layer, so the effect of changing the three parameters in the LSTM layer would also affect the process of determining the weights in the attention layer, which would in turn affect the performance of the resulting model. The hyperparameters and their values used in the training process are listed in Table 5..

Table 5. Model training hyperparameters

No	Parameters	Values
1	Hidden units	32, 64, 128, 256
2	Dropout	0.2, 0.3, 0.5
3	Recurrent dropout	0.2, 0.3, 0.5

Hyperparameter tuning was conducted to determine the effect of each parameter value on the model performance, as well as to obtain a model with the best combination of parameters.

Scenario 1 aimed to determine how changing the hidden units parameter could affect the performance of the model. Hidden units refer to the number of neurons in each cell in the LSTM layer. The hidden units parameters used in this scenario were 32, 64, 128, and 256. The graph of the effect of the number of hidden units on the F1-score is shown in Fig. 7.

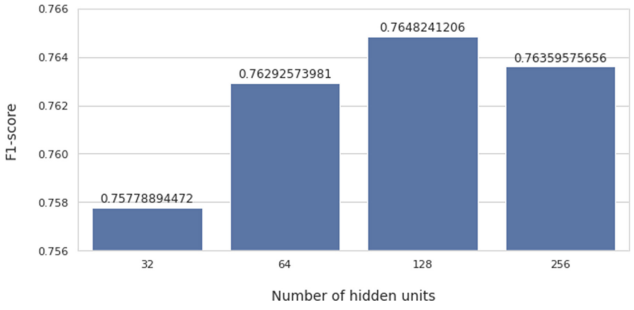

Fig. 7. Effect of hidden units parameters on the F1-score

The graph shows that the larger the number of hidden units used, the better the performance. However, this pattern ceased after 128 hidden units. Thereafter, the F1-score decreased, as observed for 256 hidden units. Based on the graph, the number of hidden units that resulted in the highest F1-score was 128, which produced an F1-score of 0.7648241206 or approximately 76.48%.

Scenario 2 was conducted to identify how changing the dropout parameter would impact the performance of the model. Dropout refers to a technique that randomly removes neurons during training. This technique aims to avoid overfitting to improve the network performance. The dropout parameters used in this scenario were 0.2, 0.3, and 0.5. The graph showing the effect on the F1-score of the dropout used can be seen in Fig. 8.

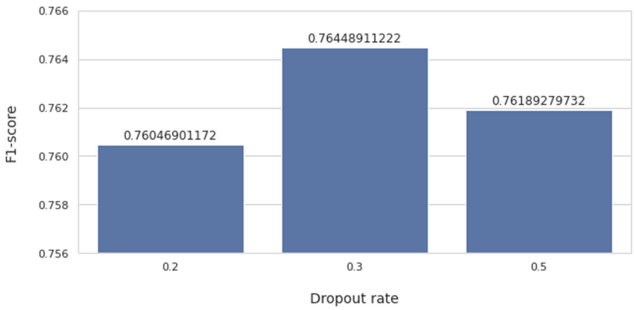

Fig. 8. Effect of dropout parameters on the F1-score

The graph shows that the dropout rate that yielded the best result was 0.3. Dropout with a rate of 0.2 yielded the worst result because it is excessively small, and there was no increase in the generalization process. The dropout with a rate of 0.5 produced slightly better results than that with a rate of 0.2, but still lower than the dropout rate of 0.3. This is because the dropout rate of 0.5 was excessively large, resulting in more neurons being removed, which made training less effective. A dropout of 0.3 was neither excessively large nor small and produced an average F1-score of 0.76448911222 or approximately 76.45%.

Scenario 3 was conducted to identify how changing the recurrent dropout parameter would impact the performance of the model. Recurrent dropout refers to the dropout technique applied to neurons between the recurrent units. The recurrent dropout parameters used in this scenario were 0.2, 0.3, and 0.5. The graph showing the effect of recurrent dropout on the F1-score can be observed in Fig. 9.

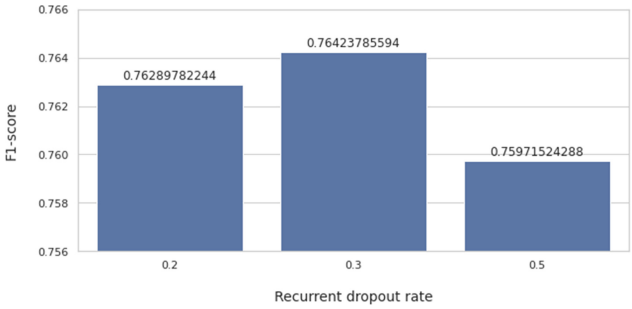

Fig. 9. Effect of recurrent dropout parameters on the F1-score

The graph shows that the rate of the recurrent dropout that yielded the best result was 0.3. Recurrent dropout with a rate of 0.5 produced the worst result because recurrent dropout was applied to the neurons between recurrent units and it can affect the memory for each word or each timestep. The recurrent dropout rate of 0.5 was excessively large, causing a large number of neurons to be removed, which resulted in the weakening of its ability to remember words in the previous timestep. Recurrent dropout with a rate of 0.2 produced a better result, close to that of the recurrent dropout with a rate of 0.3. This can happen because the memory ability with a recurrent dropout of 0.2 is relatively high considering the timestep of the previous words. However, its performance was still worse than that of the recurrent dropout with a rate of 0.3, because the recurrent dropout with a rate of 0.2 tends to not have an increase in the generalization process. Recurrent dropout with a rate of 0.3 produced the best result because the generalization process achieved an adequate increase in the recurrent dropout value. A dropout rate of 0.3 produced an average F1-score of 0.76423785594 or approximately 76.42%.

Scenario 4 was the determination of the model with the best parameters. Based on the experimental results in scenarios 1, 2, and 3, the best parameters obtained were 128 for the number of hidden units, 0.3 for the dropout rate, and 0.3 for the recurrent dropout rate. The model with the best use of these parameters produced an F1-score of 0.7628140703517587 or approximately 76.28%.

The last scenario, scenario 5, was conducted to compare the performance of the resulting model with that of the previous research model that did not use the attention mechanism technique [11]. As explained in the Introduction section, that the LSTM architecture outperforms other architectures such as CNN in processing text data, because LSTM has a memory cell that can process long sequence data such as text data, this study only compares the LSTM architecture using the attention mechanism and without attention mechanism. The results of the comparison can assess the effect of using the attention mechanism on the LSTM architecture to perform ABSA on hotel reviews in the Indonesian language. The results of the comparison between the two models are shown in Fig. 10.

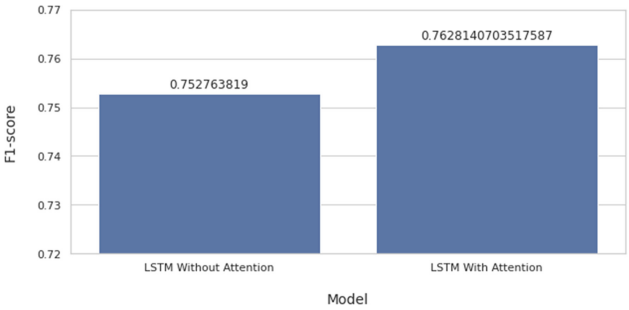

Fig. 10. Model performance comparison with previous study

The graph shows that the proposed architecture produced an F1-score of 0.7628140703517587, outperforming the architecture of the previous study, which produced an F1-score of 0.752763819. The model using the attention mechanism technique is proven to be able to give a higher priority value to words that contain aspects and sentiments. The model generated from the proposed architecture with the best parameters can increase the F1-score with the addition by 0.01005025135 or approximately 1.005%. The resulting increase is not very large, but in the case of ABSA, the increase in the F1-score is significant and is sufficient to provide significant results in real-time implementation.

Despite obtaining adequate results, some misclassifications still occurred. For example, there were several reviews with negative sentiment classes that were predicted as neutral sentiment classes. This happened because there were some sentences that were excessively short and did not refer to the aspects discussed but contained some sentiment polarity. For example, in the sentence "*Bagus, kebersihan kamar kurang,*" there are two words that can represent two different sentiments, namely "*bagus*" for positive sentiments and "*kurang*" for negative sentiments. However, there is only one aspect that is discussed, namely "*kebersihan kamar.*" The word "*bagus*" at the beginning is not clearly intended for a particular aspect, because it seems very general, while the next sentence clearly refers to the aspect of "*kebersihan kamar*" with a negative sentiment polarity. Therefore, the sentence should give a negative sentiment for the room aspect, but the model gives an incorrect classification because there are words that are too general and are not necessarily in reference to a particular aspect.

In addition, there were several positive and negative reviews that were not classified in any sentiment class (None) because the model could not recognize the aspects and sentiments discussed. This can happen if words appear that are not recognized by the model but are actually very representative of the aspects and sentiments discussed, such as the use of foreign languages or languages that are not found in the word dictionary in the Word2vec training process. Because the model does not recognize the word, it cannot detect aspect and sentiment values.

4 Conclusion

From the results of the experiments that have been conducted on the architecture of LSTM with an attention mechanism, it can be concluded that the best model is formed using a hidden units parameter of 128, dropout rate of 0.3, and recurrent dropout rate of 0.3. The best model in this study resulted in an F1-score of 0.7628140703517587 or 76.28%. This value outperforms the models from previous studies that only used the LSTM architecture without an attention mechanism, the best of which produced an F1-score of 0.752764 or approximately 75.28%.

Acknowledgment. The data used in this study were generated at Intelligent Systems Laboratory at the Department of Informatics, Diponegoro University. The data are available on request from the corresponding author, Retno Kusumaningrum. Please Email retno@live.undip.ac.id to requests for access.

References

1. Badan Pusat Statistik: Kunjungan Wisatawan Mancanegara per bulan Menurut Kebangsaan (Kunjungan), Badan Pusat Statistik (2021). https://www.bps.go.id/indicator/16/1470/1/kunjungan-wisatawan-mancanegara-per-bulan-menurut-kebangsaan.html
2. Startup Ranking: Countries - With the top startups worldwide. Startup Ranking (2021). https://www.startupranking.com/countries
3. Liu, B.: Sentiment Analysis and Opinion Mining. Morgan & Claypool Publishers (2012)
4. Ma, Y., Peng, H., Cambria, E.: Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM. In: 32nd AAAI Conference Artificial Intelligence AAAI, vol. 2018, pp. 5876–5883 (2018)
5. Priyantina, R.A., Sarno, R.: Sentiment analysis of hotel reviews using Latent Dirichlet Allocation, semantic similarity and LSTM. Int. J. Intell. Eng. Syst. **12**(4), 142–155 (2019). https://doi.org/10.22266/ijies2019.0831.14
6. Fernando, J., Khodra, M.L., Septiandri, A.A.: Aspect and opinion terms extraction using double embeddings and attention mechanism for Indonesian hotel reviews. In: Proceedings 2019 International Conference Advance Informatics Concepts, Theory, Appl. ICAICTA 2019 (2019). https://doi.org/10.1109/ICAICTA.2019.8904124
7. Yanuar, M.R., Shiramatsu, S.: Aspect extraction for tourist spot review in Indonesian language using BERT. In: 2020 International Conference Artificial Intelligence Information Communication ICAIIC, vol. 2020, pp. 298–302 (2020). https://doi.org/10.1109/ICAIIC48513.2020.9065263
8. Setiawan, E.I., Ferry, F., Santoso, J., Sumpeno, S., Fujisawa, K., Purnomo, M.H.: Bidirectional GRU for targeted aspect-based sentiment analysis based on character-enhanced token-embedding and multi-level attention. Int. J. Intell. Eng. Syst. **13**(5), 392–407 (2020). https://doi.org/10.22266/ijies2020.1031.35
9. Bangsa, M.T.A., Priyanta, S., Suyanto, Y.: Aspect-based sentiment analysis of online marketplace reviews using convolutional neural network. IJCCS (Indonesian J. Comput. Cybern. Syst. **14**(2), 123 (2020). https://doi.org/10.22146/ijccs.51646
10. Azhar, A.N.: Fine-tuning Pretrained Multilingual BERT Model for Indonesian Aspect-based Sentiment Analysis (2020)
11. Jayanto, R., Kusumaningrum, R., Wibowo, A.: Aspect-based sentiment analysis for hotel reviews using an improved model of long short-term memory, unpublished

12. Muhammad, P.F., Kusumaningrum, R., Wibowo, A.: Sentiment analysis using Word2vec and long short-term memory (LSTM) for Indonesian hotel reviews. Procedia Comput. Sci. **179**(2020), 728–735 (2021). https://doi.org/10.1016/j.procs.2021.01.061
13. Smagulova, K., James, A.P.: A survey on LSTM memristive neural network architectures and applications. Eur. Phys. J. Spec. Top. **228**(10), 2313–2324 (2019). https://doi.org/10.1140/epjst/e2019-900046-x
14. Raffel, C., Ellis, D.P.W.: Feed-forward networks with attention can solve some long-term memory problems, pp. 1–6 (2015)
15. Sun, X., Lu, W.: Understanding Attention For Text Classification, no. 1999, pp. 3418–3428 (2020). https://doi.org/10.18653/v1/2020.acl-main.312
16. Nawangsari, R.P., Kusumaningrum, R., Wibowo, A.: Word2vec for Indonesian sentiment analysis towards hotel reviews: an evaluation study. Procedia Comput. Sci. **157**, 360–366 (2019). https://doi.org/10.1016/j.procs.2019.08.178
17. Roh, Y., Heo, G., Whang, S.E.: A survey on data collection for machine learning: a big data-AI integration perspective. IEEE Trans. Knowl. Data Eng. **33**(4), 1328–1347 (2021). https://doi.org/10.1109/TKDE.2019.2946162
18. Mujilahwati, S.: Pre-Processing Text Mining Pada Data Twitter, Semin. Nas. Teknol. Inf. dan Komun., vol. **2016**, no. Sentika, pp. 2089–9815 (2016)
19. Jurafsky, D., Martin, J.H.: Speech and Language Processing: An Introduction to Natural Language Processing,Computational Linguistics, and Speech Recognition. Prentice Hall (2021)
20. Najjichah, H., Syukur, A., Subagyo, H.: Pengaruh Text Preprocessing Dan Kombinasinya Pada Peringkas Dokumen Otomatis Teks Berbahasa Indonesia, J. Teknol. Inf., vol. XV, no. **1**, pp. 1–11 (2019)
21. Rosid, M.A., Fitrani, A.S., Astutik, I.R.I., Mulloh, N.I., Gozali, H.A.: Improving Text Preprocessing for Student Complaint Document Classification Using Sastrawi, IOP Conf. Ser. Mater. Sci. Eng. **874**, 1 (2020). https://doi.org/10.1088/1757-899X/874/1/012017
22. Kurniasari, L., Setyanto, A.: Sentiment analysis using recurrent neural network-lstm in bahasa Indonesia. J. Eng. Sci. Technol. **15**(5), 3242–3256 (2020)
23. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 13–39 (1997). https://doi.org/10.1007/978-1-4757-5388-2_2
24. Zhang, F., Fleyeh, H., Bales, C.: A hybrid model based on bidirectional long short-term memory neural network and Catboost for short-term electricity spot price forecasting, J. Oper. Res. Soc. 1–25 (2020). https://doi.org/10.1080/01605682.2020.1843976
25. Vaswani, A., et al.: Attention is all you need. Adv. Neural Inf. Process. Syst. **2017**(Nips), 5999–6009 (2017)
26. Parmar, R.: Common Loss functions in machine learning, Towards Data Science (2018). https://towardsdatascience.com/common-loss-functions-in-machine-learning-46af0ffc4d23
27. Shmueli, B.: Multi-Class Metrics Made Simple, Part II: the F1-score. Towards Data Science (2019). https://towardsdatascience.com/multi-class-metrics-made-simple-part-ii-the-f1-score-ebe8b2c2ca1