Air Force Institute of Technology

AFIT Scholar

Faculty Publications

2-3-2023

Emotion Classification of Indonesian Tweets using Bidirectional LSTM

Aaron Glenn

Phillip LaCasse
Air Force Institute of Technolgy

Bruce A. Cox
Air Force Institute of Technology

Follow this and additional works at: https://scholar.afit.edu/facpub

Part of the Artificial Intelligence and Robotics Commons, and the Data Science Commons

Recommended Citation

Glenn, A., LaCasse, P., & Cox, B. (2023). Emotion classification of Indonesian Tweets using Bidirectional LSTM. Neural Computing and Applications. https://doi.org/10.1007/s00521-022-08186-1

This Article is brought to you for free and open access by AFIT Scholar. It has been accepted for inclusion in Faculty Publications by an authorized administrator of AFIT Scholar. For more information, please contact richard.mansfield@afit.edu.

Neural Computing and Applications
https://doi.org/10.1007/s00521-022-08186-1

ORIGINAL ARTICLE

Emotion classification of Indonesian Tweets using Bidirectional LSTM

Aaron Glenn[1] · Phillip LaCasse[1] · Bruce Cox[1]

Received: 24 May 2022 / Accepted: 22 November 2022
This is a U.S. Government work and not under copyright protection in the US; foreign copyright protection may apply 2023

Abstract
Emotion classification can be a powerful tool to derive narratives from social media data. Traditional machine learning models that perform emotion classification on Indonesian Twitter data exist but rely on closed-source features. Recurrent neural networks can meet or exceed the performance of state-of-the-art traditional machine learning techniques using exclusively open-source data and models. Specifically, these results show that recurrent neural network variants can produce more than an 8% gain in accuracy in comparison with logistic regression and SVM techniques and a 15% gain over random forest when using FastText embeddings. This research found a statistical significance in the performance of a single-layer bidirectional long short-term memory model over a two-layer stacked bidirectional long short-term memory model. This research also found that a single-layer bidirectional long short-term memory recurrent neural network met the performance of a state-of-the-art logistic regression model with supplemental closed-source features from a study by Saputri et al. [8] when classifying the emotion of Indonesian tweets.

Keywords DNN · BiLSTM · LSTM · Emotion classification · Machine learning

1 Introduction

Emotion classification has become an increasingly popular field for research with the growing preeminence of the Internet and social media. Prior to 2005, the text of only 42 journal articles or conference proceedings in three prominent academic databases, IEEE Xplore Digital Library, ScienceDirect, and SpringerLink, contained the exact phrase, "emotion classification." From 2005 to 2012, the same search yielded 591 results; from 2013 to 2021, the results of the same search had jumped to 3529. In that time, emotion classification has been the subject of analysis in diverse domains including image recognition [1], animation [2], root cause diagnosis [3], online reviews [4], and social network analysis [5–7]. Twitter, in particular, has become a lightning rod for researchers aiming to model human language through various machine learning techniques. Notably, most of this research has been performed in the English-speaking world—leaving other languages relatively untouched and ready for discovery [8].

This study aims to meet or exceed the performance of state-of-the-art Indonesian emotion classification by using exclusively open-source data and models. Because of the relative scarcity of emotion classification research in non-English languages, it is appropriate to explore whether open-source resources are suitable to produce useful models or if tailored, specific models are required.

A reasonable next step in extending a traditional machine learning model (e.g., logistic regression, support vector machines) is to employ more advanced algorithmic approaches such as neural networks. This research examines the efficacy of recurrent neural network (RNN) variants within a deep neural network architecture, specifically long short-term memory (LSTM), bidirectional LSTM (BiLSTM), stacked BiLSTM and gated recurrent unit (GRU).

These models will leverage pre-trained Word2Vec and FastText word embeddings provided by Saputri et al. [8]. Because RNNs train solely on a sequential representation of text, they have no dependence on proprietary dense features. This is in contrast to Saputri et al. final ensemble

✉ Aaron Glenn
aaron.k.glenn.mil@mail.mil

Phillip LaCasse
phillip.lacasse@afit.edu

Bruce Cox
bruce.cox@afit.edu

[1] Department of Operational Sciences, Air Force Institute of Technology, 2950 Hobson Way, Wright-Patterson AFB, OH 45433, USA

Published online: 06 February 2023

model which trained a logistic regression model using bespoke subject matter expert crafted features such as an Indonesian sentiment lexicon, bag-of-words, part of speech tagging, and emoticon lists. The validity of our models will be judged based on the averaged validation accuracy in a tenfold cross-validation experiment.

This research makes a novel contribution to the larger domain of language-agnostic emotion classification in three ways.

- We demonstrate that, all else being equal, RNN models outperform the traditional machine learning approaches of logistic regression, random forest, and support vector machines employed in the original study. The implication is that the existing underlying relationships are sufficiently nuanced as to require more advanced algorithmic approaches to be extracted.
- We demonstrate that open-source pre-trained Word2-Vec and FastText vector embedding approaches, combined with advanced RNN models, produce results competitive with traditional machine learning models employing dense, proprietary features. The implication is that the one million or so Indonesian tweets used to pre-train the Word2Vec and FastText embeddings are sufficiently representative of the corpus of Indonesian tweets that might be queried for analysis such as this. In natural language processing (NLP), it is always of interest whether a broad corpus can be useful for the specific application of interest; in this case, the answer appears to be in the affirmative.
- We demonstrate that, of the four RNN variants employed, BiLSTM produces significantly higher accuracy than stacked BiLSTM (significant with a p value of $p = .037$) but all other pairwise comparisons exhibit no significant differences. The implication is that these emotion classification models are generally agnostic to RNN variant with respect to classification accuracy and that choice of RNN variant can be made based on other considerations such as processing time or some other resource constraint.

The remainder of this document is organized as follows. Section 2 provides an overview of relevant background information and literature review. Section 3 details the methodology behind the various RNN models. Section 4 presents the results of the various RNN models. Finally, Sect. 5 discusses the conclusions drawn from the results.

2 Related work

2.1 Emotion classification

Emotion classification seeks to classify text into various human emotions as opposed to a binary response such as positive or negative. Emotion classification can be useful for general purpose sentiment mining due to the unstructured nature of social media [8]. Binary sentiment analysis is frequently more suitable for specific datasets such as movie reviews where there is a known subject and somewhat predictable format. Tweets have no known subject or predictable format.

A 2018 study by Saputri et al. produced the first publicly available labeled Indonesian twitter dataset for emotion classification. The study performed emotion classification on Indonesian twitter data using traditional machine learning techniques such as support vector machine (SVM), logistic regression, and random forest with an ensemble of lexicons and embedding techniques. The results of the study's models are published but the models themselves are not. The study also published pre-trained Word2Vec and FastText embeddings that were trained on over one million Indonesian tweets. This study represents the state-of-the-art in emotion classification performance for Indonesian tweets.

Traditional machine learning techniques like SVM, logistic regression, and random forest rely on "external resources or manual annotation to learn semantic and syntax features" [9]. Saputri et al.'s final model was trained on an ensemble of dense features, some of which are not publicly available (i.e., an emoticon list, an emotion word list and Vania's sentiment lexicon). Deep learning techniques like neural networks can automatically "extract the features of context" and thus have "universal applicability" [9]. The goal of this research is to meet or exceed the capabilities of traditional machine learning techniques that leverage private data using deep learning techniques with exclusively open-source data/models.

2.2 Recurrent neural network variants

LSTMs were first proposed in 1997 as a solution to RNN's issues with losing memory of its initial inputs [10]. For instance, when performing sentiment analysis on a lengthy movie review, an RNN will gradually forget the first few words (i.e., "I loved this movie, but ..."), causing the model to lose full context of the review and potentially make an incorrect sentiment classification [11]. LSTMs mitigate this issue by storing important inputs in long-term memory.

BiLSTM in conjunction with embedding techniques has been shown to be more accurate than standard LSTMs at tasks such as domain recognition, sentiment analysis, and emotion classification [12–15]. The key difference between a BiLSTM and a standard LSTM is that the BiLSTM can capture both past and future information, as opposed to only past information in standard LSTMs. This allows the BiLSTM to capture stronger dependency relationships between words and phrases [15]. This study will include a BiLSTM model in its comparative analysis.

BiLSTM have been used to perform eleven-dimensional emotion classification on Arabic tweets [14]. In this paper, the dataset consisted of 4381 Arabic tweets labeled with the following emotions: anger, anticipation, disgust, fear, joy, love, optimism, pessimism, sadness, surprise and trust. The BiLSTM model outperformed several traditional machine learning baselines such as SVM, support vector classifier and others when comparing validation accuracy. This study will perform multi-dimensional emotion classification of tweets using BiLSTM, albeit in five dimensions.

Stacking multiple BiLSTM layers has been shown to improve sentiment analysis prediction accuracy in Chinese micro-blog data in comparison with single-layer BiLSTM, standard LSTM and traditional machine learning techniques [9]. In this paper, it was argued that the extra layers aided in the extraction of complex features in the Chinese language. This study will include a stacked BiLSTM architecture in its comparative analysis.

GRU is a simplified variant of LSTM and has been shown to achieve similar performance with less training time [16]. Other comparisons of LSTMs and GRUs across multiple datasets have proven to be inconclusive, claiming that the choice of one versus the other is heavily dependent on the dataset and task [17]. A GRU model will be leveraged in this study as part of its comparative analysis.

BiLSTMs have been used in conjunction with embedding and fully connected (FC) layers to perform emotion classification of tweets [13]. In this paper, the author proposed a model architecture which combines a GloVe embedding layer and a BiLSTM layer with two subsequent FC layers. This structure outperforms standard LSTM and BiLSTM without an embedding layer in validation accuracy. Similar structures that combine BiLSTM with FC layers have been proposed that perform domain classification of acoustic communication [18]. This study will propose an architecture using embedding layers, RNN layers and FC layers.

2.3 Limitations

A key limitation for this study's model is that it is trained exclusively on tweets determined to have one singular emotion, leaving out tweets that display multiple emotions or no emotion. This study's emotion classifier is likely to predict whichever emotion is more dominant among multiple emotions.

Saputri et al. [8] study of Indonesian emotion classification provides pre-trained embeddings in the form of Word2Vec and FastText. However, multiple studies have found GloVe embeddings to be the optimal embedding technique when paired with LSTM layers [13, 19]. The exclusion of GloVe embeddings could prove to be a limitation for this study's various models and their accuracy.

3 Methodology

3.1 LSTM

The complexities of the LSTM cell allow for improved performance, faster training and detection of long-term dependencies in the data in comparison with traditional RNNs. The LSTM cell state stores long-term information and the hidden state stores short-term information. The LSTM cell is visualized in Fig. 1.

The cell state (c_t) is mathematically expressed in Eq. 1. The hidden state (h_t) is expressed in Eq. 2. The current inputs (x_t), as well as the previous hidden state, are analyzed in the main layer (g_t) expressed in Eq. 3. LSTM cells also possess three gate controllers that handle the storage and erasure of information from stored memory [11]. The forget, input and output gates are expressed in Eqs. 4, 5 and 6, respectively. These gates apply the sigmoid activation function to weighted inputs which produces element-wise binary outputs that allow inputs to either be retained or forgotten. The input gate (i_t) controls which inputs are stored as long-term memory. The forget gate (f_t) controls which portions of long-term memory should be forgotten. And the output gate (o_t) controls which portions of long-term memory will be output (y_t) at the current time step. Here, W represents a weight matrix, b refers to bias, x_t refers to the current input, $\otimes$ refers to element-wise

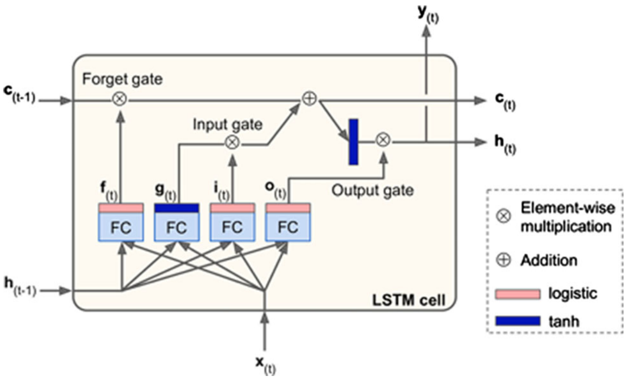

Fig. 1 LSTM cell (Reprinted, with permission, from Géron, 2019 © O'Reilly) [11]

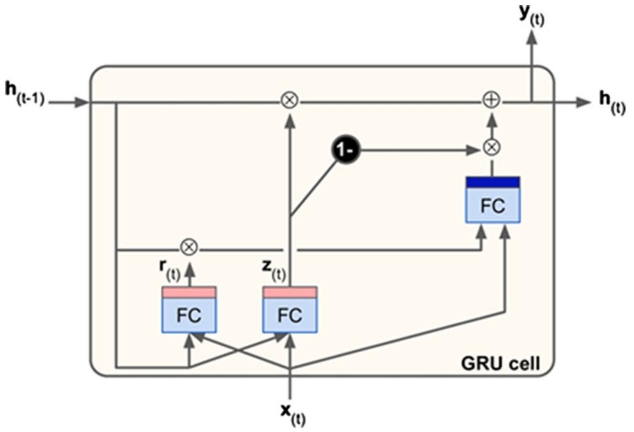

Fig. 2 GRU Cell (Reprinted, with permission, from Géron, 2019 © O'Reilly) [11]

multiplication, $\cdot$ refers to dot product and σ refers to the sigmoid activation function.

$$c_t = i_t \otimes g_t + f_t \otimes c_{t-1} \tag{1}$$

$$h_t = y_t = o_t(tanh(c_t)) \tag{2}$$

$$g_t = tanh(W_c[x_t, h_{t-1}] + b_c) \tag{3}$$

$$f_t = \sigma(W_f[x_t, h_{t-1}] + b_f) \tag{4}$$

$$i_t = \sigma(W_i[x_t, h_{t-1}] + b_i) \tag{5}$$

$$o_t = \sigma(W_o[x_t, h_{t-1}] + b_o) \tag{6}$$

3.2 GRU

The GRU cell was proposed as a simplified version of an LSTM cell [17]. The primary simplifications of GRU over LSTM are that both state vectors are merged into a single vector, the output gate is removed, and a single gate controller controls the update gate and reset gate.

The update gate (z_t) is mathematically expressed in Eq. 7 and the reset gate (r_t) is expressed in Eq. 8. The reset gate controls which information from past memory to introduce as new memory content, expressed as g_t in Eq. 9. The update gate (z_t) feeds into the final state vector (h_t) to determine which portions of the current memory (g_t) and past memory (h_{t-1}) to output (y_t) at the current time step. This behavior is expressed in Eq. 10. Here, W and U represent weight matrices, $\otimes$ refers to element-wise multiplication, $\cdot$ refers to dot product and b represents bias.

$$z_t = \sigma_g(W_z \cdot x_t + U_z \cdot h_{t-1} + b_z) \tag{7}$$

$$r_t = \sigma_g(W_r \cdot x_t + U_r \cdot h_{t-1} + b_r) \tag{8}$$

$$g_t = tanh(W_h \cdot x_t + U_h \cdot (r_t \otimes h_{t-1}) + b_h) \tag{9}$$

$$h_t = y_t = z_t \otimes h_{t-1} + (1 - z_t) \otimes g_t \tag{10}$$

Recurrent dropout has been proposed as a method of applying dropout to gated architectures such as LSTM and GRU [20]. Traditional dropout methods which are designed for feed-forward networks can cause loss of long-term memory when used with LSTM or GRU. Recurrent dropout is applied to the hidden state update vectors rather than the hidden states, themselves. This distinction has shown an increase in network performance using LSTMs to predict Twitter sentiment analysis.

3.3 BiLSTM

BiLSTMs "consist of two LSTMs that are run in parallel: one on the input sequence and the other on the reverse of the input sequence" [15]. The hidden state of the BiLSTM is the result of the concatenation of the forward and backward hidden states at each time step, allowing the BiLSTM to capture both past and future information. The BiLSTM model is visualized in Fig. 3.

Similar to BiLSTM, stacked BiLSTM can extract "rich contextual information from both past and future time sequences" [9]. However, stacked BiLSTM possesses more layers to perform feature extraction, as opposed to BiLSTM which has only one hidden layer per direction. In a two-layer stacked BiLSTM, the input sequence enters the hidden layers in the forward direction to extract information from all past time steps, while it also passes through the hidden layers in the reverse direction to extract information from all future time steps. After this, the second hidden layers receive outputs from the first hidden layers as their inputs to produce further feature extraction. And finally, the output layer integrates both of the second hidden layers' output vector as its final output.

3.4 Deep neural networks

A FC layer is the result of all neurons in a layer being connected to every neuron in the previous layer. There is

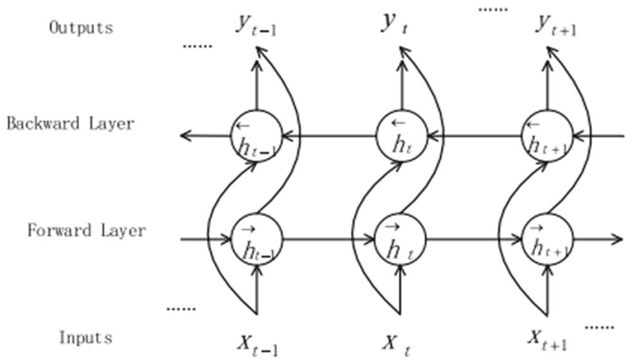

Fig. 3 Bidirectional recurrent neural network (Reprinted, with permission, from Xiao and Liang, 2016 © Springer) [15]

often an extra bias neuron, which outputs a value of 1. Networks with FC layers learn and adapt by making predictions at each training instance, then updating the weights within the network that correspond with correct predictions. For every neuron that produces an incorrect prediction, the network "reinforces the weights from the inputs that would have contributed to the correct prediction" [11]. This iterative process is referred to as backpropagation. When multiple FC layers are stacked together, it is referred to as a deep neural network (DNN). This structure forms the basis for each variant of neural network (i.e., recurrent neural networks, convolutional neural networks, etc.).

DNNs learn by training on a portion of the training data, called a batch, performing backpropagation and then repeating this process until all of the training data have been used. One complete pass over the entire training set is referred to as an epoch. Batch size and the number of epochs are both tunable hyperparameters. Another key tunable hyperparameter is the number of FC layer units. This simply refers to the dimensionality of the FC layer. Increasing the number of FC layer units can increase the DNNs ability to learn complex relationships at the cost of more computational resources.

ReLU (rectified linear unit) activation function is often used in DNN FC layers because it does not saturate for positive values [13]. It is also fast to compute as the derivative of the slope is equal to one [11].

Dropout is used in between DNN FC layers in order to reduce overfitting. Dropout excludes certain input and FC layer neurons with some probability, typically between 0.2 and 0.5. Dropout effectively causes a unique neural network to be generated at each training step, resulting in an ensemble of smaller networks that are robust against overfitting. Dropout tends to slow down model convergence, however, it does typically result in a significantly improved model when tuned appropriately [11].

Pooling layers are another method of mitigating overfitting in DNN. The goal of a pooling layer is to shrink an input in order to identify the most important feature. The pooling layer also reduces dimensionality, which helps reduce parameters and computational complexity [11].

RMSProp (Root Mean Squared Propagation) is a learning algorithm that uses the concept of decay to ignore distant observations from the past and focus on more recent inputs. RMSProp is an adaptive learning algorithm and therefore "requires less tuning of the learning rate hyperparameter" [11]. RMSProp has been used to perform emotion classification [13].

3.5 Word embeddings

Word embeddings represent words in vector form such that the distance between vectors represents the semantic relations between respective words [21]. Word2Vec is an example of a static embedding, meaning that the method learns one fixed embedding per word in the vocabulary. In an unpublished, draft-form online textbook, Dr. Dan Jurafksy observes that the intuition of Word2Vec is that instead of counting how often each word *w* occurs near, say, *apricot*, we'll instead train a classifier on a binary prediction task: "Is word *w* likely to show up near *apricot*?" This method uses self-supervision, therefore, it does not require a labeled dataset. Jurafksy calls Word2Vec "fast" and "efficient to train" on unsupervised data. As such, there are many examples of pre-trained Word2Vec embeddings publicly available. For example, Saputri et al. make publicly available a 400-dimension Word2Vec embeddings model that was trained on over one million Indonesian tweets.

FastText is another static embedding technique and an extension of Word2Vec. FastText represents words as themselves along with a "bag of constituent n-grams," according to Jurafsky. For instance, if $n = 3$, the word there would be represented by the sequence there along with the character n-grams: <th, the, her, ere, re>. Then, the skip-gram embedding is learned for each constituent n-gram and the word there is represented by the sum of all the embeddings of its constituent n-grams. Saputri et al. also provide a pre-trained 100-dimensional FastText model trained on the same set of Indonesian tweets.

3.6 Dataset

Saputri et al. produced a dataset of 4401 Indonesian tweets labeled into five emotion classes: love, anger, sadness, fear and joy. This dataset was labeled by two individuals and evaluated using a Cohen Kappa measurement—a statistic for measuring interrater reliability. A score above 0.81 is considered to be "almost perfect agreement" between two raters [22]. This study produced a Kappa score of 0.917. Multi-emotion tweets and no-emotion tweets were

Table 1 Classification balance of tweets

Emotion class	Frequency
Love	637
Fear	649
Sadness	997
Joy	1017
Anger	1101
Total	4401

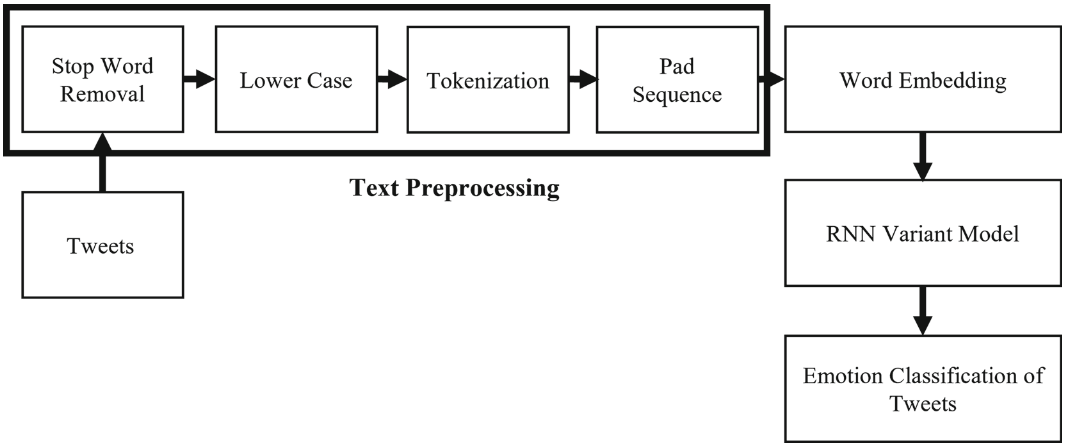

Fig. 4 Proposed framework for emotion classification

excluded from this dataset. The data classifications are slightly imbalanced with the distribution seen in Table 1.

Saputri et al. dataset contains emoticons (i.e., :) expressing joy). Saputri et al. created an unpublished list of emoticons and manually labeled each emoticon with their corresponding emotion. This study will not recreate an emoticon list, but instead allow the models to learn such features automatically.

The tweets produced by the 2018 study already have some level of preprocessing. Usernames with the @ symbol have been replaced with the generic [USERNAME], URL's, and hyperlinks have been replaced with the generic [URL], and sensitive numbers such as phone numbers, invoice numbers, and courier tracking numbers have been replaced with the generic [SENSITIVE-NO].

The preprocessing steps performed in this study were the following: eliminating punctuation, numbers, other special characters, lower case conversion, stop word removal, and tokenization. Stop words are defined as extremely common words which are of little value such as "a, an, he, is, it." Tokenization refers to splitting a sentence into separate "tokens," often delineated by white space [23]. These preprocessing steps, as well as the subsequent modeling steps, can be visualized in Fig. 4.

The data and pre-trained embeddings were downloaded from Saputri et al. GitHub page[1] in CSV form. Additional preprocessing was performed on the tweets by removal of stop words via the Natural Language Toolkit (NLTK) Indonesian Stop Word library. Tokenization is performed using Keras' Tokenizer function, which converts the tweets into lowercase tokens and then into sequences of integers. Each tweet is set to the same length by padding the sequences with zeroes up to the maximum length of the longest tweet.

The pre-trained embeddings are then loaded into a dictionary and cross-referenced with the words from the dataset of Tweets. This process results in finding each word in the dataset's vector representation. The embeddings are stored in a matrix the size of the maximum tweet length times the dimensionality of the pre-trained embeddings (i.e., 64 × 400 or 64 × 100). Words that exist in the tweets but not in the embeddings are given a value of zero.

3.7 Evaluation metrics

The Saputri et al. [8] study produced a comparative study of various machine learning techniques leveraging many different dense features. This research will compare the results of its various RNN models using Word2Vec and FastText pre-trained embeddings against the Saputri study's logistic regression model. This research will also perform a direct comparison of its models to Saputri's using only word embedding features (i.e., Word2Vec and FastText). Saputri et al. used tenfold cross-validation to split their data into training and validation sets. This study will identify each fold's optimal model by identifying the epoch with the global maxima of validation accuracy. All evaluation metrics are then averaged using the optimal models from all tenfold. This study will also examine precision, recall and F1 score for predictions of each respective class from the model with the highest validation accuracy.

Precision is defined as the ratio of true positive results over the total predicted positive (true positive + false positive) results. Recall is defined as the ratio of true positive over the total actual positive (true positive + false negative). F1 score represents the harmonic mean of precision and recall.

[1] https://github.com/meisaputri21/Indonesian-Twitter-Emotion-Dataset.

3.8 Proposed model architecture

This experiment consists of four architectural models: LSTM, GRU, BiLSTM and two-layer stacked BiLSTM. Parameters were held constant across each model with the exception that hidden layer units in BiLSTM layer were halved to maintain dimensionality. All four models are trained for 20 epochs per fold in a tenfold cross-validation experiment. Batch size is fixed to 64 for each epoch. The RMS Prop optimizer uses the default learning rate of 0.001 and default decay rate, ρ, of 0.9. Loss is measured by categorical cross-entropy. Per the normal k-fold cross-validation procedures the full dataset (i.e., all 4401 tweets) are randomly split, without replacement, into $k = 10$ fold. Ninefold are used for training each model and the 10^{th} holdout fold is used for calculating validation accuracy for each model. The tenfold were redrawn for each model. The tenfold cross-validation process thus results in 10 trained models per architecture (i.e., 40 models total), each with its own validation accuracy.

Table 2 illustrates the architecture of the LSTM, GRU, and BiLSTM models. The first layer of the model is the embedding layer which uses the pre-trained Word2Vec or FastText embeddings from Saputri et al. This layer converts each word in the corpus of tweets to a 400- or 100-dimensional vector representation, respectively. The input length is set to the maximum length of the longest tweet in the dataset (i.e., 64 words), thus every tweet is padded with zeroes up to that length. The size of the output of the embedding layer is a 64 × 400/100 matrix for every unit in the batch (i.e., represented as (Batch Size, 64, 400/100)). The embedding layer is trainable, meaning that the model will adjust the values of the embeddings during backpropagation. This matrix then passes through a one-dimensional spatial dropout layer with a rate of 0.2 to avoid overfitting.

The output of the spatial dropout layer feeds into the LSTM/GRU/BiLSTM layer with 512 units (256 × 2 for the BiLSTM layer). Thus, the output of the LSTM/GRU/BiLSTM layer is a 64 × 512 matrix. The GRU layer has slightly fewer parameters than the LSTM layer due to its simplicity. The BiLSTM's output shape accounts for both the forward and backwards directions, therefore, in order to have an output dimension of 512, the BiLSTM layer must possess half the cell units or in this case, 256. This accounts for the BiLSTM layer having less parameters than both the LSTM and GRU layers. In the stacked BiLSTM model, the two layers of BiLSTMs are stacked upon one another. The dimensionality of the other models is preserved; however, the number of parameters is increased from roughly 3 million to 4.7 million. The LSTM/GRU/BiLSTM layer has dropout at its inputs of 0.2 and recurrent dropout of its recurrent state of 0.2 to avoid overfitting [20]. This output is then fed into a one-dimensional global max pooling layer. The output of the Global Max Pooling layer is a vector of length 512.

Next, the output vector is fed into two FC layers of decreasing size. The first FC layer is of size 512 and the second is of size 256. Both layers utilize ReLU activation functions. Dropout is applied after both FC layers at a rate of 0.5 to avoid overfitting. The final output layer reduces the vector to length 5 with a softmax activation function which converts the vector of numbers into a vector of probabilities. The softmax code returns the index of the highest probability (i.e., the predicted emotion).

4 Results and analysis

4.1 Programming platform

This work was performed using the Python Programming language (3.7.12) with the following key packages: Numpy (1.19.15), Pandas (1.1.5), Keras (2.5.0), SkLearn (0.22) and NLTK (3.4.5). The primary experiments were conducted in a Google Colab environment using Google Colab's GPU hardware accelerator.

Table 2 Model structure for comparative experiment

Layer	Output shape	Number of parameters
Word2Vec/FastText embedding	Batch size, 64, 400/100	8 M/2M
Spatial dropout 1-D	Batch size, 64, 400/100	0
LSTM/GRU/BiLSTM	Batch size, 64, 512	1.2 M/0.9M/0.7M
BiLSTM (optional)	Batch size, 64, 512	1.5 M
Global max pooling	Batch size, 512	0
FC #1	Batch size, 512	2,62,656
Dropout	Batch size, 512	0
FC #2	Batch size, 256	1,31,328
Dropout	Batch size, 256	0
FC #3	Batch size, 5	1285

Table 3 Comparative experiment results[a]

Embedding	LSTM	Glenn 2022				Saputri et al. [8]		
		GRU	BiLSTM	2xBiLSTM[b]	LR	SVM	RF	LR+[c]
FastText	70.48%	70.17%	**70.71%**	**70.71%**	62.49%	62.27%	55.18%	69.73%
Word2Vec	65.92%	64.90%	66.33%	65.01%	61.83%	61.37%	53.03%	–

[a] Average validation accuracy in tenfold cross-validation

[b] Two-layer stacked BiLSTM

[c] Logistic regression with dense features

4.2 Cross-validation results

The results of this study's cross-validation comparative experiment can be seen in Table 3. These results are compared to Saputri et al.'s models using only embedding techniques as well as the best model from Saputri et al. that uses various dense features. These values represent the mean of the maximum validation accuracies for each of the 10 cross-validation splits.

Each of the models produced by this research outperform each model from Saputri et al. when embedding techniques are the only features considered. The best models from this study (FastText BiLSTM and two-layer stacked BiLSTM) outperform Saputri et al.'s logistic regression model with FastText embeddings by more than an 8% margin of accuracy in tenfold cross-validation. The best models from this research also outperform Saputri et al.'s random forest model by more than 15%.

Each of the models with FastText embeddings outperforms the best model from the comparison study when all features are considered, while each of the models using Word2Vec embeddings under-perform relative to its FastText counterpart. The increase in performance when using FastText over Word2Vec is consistent with the results of the logistic regression model from Saputri et al. The best performing models are the FastText BiLSTM and FastText two-layer stacked BiLSTM models with the same validation accuracy of 70.71%. The stacked BiLSTM model has significantly more parameters than the single-layer BiLSTM and yet does not produce higher validation accuracy. The distribution of results over tenfold for the FastText RNN variants can be seen in Fig. 5.

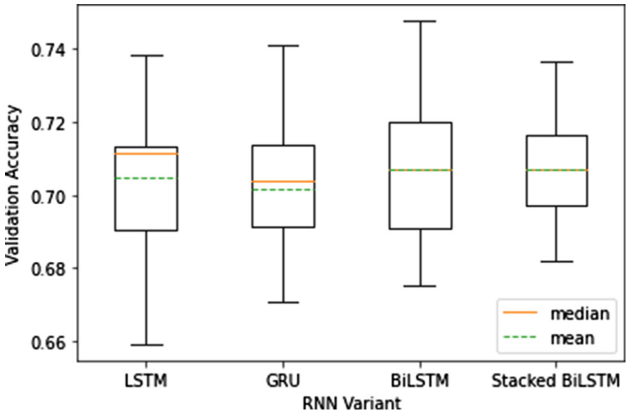

Fig. 5 FastText RNN variant CV results

4.3 Statistical significance

Two sets of statistical tests were performed. The first set compares the results of the models depicted in Table 3. The second set compares the four RNN variants against each other.

In the first case, the top performing RNN variants were BiLSTM and 2xBiLSTM, each with a classification accuracy of 0.7071. The top performing model employing open-source embeddings in the comparison study is the logistic regression model, achieving a classification accuracy of 0.6249. Applying a hypothesis test for equality of proportions of independent populations, the RNN models exhibit significantly higher accuracy than the logistic regression model ($z = 8.18$, $p < .0001$). The top performing RNN variants using open-source embeddings were not significantly different from the top model in the comparison study using dense, proprietary features denoted by LR+ in Table 3 ($z = 1.00$, $p = .3162$).

In the second case, an internal comparison was performed to determine any statistical significance among this study's models using the bootstrapping method. Bootstrapping is similar to cross-validation but with sample replacement. This distinction satisfies the ANOVA test's assumption of statistical independence among samples. The results of the bootstrapping experiment with 10 training splits can be seen in Table 4 and Fig. 6.

Table 4 10-Sample bootstrap results

Embedding	LSTM	GRU	BiLSTM	2xBiLSTM[a]
FastText	68.26%	68.56%	**68.62%**	68.09%

Average validation accuracy

[a] 2-layer stacked BiLSTM

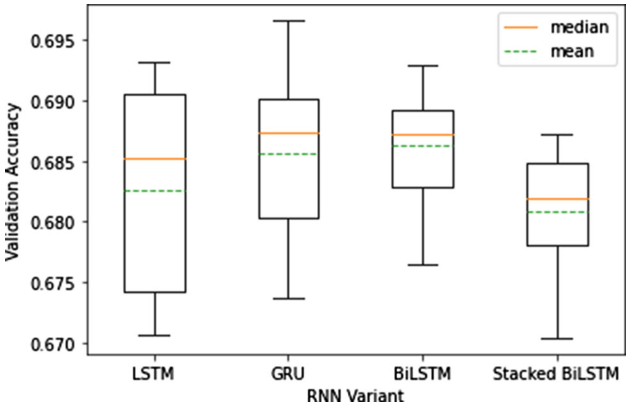

Fig. 6 FastText RNN variant bootstrap results

A single-factor ANOVA was performed on the bootstrap comparative results with a null hypothesis that all classifier means are equal and an alpha of 0.05. The single-factor ANOVA fails to reject the null hypothesis (*F*-stat = 1.35 and *p* = .27), meaning that the data cannot conclude that the four classifiers' means are not equal. In addition, post hoc pairwise t tests were conducted with a null hypothesis that the mean difference between two classifiers is zero and an alpha of 0.05. For the comparison of the single-layer BiLSTM model and the two-layer stacked BiLSTM model, this test rejects the null hypothesis (*p* = .037) which reveals a statistical significance between the two models. All other pairwise t tests were found to be statistically insignificant. Therefore, this research can conclude that the single-layer BiLSTM model is optimal in comparison with the two-layer stacked BiLSTM model.

4.4 Hyperparameter tuning

The single-layer BiLSTM model with FastText embeddings was chosen as the candidate model for hyperparameter tuning based on its statistical significance over the two-layer stacked BiLSTM, and it is possessing the highest validation accuracy in both the cross-validation and bootstrap comparative experiments. Hyperparameter tuning can identify model dependencies that can potentially be exploited in order to boost performance.

Table 5 BiLSTM hidden layer unit tuning results

Hidden layer units	256	512	1024
Val accuracy[a]	70.71%	**70.83%**	70.64%

Single-layer stacked BiLSTM with FastText embeddings

[a]Average validation accuracy in tenfold cross-validation

Table 6 Batch size tuning results

Batch size	16	32	64	128
Val accuracy[a]	69.44%	70.28%	**70.83%**	70.42%

Single-layer stacked BiLSTM with FastText embeddings

[a]Average validation accuracy in tenfold cross-validation

The first hyperparameter for consideration is the number of units in the BiLSTM's hidden layer. Variation in the number of units in the BiLSTM's hidden layer produces the following results in Table 5. Increasing the number of hidden layer units in the BiLSTM from 256 to 512 leads to an increase in validation accuracy up to 70.83%. However, increasing the hidden layer units up to 1024 shows a decrease in performance.

Lowering the batch size can increase accuracy at the cost of additional run time. Variation in batch size with BiLSTM layer set to 512 can be seen in Table 6. Decreasing the batch size below 64 did not produce higher accuracy. Notably, increasing the batch size to 128 also caused a decrease in validation accuracy, suggesting 64 is an optimal batch size.

After tuning, the best model is determined to be a FastText single-layer BiLSTM with 512 units in the BiLSTM layer and a batch size of 64. This model is able to produce a study-best 70.83% accuracy average across tenfold cross-validation. The final model architecture can be seen in Table 7. Note: The output shape of the BiLSTM layer is twice the BiLSTM layer units to account for both the forward and backward directions (i.e., 512 × 2 = 1024).

Tunable elements such as learning rate, dropout rates and the number of neurons in FC layers 1 and 2 were not tuned due to computational resource constraints. Learning and dropout rates largely impact convergence time within a certain number of epochs. The model was found to

Table 7 Final single-layer BiLSTM model

Layer	Output shape	Number of parameters
FastText embedding	Batch size, 64, 100	2000000
Spatial dropout 1-D	Batch size, 64, 100	0
BiLSTM	Batch size, 64, 1024	2510848
Global max pooling	Batch size, 1024	0
FC #1	Batch size, 512	524800
Dropout	Batch size, 512	0
FC #2	Batch size, 256	131328
Dropout	Batch size, 256	0
FC #3	Batch size, 5	1285

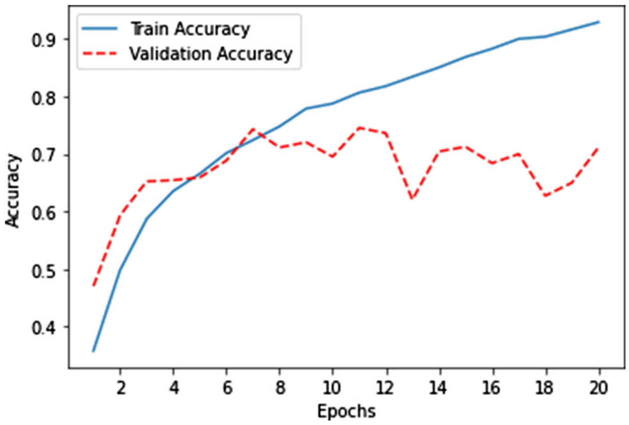

Fig. 7 Single-layer BiLSTM model accuracy

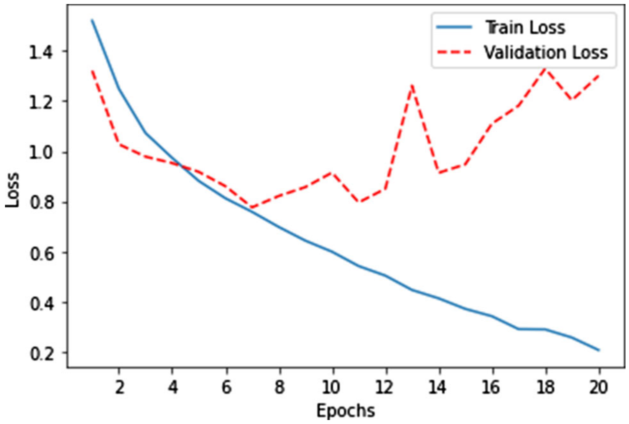

Fig. 8 Single-layer BiLSTM model loss

Table 8 Classification report

Emotion class	Precision	Recall	F1-Score
Love	78%	88%	83%
Joy	77%	65%	71%
Anger	75%	85%	80%
Sadness	62%	65%	64%
Fear	84%	71%	77%
Avg/Total	76%	75%	75%

Table 9 Classification report (Saputri et al. [8])

Emotion class	Precision	Recall	F1-Score
Love	64%	75%	69%
Joy	81%	60%	69%
Anger	61%	81%	70%
Sadness	89%	72%	80%
Fear	65%	53%	59%
Avg/Total	70%	68%	68%

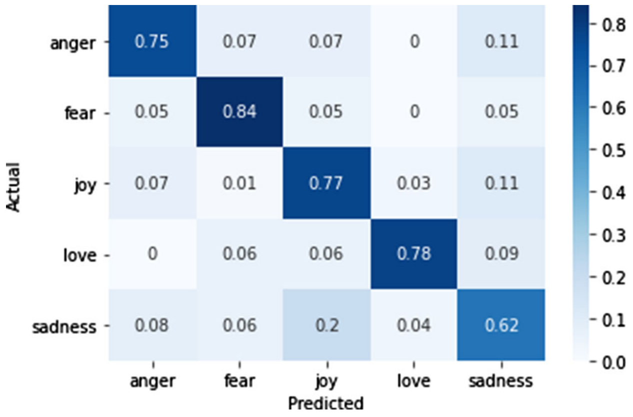

Fig. 9 Single-layer BiLSTM confusion matrix

converge to a global maximum validation accuracy within the given number of epochs during every training split during the cross-validation experiment. Additionally, the RMSprop optimizer is an adaptive learning algorithm which makes the default learning rate suitable in most cases. The number of neurons in the FC layers was chosen to preserve the dimensionality of the RNN layer's output with a stepwise decrease until the final FC layer.

4.5 Model Examination

The single-layer BiLSTM model will now be further examined to show its behavior during training. The model's accuracy and loss values over 20 epochs can be seen in Figs. 7 and 8. These values were extracted from one of the 10 cross-validation training splits from the previous experiment.

The optimal model was identified by maximizing the validation accuracy at epoch 11 (74.55%). The slight increase in validation loss after epoch 7 suggests some slight overfitting, however, this increase is not significant enough to rule out the optimal model at epoch 11. It is noteworthy that validation accuracy at epoch 7, the validation loss minimum, is 74.32%, which is still greater than the optimal model from Saputri et al. This suggests that the model is robust against overfitting.

Examination of the model's classification report reveals that the fear class has the highest level of precision at 84%, which indicates a false positive rate of 16%. Recall for the fear class is also high with a score of 71%. This performance is in contrast with the baseline model from Saputri et al. with a precision rate of 65% and recall rate of 53% for the fear class. The classification reports from this study and Saputri et al. can be seen in Tables 8 and 9.

The confusion matrix in Fig. 9 demonstrates the classification performance of the model in greater detail. It can clearly be seen that the model produces the most true positives in the fear category—meaning 84% of its "fear" predictions are actual "fear" tweets according to the labeled dataset. The model's worst performing category is "sadness," which could be due to the prevalence of sarcasm and irony on social media [24]. This could explain why sadness is mistaken for joy 20% of the time.

5 Conclusions and future work

The results of this study demonstrate the efficacy of RNNs when leveraging pre-trained FastText embeddings in comparison with traditional machine learning techniques. Specifically, these results show that RNN variants can produce more than an 8% gain in accuracy in comparison with logistic regression and SVM techniques and a 15% gain over random forest.

This research found a statistical significance in the performance of a single-layer BiLSTM model over a two-layer stacked BiLSTM model. This research also found that single-layer BiLSTM models produce comparable validation accuracy when compared to the best model from Saputri et al. The final ensemble method from Saputri et al. was a logistic regression model leveraging FastText pre-trained embeddings, bag-of-words feature extraction, an emotion word list, and several other lexical features for a final validation accuracy of 69.73%. This study's single-layer BiLSTM model achieves a validation accuracy of 70.83% using only the pre-trained FastText embeddings. This suggests that RNNs are successfully able to automatically extract the dense features manually provided in the study by Saputri et al. (i.e., emoticons, parts of speech, etc.). These results satisfy this research's goal of meeting or exceeding the performance of Saputri et al.'s traditional machine learning methods using exclusively open-source data and models.

Future work could include producing a model that handles tweets with neutral emotion or multi-emotion tweets. Saputri et al. indicate they may produce a dataset in the future that has multi-emotion and neutral emotion labeling [8]. It may be necessary to acquire additional labeled data from other Indonesian sources in order to train a model that can predict neutral and multiple emotions.

Other research areas such as topic modeling and network analysis could be applied in conjunction with this emotion classifier model in order to further expand upon its ability to provide characterization and understanding of societal behaviors. For example, network analysis could delineate social groups by examining Twitter interactions and emotion classification could be used to determine how those social groups respond to specific topics.

Data availability All data generated or analyzed during this study are included in this published article (and its supplementary information files).

Declarations

Conflict of interest All authors declare that they have no conflict of interest.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00521-022-08186-1.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Zhang H, Xu D, Luo G, Kangjian H (2022) Learning multi-level representations for affective image recognition. Neural Comput Appl. https://doi.org/10.1007/s00521-022-07139-y
2. Nikita J, Vedika G, Shubham S, Agam M, Ankit C, Santosh KC (2021) Understanding cartoon emotion using integrated deep neural network on large dataset. Neural Comput Appl. https://doi.org/10.1007/s00521-021-06003-9
3. Diao Y, Lin H, Yang L, Fan X, Chu Y, Di W, Kan X, Bo X (2020) Multi-granularity bidirectional attention stream machine comprehension method for emotion cause extraction. Neural Comput Appl 32(12):8401–8413. https://doi.org/10.1007/s00521-019-04308-4
4. Li D, Li M, Han G, Li T (2021) A combined deep learning method for internet car evaluation. Neural Comput Appl 33(10):4623–4637. https://doi.org/10.1007/s00521-020-05291-x
5. Ghanbari-Adivi F, Mosleh M (2019) Text emotion detection in social networks using a novel ensemble classifier based on parzen tree estimator (tpe). Neural Comput Appl 31(12):8971–8983. https://doi.org/10.1007/s00521-019-04230-9
6. Nagarajan S, Gandhi U (2019) Classifying streaming of twitter data based on sentiment analysis using hybridization. Neural Comput Appl 31(5):1425–1433. https://doi.org/10.1007/s00521-018-3476-3
7. Chen J, Yan S, Wong KC (2020) Verbal aggression detection on twitter comments: convolutional neural network for short-text sentiment analysis. Neural Comput Appl 32(15):10809–10818. https://doi.org/10.1007/s00521-018-3442-0
8. Saputri MS, Mahendra R, Adriani M (2018) Emotion Classification on Indonesian Twitter Dataset. In: Proceeding of

international conference on asian language processing, pages 90–95. https://doi.org/10.1109/IALP.2018.8629262

9. Zhou J, Lu Y, Dai HN, Wang H, Xiao H (2018) Sentiment Analysis of Chinese Microblog Based on Stacked Bidirectional LSTM. In: 2018 15th international symposium on pervasive systems, algorithms and networks (I-SPAN) pages 162-167. IEEE. URL https://doi.org/10.1109/ACCESS.2019.2905048
10. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
11. Aurélien G (2019) Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow, 2nd edn. O'Reilly Media Inc., California
12. Rathor S, Agrawal S (2021) A robust model for domain recognition of acoustic communication using bidirectional lstm and deep neural network. Neural Comput Appl 33(17):11223–11232. https://doi.org/10.1007/s00521-020-05569-0
13. Devi S, Naveenkumar K, Ganesh SS, Ritesh S (2021) Location Based Twitter Emotion Classification for Disaster Management. In: 2021 Third international conference on inventive research in computing applications (ICIRCA) pages 664-669. IEEE. URL https://doi.org/10.1109/ICIRCA51532.2021.9544994.
14. Elfaik H, Nfaoui EH (2020) Deep bidirectional LSTM network learning-based sentiment analysis for Arabic text. J Intell Syst 30(1):395–412. https://doi.org/10.1515/jisys-2020-0021
15. Xiao Z, Liang P (2016) Chinese Sentiment Analysis Using Bidirectional LSTM with Word Embedding. In: International conference on cloud computing and security, pages 601–610. https://doi.org/10.1007/978-3-319-48674-1_53
16. Liang X, Liu Z, Ouyang C (2018) A Multi-Sentiment Classifier Based on GRU and Attention Mechanism. In: 2018 IEEE 9th International conference on software engineering and service science (ICSESS) pages 527-530. IEEE. URL https://doi.org/10.1109/ICSESS.2018.8663799.
17. Chung J, Gulcehre C, Cho K, Bengio Y (2014) Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *NIPS 2014 Workshop on Deep Learning*. arXiv:1412.3555
18. Rathor S, Agrawal S (2021) A robust model for domain recognition of acoustic communication using Bidirectional LSTM and deep neural network. Neural Comput Appl 33(17):11223–11232. https://doi.org/10.1007/s00521-020-05569-0
19. Imaduddin H, Widyawan, FS (2019) Word Embedding Comparison for Indonesian Language Sentiment Analysis. In: 2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT), pages 426–430. IEE. https://doi.org/10.1109/ICAIIT.2019.8834536
20. Semeniuta S, Severyn A, Barth E (2016) Recurrent Dropout without Memory Loss. arXiv:1603.05118
21. Garg N, Schiebinger L, Jurafsky D, Zou J (2018) Word embeddings quantify 100 years of gender and ethnic stereotypes. Proc Nat Acad Sci 115(16):E3635–E3644. https://doi.org/10.1073/pnas.1720347115
22. McHugh Mary L (2012) Interrater reliability: the kappa statistic. Biochem Med 22(3):276–282. https://doi.org/10.11613/BM.2012.031
23. Manning C (2008) Introduction to information retrieval. Cambridge University Press, Cambridge
24. Lailiyah M, Sumpeno S, Purnama IKE (2017) Sentiment Analysis of Public Complaints Using Lexical Resources Between Indonesian Sentiment Lexicon and Sentiwordnet. In: 2017 International seminar on intelligent technology and its applications, pages 307–312, New York. IEEE Press. https://doi.org/10.1109/IALP.2018.8629262

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.