Performance of Lexical Resource and Manual Labeling on Long Short-Term Memory Model for Text Classification

Aqsal Harris Pratama, Mardhiya Hayaty
Universitas Amikom Yogyakarta, Ring Road Utara, Yogyakarta 55283, Indonesia

ARTICLE INFO

Article history:

Received December 12, 2022
Revised January 30, 2023
Published February 08, 2023

Keywords:

Lexicon Resource;
Data Labeling;
Long Short- Term Memory

ABSTRACT

Data labeling is a critical aspect of sentiment analysis that requires assigning labels to text data to reflect the sentiment expressed. Traditional methods of data labeling involve manual annotation by human annotators, which can be both time-consuming and costly when handling large volumes of text data. Automation of the data labeling process can be achieved through the utilization of lexicon resources, which consist of pre-labeled dictionaries or databases of words and phrases in sentiment information. The contribution of this study is an evaluation of the performance of lexicon resources in document labeling. The evaluation aims to provide insight into the accuracy of using lexicon resources and inform future research. In this study, a publicly available dataset was utilized and labeled as negative, neutral, and positive. To generate new labels, a lexicon resource such as VADER, AFINN, SentiWordNet, and Liu & Hu was employed. An LSTM model was then trained using the newly generated labels. The performance of the trained model was evaluated by testing it on data that had been manually labeled. The study found manual labeling led to highest accuracy of 0.79, 0.80, and 0.80 for training, validation, and testing respectively. This is likely due to manual creation of test data labels, enabling the model to learn and capture balanced patterns. Models using lexicon resources (VADER and AFINN) had lower accuracy of 0.54 and 0.56. SentiWordNet had lowest accuracy among all models with 0.49, and Liu&Hu model had the lowest testing score of 0.26. Our research indicates that lexicon resources alone are not sufficient for sentiment data labeling as they are dependent on pre-defined dictionaries and may not fully capture the context of words within a sentence, thus, manual labeling is necessary to complement lexicon-based methods to achieve better result.

Corresponding Author:

Mardhiya Hayaty, Universitas Amikom Yogyakarta, Ring Road Utara, Yogyakarta 55283, Indonesia
Email: mardhiya_hayati@amikom.ac.id

1. INTRODUCTION

Natural Language Processing (NLP) is a computer science sub-discipline to bridge between human and computer language. It helps computers to understand, process, and analyze human language [1]. Sentiment analysis is one of the fields of NLP [2]. The primary purpose of sentiment analysis is to determine the sentiments polarization in the text on a particular topic [3]. One of the major challenges in the field of sentiment analysis is the labor-intensive and costly process of manually labeling large volumes of text data [4].

Several studies have been done regarding the data labeling on sentiment analysis. Study [5] uses VADER Lexicon in document labeling and then applies the Naïve Bayes algorithm for the classification process. The results of the Naive Bayes classification show an accuracy of 0.79. Research [6] conducted a sentiment analysis of the Enron Email dataset through two stages: automatic labeling and creating a classifier model. The automatic labeling process uses VADER and K-Means, while the classification model uses Naive Bayes and Support Vector Machine (SVM). Labeling results using VADER and K-Means indicate that K-Means cannot find associations connecting clusters with data labels. At the same time, VADER can produce labels based on existing data. Furthermore, the Naive Bayes and SVM classification models get an accuracy of 0.58 and 0.82, respectively. While [7] appiles VADER for data labeling and SVM for classification on 168010 unlabeled email data of Swedish Telecom corporation. The LinearSVM algorithm demonstrated the ability to accurately predict the sentiment of emails with an average F1-score of 0.688 and an average AUC of 0.805. Study [8] implements TextBlob in labeling and used it to train several machine learning and deep learning algorithm models on the US airline tweets dataset. The results show that ETC and SVC with BoW - TF-IDF have an accuracy of 0.92, and TextBlob & LSTM - GRU have the highest accuracy of 0.97.

According to [5]–[8] it is known that data labeling can be done using the lexicon resources. To determine the labels in the lexicon resource, an analysis is performed using a sentiment dictionary to evaluate the sentiments of the text data. In this lexicon-based approach, the sentiment assessment is based on a dictionary of words and phrases [9]. Study [10] compares several lexicon resources for sentiment analysis on application review datasets through translation and classification. The translation process is done by converting the dataset from Indonesian to English using Bing and Google. Lexicon resources used in this study include SentiWordNet, SentiStrength, and AFINN. Six experimental scenarios combine each machine translator and Lexicon resource. The six experiments were applied to 533 user reviews in the dataset, and the combination of Google Translate and SentiWordNet engine translators obtained the highest accuracy of 0.72. Comparison of other lexicon resources has also been carried out by [11] using VADER Lexicon, SentiWordNet, SentiStrength, Liu and Hu Lexicon, and AFINN. The Stanford and the Sandars datasets are used, each of which is 498 and 5513 data. The results of applying each Lexicon resource to the two datasets show that the Lexicon VADER can obtain the highest accuracy values of 0.72 and 0.65. Study [12] compares three lexicon resources in sentiment analysis of StockTwits tweets, and the results show that VADER lexicon produces the highest accuracy of 0.94. Another study about comparison of lexicon resources by [13] presents experimentation results comparing the performance of lexicon-based and Sentence-BERT models for sentiment analysis on code-mixed low-resource texts. Code-mixed texts of Bahasa Indonesia and Javanese were translated to English using Google Machine Translation. The Sentiwordnet and VADER lexicons were used for predicting sentiments with the lexicon-based method. Sentence-BERT was used as a classification model on the translated text. The models' performance was measured using accuracy, precision, recall, and F1 score, with the Sentence-BERT model achieving an average accuracy of 0.83, precision of 90, recall of 76%, and F1 score of 83%.

The studies above suggest that lexicon resources can be used to produce labels for text data, which can then be compared to manually labeled data. However, there currently exists a gap in knowledge regarding the performance comparison between models generated by manual labeling and those generated by lexicon resources when tested against manually labeled test data. This highlights the importance of further research in this area to determine the effectiveness of lexicon labeling compared to manual labeling.

This study aims to make two research contributions related to the use of lexicon resources for document labeling. The first contribution is an evaluation of the performance of lexicon resources in this task. Through this evaluation, we aim to gain insight into the accuracy of using lexicon resources for document labeling, and to inform future research on natural language processing and machine learning. The second contribution of this study is the use of manually labeled data as ground truth for evaluating the performance of the LSTM model trained on lexicon-labeled data. This approach allows for a direct comparison of the performance of lexicon labeling to manual labeling, providing a clear understanding of the strengths and limitations of using lexicon resources for document labeling. Overall, this study aims to contribute to the field by providing an analysis of the performance of lexicon resources in document labeling, and to inform future research in this area.

2. METHODS

This study includes several steps: collecting data, data labeling using multiple lexicon resources, preprocessing the data, converting and embedding the text using pretrained word embedding models, modeling using LSTM based on datasets labeled by lexicon resources, and finally, validating and testing the models. The overall process is illustrated in Fig. 1.

2.1. Data Collection

This study uses a dataset from Kaggle: Twitter US Airline Sentiment by CrowdFlower collected in 2017. The dataset contains a total of 14640 data of customer reviews about six American airline service providers, including American, United, US Airways, South-West, Delta and Virgin America Airlines. The dataset has been labeled as negative, neutral, and positive which is considered as manual labeling in this study, and their respective confidence scores ranged from 0 to 1 with number label 9178, 3099, and 2362 for negative, neutral and positive, respectively.

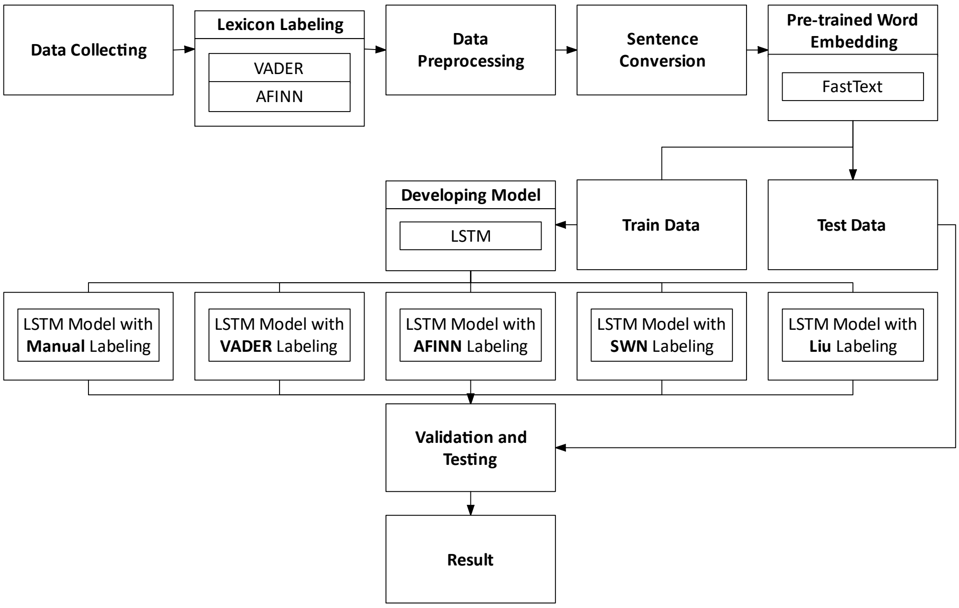
Data Collecting
Lexicon Labeling
VADER
AFINN
Data Preprocessing
Sentence Conversion
Pre-trained Word Embedding
FastText
Developing Model
LSTM
Train Data
Test Data
LSTM Model with Manual Labeling
LSTM Model with VADER Labeling
LSTM Model with AFINN Labeling
LSTM Model with SWN Labeling
LSTM Model with Liu Labeling
Validation and Testing
Result

Fig. 1. Proposed Framework

2.2. Lexicon Labeling

The data labeling uses four lexical resources: VADER, AFINN, SentiWordNet and Hu Liu Lexicon. VADER (Valence Aware Dictionary for Sentiment Reasoning) is a sentiment analysis model that combines lexicon and rule-based [14]. VADER considers punctuation and capitalisation to increase the accuracy of sentiment values in a text. VADER not only classifies categories of negative, neutral, and positive sentiments but also divides the intensity of each category. The VADER framework includes three stages [15] : Preprocessing, which consists of tokenization, stopword removal and punctuation removal; Data boosting is carried out on words such as 'great', 'extremely', and 'very'. If the word is found in the text, then the idiom phrase is checked, followed by the search for the word 'but', and then the comment will be improved; Valence calculation produces a score between -4 to $+4$ [14]. This study uses VADER from the SentimentIntensityAnalyzer package in Python, which has compound values with a range of -1 and 1, to determine the type of sentiment for each row in the dataset. The resulting compound values are then categorized into three types, values $<= -0.05$ are negative, values > -0.05 and < 0.05 are neutral, and values $>= 0.05$ will produce a positive label.

AFINN is a word list in English consisting of words manually rated -5 (negative) to 5 (positive). The latest version of AFINN has 2477 unique words, acronyms, and phrases. Each word in each row in the dataset will be matched with the AFINN dictionary, which is then scored according to the same word to determine the sentiment value. Most words with positive meanings have a score of $+2$, and talks with negative connotation have a score of -2 [16]. The score classification generated by AFINN, namely, a score of 0 is neutral, a score < 0 is negative, and a score > 0 will result in a positive label.

SentiWordNet is a lexical resource based on WordNet Lexicon. These lexical resources are grouped into synsets such as adjectives, nouns, and verbs. Each WordNet synset is categorized into three numerical scores, Objective(s), Positive(s), and Negative(s), which indicate how objective, positive, and negative the terms in a synset are [17]. This research uses SentiWordNet 3.0. SentiWordNet 3.0 uses two main stages [18]: Semi-supervised learning step, which consists of two positive and negative synset 'seeds' which are then automatically expanded where WORDNET binary relations are involved and can retain or reverse their positive and negative properties, then classifier training uses the ternary classifier glosses of sysnset to determine the polarity of the sentiment. The resulting polarity is then classified into Pos, Neg, or Obj. The previous ternary classification process could use different radius parameters and supervised learning; Random-walk step consists of viewing at WORDNET 3.0 as a diagram and performs an iterative "random walk" process in which Pos(s) and Neg(s) (hence Obj(s)) -Values may vary. Each iteration is based on what was determined in the previous step. A random walk step ends when the iterative process converges. The graph used in the unexpected walk step is implicitly defined by the definitions relation of the binomials in WORDNET. In this study, the resulting polarity value is converted into a label with a value of $<= -0.05$, which is negative, a value of > -0.05, and < 0.05 is neutral, and a value of $>= 0.05$ will produce a positive label.

Hu & Liu is a dictionary-based lexical resource divided into two parts, namely, positive and negative. Hu & Liu Dictionary has 6783 words consisting of 4783 negative and 2006 positive words [19]. The process of determining the sentiment value is by adding and subtracting the words in the dataset using a dictionary. Each word found in a positive dictionary has added a value of 1, and a word in a negative dictionary is reduced by a value of 1. The result of the addition of these words is used to determine the label, if will result in neutral, $>= 1$ will result in positive, and $<= -1$ is negative

2.3. Data Preprocessing

Preprocessing ensures that the data is clean and ready to be processed at the next stage [20]. In this study, several steps will be taken to prepare the text for analysis. These include converting all capital letters to lowercase, filtering out punctuation marks and non-alphabetic characters, as well as removing any numbers. Additionally, a process known as stopword removal will be applied, which involves eliminating words that are not deemed necessary for the analysis [21]. An example of the text preprocessing is shown in Table 1.

Table 1. Data Preprocessing

Process	Text Result
Text Example	It's really aggressive to blast obnoxious "entertainment" in your guests' faces &amp; they have little recourse
Casefolding	it's really aggressive to blast obnoxious "entertainment" in your guests' faces &amp; they have little recourse
Filtering	its really aggressive to blast obnoxious entertainment in your guests faces amp they have little recourse
Stopword Removal	really aggressive blast obnoious entertainment guests face little recourse

2.4. Sentence Conversion

Word in each row in the dataset will be converted into numbers. This process will convert the words into tokens and then convert the tokens into a set of arrays containing numbers [22]. The following process is adding padding to each sentence with the goal of the sentence length in the array being the same [23]. In this study, the padding is given according to the maximum length of words in sentences in the dataset, which is 19. Table 2 illustrates the results of sentence conversion, the Word Tokens step separates each word using quotation marks and commas into an array, then the sequences generate a number that represents the words in the token, and finally, Padding adds zero (0) at the beginning of the token so that its length becomes 19.

Table 2. Sentence Conversion

Process	Text Result
Word Tokens	['really', 'aggressive', 'blast', 'obnoxious', 'entertainment', 'guests', 'faces', 'little', 'recourse']
Sequences	[130, 3694, 4571, 1001, 4199, 3695, 54, 498, 2739]
Padding	array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 130, 3694, 4571, 1001, 4199, 3695, 54, 498, 2739], dtype=int32)

2.5. Pre-trained Word Embedding

This process will convert the word into a vector representing the changed word. The resulting vector has dimensions that can be determined. The longer the vector, the more accurate the representation of the result [24]. The maximum number of words in each row of the dataset is 19, and the dimension length used is 300. This study uses pre-trained word embedding FastText [25]. FastText is commonly used to tackle sentence classification and word representation tasks in a more efficient and faster manner than Word2vec and GloVe. [26]. After going through this stage, 10092 words were found in the wiki vocab and 5247 new words were found in the dataset. Table 3 shows words in the form of vectors after going through the pre-trained word embedding process.

Table 3. Pretrained Word Embedding Word Example

Word	Dimension				
	1	2	3	…	300
flight	0.10809999	0.13380000	0.060699993	…	0.11479999
delayed	0.00350001	-0.00779997	-0.032000012	…	-0.03759999
airport	0.20029999	0.17340000	0.062600000	…	0.16750000
weather	-0.0065999	-0.03620001	-0.239199999	…	0.16899999
canseled	0.22789999	-0.06610001	-0.047600001	…	0.20640007

2.6. Developing Model

Five models will be created using manual labeling and labels generated by four lexicon resources. This study uses the LSTM (Long Short Term Memory) algorithm in modelling. Long short-term memory (LSTM) has recently gained popularity among NLP researchers for their superior ability to model and learn from sequential data [27]. LSTM has been proven to be effective in addressing various challenges in natural language processing tasks, such as sentence classification [28], various tagging problems [29][30], and sequence-to-sequence predictions [31]. LSTM is a type of RNN that is capable of remembering and utilizing previous pattern information. Unlike traditional neural networks, which only take into account the current input data, LSTM models are able to incorporate previous input data and context, allowing them to better capture the long-term dependencies and sequential nature of natural language data [32].

LSTM has a hidden layer to store and update previous information. The hidden layer consists of 3 gates or gates, namely forget gate, input gate and output gate [33]. Forget gate is a gate used to decide which information is deleted from the cell state. The sigmoid layer takes the decision. The input gate will determine the new information stored in the cell state by selecting the part to update and the context candidate value. The sigmoid layer will decide which part of the cell to output [34]. The calculation of the value of each gate on the LSTM is shown in the equation below [35].

$$i_t = \sigma(U_i t_{t-1} + W_i x_t + b_i) \tag{1}$$

$$f_t = \sigma(U_f t_{t-1} + W_f x_t + b_f) \tag{2}$$

$$o_t = \sigma(U_o t_{t-1} + W_o x_t + b_o) \tag{3}$$

$$c_t = f_t \times c_{t-1} + i_t \times \tanh(U_c h_{t-1} + W_c x_t + b_c) \tag{4}$$

$$h_t = o_t \times \tanh\,(c_t) \tag{5}$$

Where (1)-(3), n-d vectors i_t, f_t, and o_t shows input gate, forget gate, and output gate, respectively at time t. Equations (4)-(5) show n-d cell state, c_t, hidden unit h_t at time t.

The model architecture in this study uses a triple-layer LSTM adapted from [36] with a slight change in the shape of several layers shown in the Fig. 2.

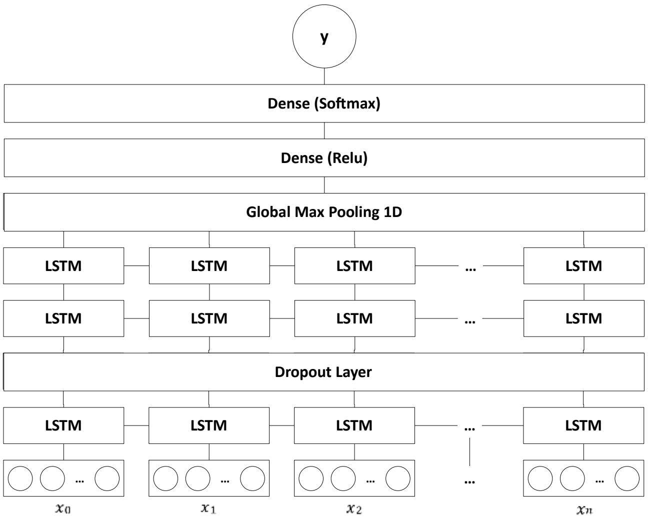

Fig. 2. Model Architecture

Fig. 2 shows the classification model architecture used in this research, starting with an embedding matrix containing vectors generated by the FastText word embedding. It is then followed by LSTM layers, dropout layers, two LSTM layers, GlobalMaxPooling 1D, and finally closed with a softmax activation function on the dense layer.

Table 4 shows the details of the LSTM architecture layer to train the model with each LSTM layer using 64 units followed by a dropout layer, two LSTM layers, a global max pooling layer, and a dense 'relu' layer and ending with three softmax layers, additonaly uses a hyperparameter learning rate of 0.001 along with the Adam optimizer.

Table 4. LSTM Architecture

Layer	Output Shape	Number of Parameters
Embedding	(None, 19, 300)	4602000
LSTM Layer + Dropout 0.5	(None, 19, 8)	9888
LSTM Layer + Dropout 0.5	(None, 19, 8)	544
LSTM Layer + Dropout 0.5	(None, 19, 8)	544
Max Pooling	(None, 8)	0
Dense Layer	(None, 16)	144
Dropout Layer	(None, 16)	0
Dense Layer	(None, 3)	51

2.7. Validation and Testing

The testing model uses 0.2 of the total data with 2928 data. The validation process during training by each model uses a label generated by each lexicon resource of 0.1. In contrast, testing for all models will be carried out on test data that has been manually labeled.

3. RESULTS AND DISCUSSION

This study aims to compare the performance of lexical labeling aginst manual labeling in the LSTM model, which is then tested on test data that has been manually labeled.

3.1. Data Labeling

Each lexicon resource able to determine the sentiment value of the entire dataset with the output in the form of positive, neutral, and negative labels. Table 5 shows the number of labels generated by each lexicon resource

Table 5. Labeling Result

Method	Negative	Neutral	Positive
Manual	9178	3099	2363
VADER	5175	3302	6163
AFINN	4973	4074	5593
SentiWordNet	5210	4263	5167
Hu & Liu	3355	7211	4074

Table 5 shows that each Lexicon resource has its pattern in determining the polarity of sentiment from the existing data. VADER lexicon identified a relatively balanced number of negative, neutral, and positive sentiment occurrences. This suggests that it may be well-suited for datasets where all three sentiment types are present in roughly equal proportions. AFINN and SentiWordNet lexicon resources identified more occurrences of negative sentiment than neutral or positive sentiment. This suggests that they may be more suited for datasets where negative sentiment is more prevalent. The Hu & Liu lexicon identified a large number of neutral sentiment occurrences and less occurrences of negative and positive sentiment.

The labeling results show a rather far difference between manual and lexicon. Labeling by humans pays attention to the context and emotions of a sentence. At the same time, the lexicon resource is fixed on the existing dictionary so that if there is an out of vocabulary, it will produce a value of zero; besides that, the number of vocabulary and the weight of each word affect the results of the label. For example, in the sentence 'guys need serious training customer service many better options put way guys handle ur mistakes' the calculations and labels are shown in Table 6.

Table 6 illustrates the method used to calculate scores for each lexicon in the sentence "guys need serious training customer service many better options put way guys handle our mistakes." The data reveals that the words "better" and "mistakes" have opposing values which ultimately results in a neutral label. However, the SentiWordNet (SWN) lexicon assigns positive labels to several words, despite having low scores, while other words receive low or even zero scores. In contrast, human evaluation takes into account the entire sentence and its context. In this particular case, the overall sentiment conveyed by the sentence is one of complaint, which would be classified as negative.

The example sentence demonstrates that the lexical source used for sentiment analysis assigns labels based on the score calculated for each individual word, rather than taking into consideration the context of the sentence as a whole. This can lead to a discrepancy between the label assigned by the lexical source and the label assigned through manual evaluation, as seen in Table 5. This difference in labeling can have an impact on the performance of the model during training.

Table 6. Lexicon Resource Score Calculation Example

Word	VADER	AFINN	SWN	Liu Hu
guys	0.0	0.0	0.0	0
need	0.0	0.0	0.0	0
serious	-0.0772	0.0	0.125	0
training	0.0	0.0	0.125	0
customer	0.0	0.0	0.0	0
service	0.0	0.0	0.0	0
many	0.0	0.0	0.0	0
better	0.4404	2.0	0.875	1
options	0.0	0.0	-0.250	0
put	0.0	0.0	0.0	0
way	0.0	0.0	0.0	0
guys	0.0	0.0	0.0	0
handle	0.0	0.0	0.0	0
ur	0.0	0.0	0.0	0
mistakes	-0.3612	-2.0	-0.625	-1
Final Score	0.0258	0.0	0.125	0
Label	Neutral	Neutral	Positive	Neutral

3.2. Model Development

The development model is based on the architecture in Fig. 2 with the number of units in Table 4. The training process for all LSTM models uses the same architecture using 200 epochs. The model trained with manual labeling achieved the best results in terms of training, validation, and testing, with scores of 0.79, 0.80, and 0.80 respectively, compared to models trained using lexicon resources. This is likely due to the fact that the test data labels were also created manually, allowing the model to learn and capture the most balanced patterns compared to models trained on lexicon-based labels. As a result, the manually-labeled model was able to perform better on the test data, achieving a higher accuracy score.

The VADER model performed well in terms of training and validation accuracy, but did not achieve a high testing accuracy. The VADER model's scores were 0.78, 0.82, and 0.54 for training, validation, and testing, respectively. This suggests that the VADER model is unable to produce the same labeling pattern as manual labeling, resulting in a lower testing accuracy compared to the manually-labeled model.

The AFINN model achieved training, validation, and testing accuracy scores of 0.79, 0.85, and 0.56, respectively. Although this is the highest testing accuracy among the lexicon resources, the AFINN labeling pattern still falls short of manual labeling.

Meanwhile, the SentiWordNet model had the lowest training and validation accuracy among the models, and its testing score was the second-lowest with a score of 0.68, 0.71, and 0.49 for training, validation, and testing, respectively.

The Hu & Liu model had the highest average training and validation accuracy of 0.79 and 0.85 compared to the other lexicon resources, but had the lowest testing accuracy of 0.26. This is likely due to the fact that Hu & Liu's simple dictionary-based labeling model needs to be regularly updated to adapt to new vocabulary.

Overall, these results suggest that while some lexicon resources may perform well on certain metrics, they are not able to consistently outperform manual labeling in terms of overall performance and accuracy. Based on the five existing models, the model trained with manual labeling has the highest level of testing accuracy compared to other models, amounting to 0.80. The model trained by the lexicon resource label can not replace the manual labeling method.

The closest research of this study are [5]–[8]; they use lexicon resources for labelling data and training the models using its label on machine learning or deep learning algorithm. The initial data they use is unlabeled, so the performance of the lexicon resource label compared to manual labeling is unknown. Although that may be the case, the research indicates that data trained and tested using labels generated by that lexicon resource itself have quite high results. In study [5], the use of VADER Lexicon for document labeling and the Naïve Bayes algorithm for classification resulted in an accuracy of 0.79. In research [6], automatic labeling using VADER and K-Means was performed, and the Naive Bayes and SVM algorithms were used for classification, resulting in an accuracy of 0.58 and 0.82 respectively.. Study [7] applied VADER for data labeling and SVM for classification resulting in an average F1-score of 0.688 and an average AUC of 0.805. In study [9], TextBlob was used for labeling and several machine learning and deep learning algorithm models were trained on the

US airline tweets dataset, resulting in an accuracy of 0.92 for ETC and SVC with BoW - TF-IDF and the highest accuracy of 0.97 for TextBlob & LSTM - GRU.

. Fig. 3 shows each model's training and validation history where training accuracy and loss are represented by blue lines, while yellow lines represent validation accuracy and loss. As we can see, the model trained using manual labeling has the best training and loss results, followed by SWN and VADER, while overfitting occurs on AFINN and Liu & Hu labeling.

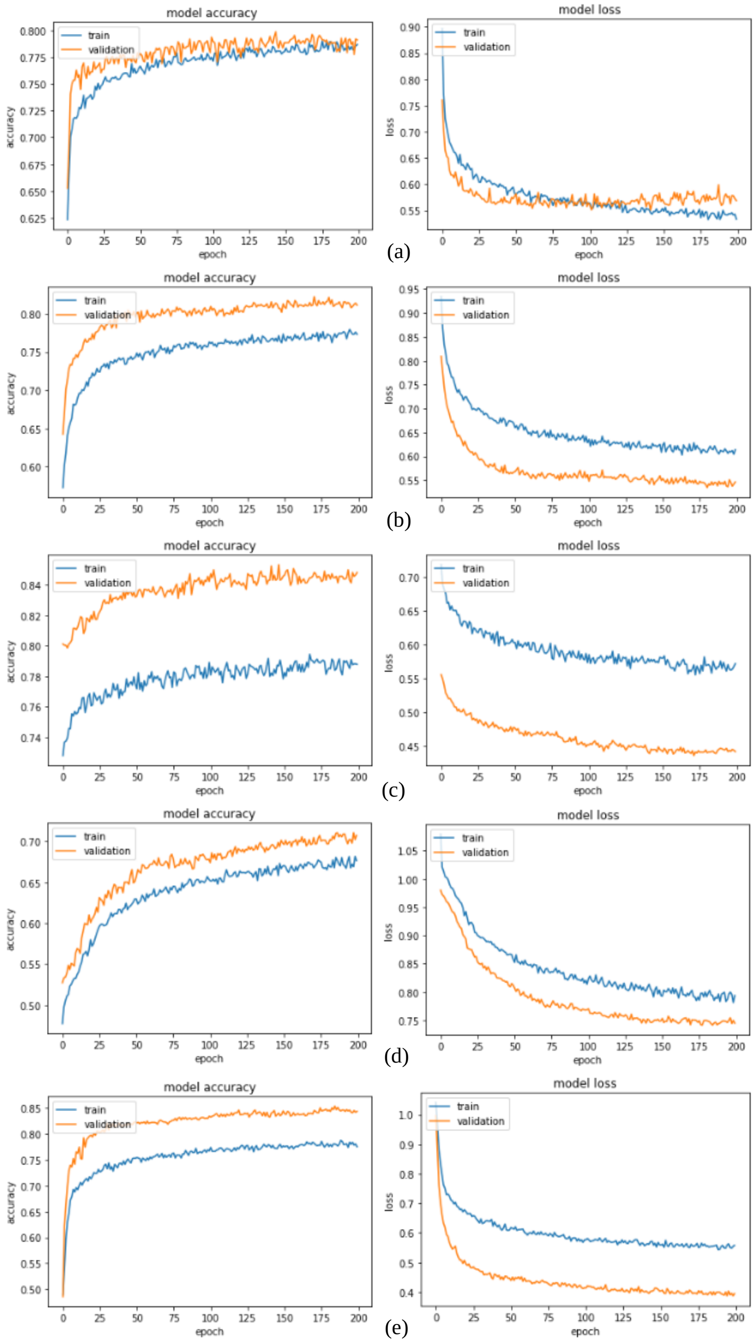

Fig. 3. Training accuracy and training loss of LSTM model using: (a) manual labeling; (b) VADER; (c) AFINN; (d) SentiWordNet; (e) Hu & Liu

Table 7 shows the results of training accuracy, validation accuracy, and testing accuracy on test data with manual labeling. Based on the table, the AFINN model has the highest training accuracy and validation accuracy scores among other lexical resources, followed by VADER, SWN, and Hu & Liu respectively; the highest testing accuracy is achieved by manual labeling.

Table 7. Model Result

Model	Training Accuracy	Validation Accuracy	Testing Accuracy
Manual	0.79	0.80	0.80
VADER	0.78	0.82	0.54
AFINN	0.79	0.85	0.56
SentiWordNet	0.68	0.71	0.49
Hu Liu	0.79	0.85	0.26

This study uses a different approach where the author compares how close the lexicon resource can achieve the same pattern with manual labelling. Using a lexicon resource to train and test a model can produce high results, as the model is able to learn and predict labels that are generated by the lexicon resource itself. However, when the model is tested against manually labeled ground truth, the results may not be as accurate. This is because lexicon resources are heavily dependent on their own lexicon dictionaries, and may not perform as well when applied to labels that are not included in the lexicon resource. This is demonstrated in Table 7, which shows that the model's performance may not be as reliable when tested against manually labeled ground truth.

Overall, the results of the study suggest that while some lexicon resources, may perform well on certain metrics such as training and validation accuracy, they are not able to consistently outperform a model trained using manual labeling in terms of overall performance and accuracy on test data. This indicates that manual labeling may be more effective in capturing the nuances and complexities of sentiment in language, leading to more accurate sentiment analysis models.

On the other hand, the model trained by manually labeled data in this study can still be improved by properly handle unbalanced data [37]. In the same dataset, study [38] used the LSTM model with a train-to-test data ratio of 3:1 without addressing the imbalance in the data and without using word embedding. As a result, they only achieved an accuracy of 0.69. In addition, the model's accuracy with built-in labels introduced using this dataset was below [39] and [40], which achieved testing accuracy of 0.90 and 0.92, respectively. Study [39] aims to design optimal LSTM topologies using the clonal selection algorithm (CSA) to hyper-tune parameters until achieving the best topology. While [40] used a more complex model, which they referred to as an ensemble model combining Robustly optimised Bidirectional Encoder Representations from Transformers approach (RoBERTa), Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (BiLSTM) and Gated Recurrent Units (GRU).

4. CONCLUSION

This study compares the performance of LSTM models trained using manual labeling and lexicon resources such as VADER, AFINN, SentiWordNet, and Hu & Liu on the US Airline Sentiment by CrowdFlower Twitter dataset, which contains 14640 data. The results of labeling using these lexicon resources differ from manual labeling, as each lexicon resource has its own method for determining the sentiment score of each word or phrase. The weight and diversity of vocabulary also impact the score and ultimately the label assigned to each word. Our results show that the model trained with manual labeling performs better than models trained using lexicon resources, with a testing accuracy value of 0.80. The best-performing lexicon resource, AFINN, only achieved a testing accuracy value of 0.56. Based on these findings, we conclude that using lexicon resources for labeling sentiment data cannot fully replace manual labeling, as these resources are limited by their reliance on a dictionary and are unable to understand the context of individual words within a sentence.

In future studies, it would be beneficial to investigate utilizing models or resources that possess the ability to comprehend the context and meaning within a sentence to create a more efficient and effective method for automatic labeling. This method has the potential to decrease the time and expenses associated with the data labeling phase, if it proves to be successful. By utilizing models or resources that comprehend the context of a sentence, it may be possible to enhance the precision and dependability of the automatic labeling process, ultimately resulting in enhanced performance of the overall sentiment analysis system.

REFERENCES

[1] D. Khurana, A. Koli, K. Khatter, and S. Singh, “Natural language processing: state of the art, current trends and challenges,” *Multimed. Tools Appl.*, pp. 1–32, 2022, https://doi.org/10.1007/S11042-022-13428-4.

[2] R. Socher *et al.*, “Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank,” in

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Oct. 2013, pp. 1631–1642, [Online]. Available: https://aclanthology.org/D13-1170.

[3] A. V. Kotelnikova, "Comparison of Deep Learning and Rule-based Method for the Sentiment Analysis Task," *2020 Int. Multi-Conference Ind. Eng. Mod. Technol. FarEastCon 2020*, pp. 1-6, 2020, https://doi.org/10.1109/FAREASTCON50210.2020.9271333.

[4] T. Fredriksson, D. I. Mattos, J. Bosch, and H. H. Olsson, "An empirical evaluation of algorithms for data labeling," *Proc. - 2021 IEEE 45th Annu. Comput. Software, Appl. Conf. COMPSAC 2021*, pp. 201–209, 2021, https://doi.org/10.1109/COMPSAC51774.2021.00038.

[5] C. V D, "Hybrid approach: naive bayes and sentiment VADER for analyzing sentiment of mobile unboxing video comments," *Int. J. Electr. Comput. Eng.*, vol. 9, no. 5, pp. 4452–4459, 2019, https://doi.org/10.11591/ijece.v9i5.pp4452-4459.

[6] R. S. Hag Ali and N. El Gayar, "Sentiment Analysis using Unlabeled Email data," *Proc. 2019 Int. Conf. Comput. Intell. Knowl. Econ. ICCIKE 2019*, pp. 328–333, 2019, https://doi.org/10.1109/ICCIKE47802.2019.9004372.

[7] A. Borg and M. Boldt, "Using VADER sentiment and SVM for predicting customer response sentiment," *Expert Syst. Appl.*, vol. 162, p. 113746, 2020, https://doi.org/10.1016/J.ESWA.2020.113746.

[8] W. Aljedaani *et al.*, "Sentiment analysis on Twitter data integrating TextBlob and deep learning models: The case of US airline industry," *Knowledge-Based Syst.*, vol. 255, p. 109780, 2022, https://doi.org/10.1016/J.KNOSYS.2022.109780.

[9] T. Wilson, J. Wiebe, and P. Hoffmann, "Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis," in *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pp. 347–354, 2005, https://doi.org/10.3115/1220575.1220619.

[10] B. T. Pratama, E. Utami, and A. Sunyoto, "A comparison of the use of several different resources on lexicon based Indonesian sentiment analysis on app review dataset," *Proceeding - 2019 Int. Conf. Artif. Intell. Inf. Technol. ICAIIT 2019*, pp. 282–287, 2019, https://doi.org/10.1109/ICAIIT.2019.8834531.

[11] M. A. Al-Shabi, "Evaluating the performance of the most important Lexicons used to Sentiment analysis and opinions Mining," *IJCSNS Int. J. Comput. Sci. Netw. Secur.*, vol. 20, no. 1, pp. 51–617, Accessed: Feb. 09, 2022. [Online]. Available: http://paper.ijcsns.org/07_book/202001/20200107.pdf.

[12] S. Sohangir, N. Petty, and Di. Wang, "Financial Sentiment Lexicon Analysis," *Proc. - 12th IEEE Int. Conf. Semant. Comput. ICSC 2018*, pp. 286–289, 2018, https://doi.org/10.1109/ICSC.2018.00052.

[13] C. Tho, Y. Heryadi, I. H. Kartowisastro, and W. Budiharto, "A Comparison of Lexicon-based and Transformer-based Sentiment Analysis on Code-mixed of Low-Resource Languages," *Proc. 2021 1st Int. Conf. Comput. Sci. Artif. Intell. ICCSAI 2021*, pp. 81–85, 2021, https://doi.org/10.1109/ICCSAI53272.2021.9609781.

[14] C. Hutto and E. Gilbert, "VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text," *Proc. Int. AAAI Conf. Web Soc. Media*, vol. 8, no. 1, pp. 216–225, 2014, [Online]. Available: https://ojs.aaai.org/index.php/ICWSM/article/view/14550.

[15] C. Dev, A. Ganguly, and H. Borkakoty, "Assamese VADER: A Sentiment Analysis Approach Using Modified VADER," *2021 Int. Conf. Intell. Technol. CONIT 2021*, pp. 1-5, 2021, https://doi.org/10.1109/CONIT51480.2021.9498455.

[16] F. Å. Nielsen, "A new ANEW: Evaluation of a word list for sentiment analysis in microblogs," *CEUR Workshop Proc.*, vol. 718, pp. 93–98, 2011, https://doi.org/10.48550/arxiv.1103.2903.

[17] A. Esuli and F. Sebastiani, "SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining," in *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, pp. 417–422, 2006, [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2006/pdf/384_pdf.pdf.

[18] S. Baccianella, A. Esuli, and F. Sebastiani, "SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining.," in *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, vol. 10, 2010, [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2010/pdf/769_Paper.pdf

[19] M. Hu and B. Liu, "Mining and summarizing customer reviews," *Proc. Tenth ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 168–177, 2004, https://doi.org/10.1145/1014052.1014073.

[20] E. Haddi, X. Liu, and Y. Shi, "The role of text pre-processing in sentiment analysis," *Procedia Comput. Sci.*, vol. 17, pp. 26–32, 2013, https://doi.org/10.1016/J.PROCS.2013.05.005.

[21] A. I. Saad, "Opinion Mining on US Airline Twitter Data Using Machine Learning Techniques," *16th Int. Comput. Eng. Conf. ICENCO 2020*, pp. 59–63, 2020, https://doi.org/10.1109/ICENCO49778.2020.9357390.

[22] K. K. Agustiningsih, E. Utami, and O. M. A. Alsyaibani, "Sentiment Analysis and Topic Modelling of The COVID-19 Vaccine in Indonesia on Twitter Social Media Using Word Embedding," *J. Ilm. Tek. Elektro Komput. dan Inform.*, vol. 8, no. 1, pp. 64–75, 2022, https://doi.org/10.26555/jiteki.v8i1.23009.

[23] M. A. Nurrohmat and A. SN, "Sentiment Analysis of Novel Review Using Long Short-Term Memory Method," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 13, no. 3, pp. 209–218, 2019, https://doi.org/10.22146/IJCCS.41236.

[24] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *1st Int. Conf. Learn. Represent. ICLR 2013 - Work. Track Proc.*, 2013, https://doi.org/10.48550/arxiv.1301.3781.

[25] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin, "Advances in Pre-Training Distributed Word Representations," *Lr. 2018 - 11th Int. Conf. Lang. Resour. Eval.*, pp. 52–55, 2017, https://doi.org/10.48550/arxiv.1712.09405.

[26] A. G. D'Sa, I. Illina, and D. Fohr, "BERT and fastText Embeddings for Automatic Detection of Toxic Speech," *Proc. 2020 Int. Multi-Conference Organ. Knowl. Adv. Technol. OCTA 2020*, pp. 1-5, 2020, https://doi.org/10.1109/OCTA49274.2020.9151853.

[27] W. K. Sari, D. P. Rini, and R. F. Malik, "Text Classification Using Long Short-Term Memory With GloVe Features," *J. Ilm. Tek. Elektro Komput. dan Inform.*, vol. 5, no. 2, pp. 85–100, 2019, https://doi.org/10.26555/jiteki.v5i2.15021.

[28] J.-H. Wang, T.-W. Liu, X. Luo, and L. Wang, "An LSTM Approach to Short Text Sentiment Classification with Word Embeddings," in *Proceedings of the 30th Conference on Computational Linguistics and Speech Processing ({ROCLING} 2018)*, pp. 214–223, 2018, [Online]. Available: https://aclanthology.org/O18-1021.

[29] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF Models for Sequence Tagging," 2015, https://doi.org/10.48550/arxiv.1508.01991.

[30] P. Wang, Y. Qian, F. K. Soong, L. He, and H. Zhao, "Part-of-Speech Tagging with Bidirectional Long Short-Term Memory Recurrent Neural Network," 2015, https://doi.org/10.48550/arxiv.1510.06168.

[31] I. Sutskever, O. Vinyals, and Q. V Le, "Sequence to Sequence Learning with Neural Networks," in *Proceedings of the 27th International Conference on Neural Information Processing Systems*, vol. 2, pp. 3104–3112, 2014, https://doi.org/10.48550/arXiv.1409.3215.

[32] P. F. Muhammad, R. Kusumaningrum, and A. Wibowo, "Sentiment Analysis Using Word2vec And Long Short-Term Memory (LSTM) For Indonesian Hotel Reviews," *Procedia Comput. Sci.*, vol. 179, pp. 728–735, 2021, https://doi.org/10.1016/J.PROCS.2021.01.061.

[33] M. S. Islam, M. S. Sultana, M. U. Kumar, J. Al Mahmud, and S. J. Islam, "HARC-New Hybrid Method with Hierarchical Attention Based Bidirectional Recurrent Neural Network with Dilated Convolutional Neural Network to Recognize Multilabel Emotions from Text," *J. Ilm. Tek. Elektro Komput. dan Inform.*, vol. 7, no. 1, pp. 142–153, 2021, https://doi.org/10.26555/jiteki.v7i1.20550.

[34] A. Kumar and R. Rastogi nee Khemchandani, "Attentional recurrent neural networks for sentence classification," *Adv. Intell. Syst. Comput.*, vol. 757, pp. 549–559, 2019, https://doi.org/10.1007/978-981-13-1966-2_49.

[35] Y. Lu and F. M. Salem, "Simplified Gating in Long Short-term Memory (LSTM) Recurrent Neural Networks," *Midwest Symp. Circuits Syst.*, pp. 1601–1604, 2017, https://doi.org/10.48550/arxiv.1701.03441.

[36] A. Setyanto *et al.*, "Arabic Language Opinion Mining Based on Long Short-Term Memory (LSTM)," *Appl. Sci. 2022*, vol. 12, no. 9, p. 4140, 2022, https://doi.org/10.3390/APP12094140.

[37] S. Adi, A. Hikmah, B. W. Sari, A. Sunyoto, A. Yaqin, and M. Hayaty, "The Best Techniques to Deal with Unbalanced Sequential Text Data in Deep Learning," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 11, 2022, https://doi.org/10.14569/IJACSA.2022.0131177.

[38] F. Rustam, I. Ashraf, A. Mehmood, S. Ullah, and G. S. Choi, "Tweets Classification on the Base of Sentiments for US Airline Companies," *Entropy*, vol. 21, no. 11, 2019, https://doi.org/10.3390/e21111078.

[39] A. Al Bataineh and D. Kaur, "Immunocomputing-Based Approach for Optimizing the Topologies of LSTM Networks," *IEEE Access*, vol. 9, pp. 78993–79004, 2021, https://doi.org/10.1109/ACCESS.2021.3084131.

[40] K. L. Tan, C. P. Lee, K. M. Lim, and K. S. M. Anbananthen, "Sentiment Analysis With Ensemble Hybrid Deep Learning Model," *IEEE Access*, vol. 10, pp. 103694–103704, 2022, https://doi.org/10.1109/ACCESS.2022.3210182.

BIOGRAPHY OF AUTHORS

Aqsal Harris Pratama is an undergraduate student at Universitas Amikom Yogyakarta. He is currently studying Informatics, with a particular interest in the fields of Artificial Intelligence, Natural Language Processing, and Data Science. Email : aqsal.pr@students.amikom.ac.id

Mardhiya Hayaty is currently working as Assistant Professor, at Informatics Department in Faculty of Computer Science, Universitas Amikom Yogyakarta, Indonesia. Her research interest is natural language processing, sentiment analysis, and automatic text summarization. She can be contacted at email: mardhiya_hayati@amikom.ac.id