International Journal of Information System & Technology
Akreditasi No. 36/E/KPT/2019 | Vol. 5, No. 3, (2021), pp. 299-307

Sentiment Analysis of Student Emotion During Online Learning Using Recurrent Neural Networks (RNN)

Nisa Hanum Harani[1], Cahyo Prianto[2]

[1,2]Politeknik Pos Indonesia, Bandung, Indonesia

[*]nisahanum@poltekpos.ac.id

Abstract

There are many limitations in online learning process where communication effect student productivity, such as interpretation in the delivery of information can be different if it is in text form . The unstable internet network in some parts of Indonesia is also an obstacle in the learning process. Emotional factors are very influential on student motivation in learning, in online learning emotions can be read from textual dialogue in providing responses. We propose trainable model capable of identifying the tendency of emotions / responses felt by students. With using natural language processing we can extract information and insights contained in conversations from WhatsApp, then organize them into their respective categories. The selection of the RNN algorithm can increase the accuracy by 75% in analyzing student emotions in online learning .

Keywords: *Sentimen Analysis, NLP, Emotions, COVID19*

1. Introduction

The spread of Covid-19 forcing the government to implementing social distancing. This policy is intended to reduce interaction between individuals in the wider community [1] . Various sectors, especially education, must be able to start adjusting the learning process[2]. Information Technology (IT) as a learning tools has become a necessity. Learning tools must be commonly used and attractive so that it will not hinder the transfer of knowledge. With social restrictions, the habit is transform to with learning platforms and social messengers. The ICT[3] tools become an important entity for the learning process, this is an obstacle where infrastructure development not distributed equally across the archipelago [4]. With many activity online , the height of Internet use especially data-hungry apps that are usually reserved for school became difficult for students before the Ministry of Education and Culture provided a study quota program.

Each student has different characteristics in responding to online learning, some have difficulty due to limited devices and inadequate internet networks. Some feel that they focus on online learning because it is done in a conducive environment with the help of a strong network. These dynamics are interesting things to analyze. Learning involves the functioning of all organisms in the body to think, feel, observe, and behave [5]. Emotional factors are very influential on students' learning motivation [6]. Where, sadness can cause students to have difficulty focusing while studying [5], weak memory and irritability. In online learning, emotions can be read from textual dialogue in giving responses [7]. In recent years, Data Science has emerged as a new discipline. This can be seen from several disciplines that form the basis for the principles of Data Science, such as statistics, data mining, databases and distributed systems[8]. Most of the data stored in the digital world is unstructured, and organizations have problems dealing with such large amounts of data. One of the main challenges for organizations today is extracting information and value from the data stored in their information systems. Data Scientist assists organizations in turning data into value.

This study will discuss the analysis of the emotional sentiment[9][10] of students' responses from conversations in the WhatsApp messaging application using the Recurrent Neural Network (RNN) [11]algorithm. The RNN algorithm is a multi-layer neural

ISSN : 2580-7250

network that can be used to analyze sequential inputs, such as text, speech or video, for classification and prediction purposes. In addition, the RNN [12]algorithm is not limited by the input length and can use the temporal context to predict meaning better

2. Research Methodology

2.1. Method of collecting data

The research methods are processes that will be carried out during the preparation of the report and can be used as a guide in conducting research so that the results achieved do not deviate from the expected goals. The research method can be seen in Figure 1.

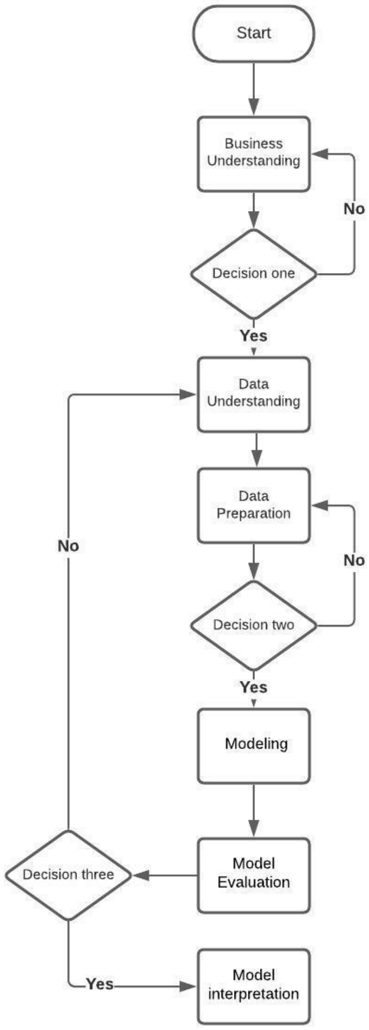

Figure 1. Research Methodology

The step of the Method can be explained as follows:

1. Business Understanding
   This stage it starts from understanding how the WhatsApp API works to take a collection of opinions from student conversations. Every class create a group in WA consisting of lecture, students and also bots. Bots must be include in a group in order to save conversation history.
2. Data Understanding
   The data is conversation history data from WhatsApp which was collected from the beginning of online learning (2019-2020). WhatsApp Group Chat Log Dataset is data retrieved by importing chat history from several WhatsApp .

International Journal of Information System & Technology
Akreditasi No. 36/E/KPT/2019 | Vol. 5, No. 3, (2021), pp. 299-307

.The following is an example of data preparation:

a) Slang Word
Change slang words and abbreviations into KBBI words. Slang words must be adapted to the data set that will be used, there are several sources or repositories but the researcher also adds some words in the repository of slang words . It depends on the common words used in everyday conversation, so each slang word in some studies may be different.

b) Data Cleaning.
In this process, symbols, characters, urls and numbers are removed

c) Convert to lowercase
In this process, uniformity is carried out on the data, where static letters will be changed to lowercase letters.

d) Remove Double word
In this process, the repeated sentences will be removed.

e) Spliting Data
This process aims to divide the cleaned datasets into trainee datasets and test datasets.

3. Decision 1.
This is an iterative process that is needed if some problem need to be solved and return to the business understanding process.
4. Modeling
The text mining approach is used to extract information textually. NLP is needed to help computers understand linguistics and sentiment analysis to understand the types of emotions that students feel during online learning. The modeling technique that will be used in this research is using the RNN algorithm.
5. Decision 2
This iteration needed to return to the process of understanding the data, An example of deepening is recognizing the characteristics of each emotion, because the spectrum of emotions is very wide.
6. Model Evaluation
The model will be evaluated using the Confusion Matrix..
7. Decision 3
This iterative process to the business understanding stage to determine whether the resulting model can answer the goals set in the business process understanding phase
8. Model Interpretation
This stage is the result of the evaluation as measured by the confusion matrix and the conclusion of the application of the NLP method using RNN.

3. Results and Discussion

3.1. Dataset Formations

The dataset collected in raw form has attributes of timestamp, number or number, message or message body, alias or name of the sender of the message, groupname or group name, Isgroup and type. The data that has been labeled by the author is 1515 data, an example of a dataset can be seen in table 1. The data understanding stage is also included in the preparation of data, changing raw data to a dataset for sentiment analysis.

Table 1. Datasets

No.	Message
1.	Saya sudah mengisi No cuma ada kesalahan pak saya pikir yang di masukan No yang aktif ternyata No wa pak mohon maaf pak
2.	Ini robot sudah disesuaikan cuma jgn menyesuaikan keinginan kamu, sekalian aja left dari grub
3.	Alhamdulillah janji kita buat prodi D4 TI sudab terkabulkan, sedikitnya pak

No.	Message
	yusril sudah senyum bahagia
4.	Soalnya VA berhasil bakalan buming nama TIK produk inovasinya semakin meraja lela
5.	lumayan panjang query nya kalu mau mencari dia mengulang atau gak dan ribet, kasian ntr iteungnya stress

3.2. Preposessing Data

The data prepossessing process is a step to complete and clean the data to maintain consistency. The prepossessing stage consists of changing lowercase letters, removing punctuation marks and replacing abbreviated words and slang words.

3.3. Spliting Data

After the preprocessing, the next process is the data split process as shown in Figure 2. The data split process is carried out using one of the functions from the sklearn library, namely train_test_split. This process aims to divide the cleaned dataset into a train dataset and a test dataset, with a percentage of 80% and 20%.

```
df_train, df_test = train_test_split(df, test_size=0.2, random_state=42, stratify=df['label'])

train_percentage = df_train['label'].value_counts(normalize=True)*100

test_percentage  = df_test['label'].value_counts(normalize=True)*100

print('TRAIN DATASET')
print('Train size:', len(df_train))
print('Persentase Train Dataset:')
print(round(train_percentage, 2))

print('=================================================================')

print('TEST DATASET')
print('Test size:', len(df_test))
print('Persentase Test Dataset:')
print(round(test_percentage, 2))
```

```
TRAIN DATASET
Train size: 1212
Persentase Train Dataset:
NEGATIF    62.54
POSITIF    37.46
Name: label, dtype: float64
=================================================================
TEST DATASET
Test size: 303
Persentase Test Dataset:
NEGATIF    62.38
POSITIF    37.62
Name: label, dtype: float64
```

Figure 2. *Spliting Data*

3.4. Word2Vec

There are several parameters used in the modeling process, namely size which is the size or dimension of the vector to be generated, then min_count which determines the minimum number of words needed in the whole corpus for words to be considered in the vocabulary or it can also be said as a parameter to determine number of frequencies. at least a word, and a window, which is a parameter used to set the context or a measure that determines the length of the word to pay attention to in the vocabulary or the range between the context word and the current word position. The model that has been created then proceed with a training process to train the word2vec model so that it can convert into vectors see Figure 3.

```
%%time
words = w2v_model.wv.vocab.keys()
vocab_size_w2v = len(words)
print("Vocab size", vocab_size_w2v)

Vocab size 234482
CPU times: user 2.24 ms, sys: 0 ns, total: 2.24 ms
Wall time: 2.89 ms
```

Figure 3. *word2vec*

3.5. Tokenize Text

In text mining, the tokenize text process to divide the words contained in a sentence, making it easier to carry out the classification process. The process of dividing words from a sentence into a token is done using one of the classes in the Keras library, the Tokenizer. The user class utility tokenizer aims to create a vector from a corpus of text into a list of binary numbers, this turns each number/integer into a value in the dictionary which then encodes the entire corpus, with the keywords in the dictionary being terms in the vocabulary itself. . Then after tokenize text is done, a padding process is also carried out which aims to equalize the size of each sentence to be classified, where the length of the tweet that is less will be filled in using the number 0. The tokenize text process can be seen in Figure 4.

```
%%time
tokenizer = Tokenizer()
tokenizer.fit_on_texts(df_train.pesan)

vocab_size = len(tokenizer.word_index) + 1
print("Total words", vocab_size)

Total words 2105
CPU times: user 86.4 ms, sys: 0 ns, total: 86.4 ms
Wall time: 87 ms
```

Figure 4. *Tokenize Text*

3.6. Label Encoding

Label encoding is a process to convert data from a category which in this study the label becomes numeric so that the category or label can be processed and understood by the machine. The label encoding process shown in Figure 5 is carried out using one of the functions contained in the sklearn library, namely the label encoder.

Label Encoder

```
[41] labels = df_train.label.unique().tolist()
     labels

     ['NEGATIF', 'POSITIF']

[43] encoder = LabelEncoder()
     encoder.fit(df_test.label.tolist())

     y_train = encoder.transform(df_train.label.tolist())
     y_test = encoder.transform(df_test.label.tolist())
```

Figure 5. *Label Encoding*

3.7. Embedding Layer

The embedding layer as can be seen in Figure 6. Embedding layer can be defined as the first hidden layer that converts the word token into a certain embedding size. There are several parameters in the embedding layer, input_dim ,the size of the vocabulary in

International Journal of Information System & Technology
Akreditasi No. 36/E/KPT/2019 | Vol. 5, No. 3, (2021), pp. 299-307

the text (vocab_size), output_dim, which is the size of the vector space in which words will be embedded (embed_dim) and input_length, which is the length of the input sequence (max_len). Then also included the weight (embedding_matrix) to initialize the existing weights on this layer.

```
[44] embed_dim = 300

     embedding_matrix = np.zeros((vocab_size, embed_dim))
     for word, i in tokenizer.word_index.items():
       if word in w2v_model.wv:
         embedding_matrix[i] = w2v_model.wv[word]
     print(embedding_matrix.shape)

     (2105, 300)

[45] embedding_layer = Embedding(vocab_size, embed_dim, weights=[embedding_matrix], input_length=max_input, trainable=False)
```

Figure 6. *Embedding Layer*

3.8. Clasification Model

This is the core process of modeling, namely the process of making a sentiment analysis classification model using the LSTM method as shown in Figure 7. LSTM is one of the variants of the RNN that was created with the aim of solving problems while at the same time covering the shortcomings of the RNN, gradient vanishing and exploding, where the LSTM hidden vector RNN method is replaced with a memory block that has a gate system, this makes LSTM able to maintain long-term memory which in principle is to train the right gating weight and has proven to be very useful and can solve problems.

```
%%time

model = Sequential()
model.add(embedding_layer)
model.add(Dropout(0.5))
model.add(LSTM(128, activation='tanh'))
model.add(Dense(1, activation='sigmoid'))
model.summary()

Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 55, 300)           631500
_________________________________________________________________
dropout (Dropout)            (None, 55, 300)           0
_________________________________________________________________
lstm (LSTM)                  (None, 128)               219648
_________________________________________________________________
dense (Dense)                (None, 1)                 129
=================================================================
Total params: 851,277
Trainable params: 219,777
Non-trainable params: 631,500
_________________________________________________________________
CPU times: user 410 ms, sys: 42.2 ms, total: 452 ms
Wall time: 665 ms
```

Figure 7. *Model Klasifikasi*

In the classification model, several layers are used including the embedding layer, then the LSTM layer with 128 neurons, then the density layer which functions as a fully connected layer, namely the layer that maps the LSTM layer output to the desired output size, besides that in the solid layer, sigmoid activation is also given. which serves to change all output values to be between 0 and 1.

International Journal of Information System & Technology
Akreditasi No. 36/E/KPT/2019 | Vol. 5, No. 3, (2021), pp. 299-307

3.9. Compile dan Training Model

The *compiling model* process shown in Figure 8, the model determine the loss function, optimizer and matrix for prediction. Loss function is a function that is intended to minimize errors contained in the model, especially models based on artificial neural networks. The model training process, where the training process for the model is configured with the data trained, x_train and y_train then batch_size or the number of samples distributed to the neural network is set to 32 and epoch, which is a process where the entire dataset has gone through the training process until it is returned to start one round in sets to 50.

Compile model

```
[47] opt = RMSprop(learning_rate=0.0001)

     model.compile(loss='binary_crossentropy',
                   optimizer=opt,
                   metrics=['accuracy'])
```

Training Model

```
%%time

history = model.fit(x_train, y_train,
                    batch_size=32,
                    epochs=50,
                    validation_split=(0.1),
                    verbose=1)
```

```
Epoch 1/50
35/35 [==============================] - 7s 133ms/step - loss: 0.6704 - accuracy: 0.5872 - val_loss: 0.6864 - val_accuracy: 0.5492
Epoch 2/50
35/35 [==============================] - 4s 114ms/step - loss: 0.6476 - accuracy: 0.6284 - val_loss: 0.6838 - val_accuracy: 0.5574
Epoch 3/50
35/35 [==============================] - 4s 113ms/step - loss: 0.6338 - accuracy: 0.6477 - val_loss: 0.6775 - val_accuracy: 0.5820
Epoch 4/50
```

Figure 8. *Training Model*

3.10. Performance Test Result

After modeling, the next step is to analyze and evaluate the modeling process. One focus of modeling evaluation is the performance of the model itself, because if the model has a good performance then the model has learned correctly using the specified parameters as well as if the model has the opposite performance. Based on the value obtained from the confusion matrix, the performance evaluations that can be measured are accuracy, recall, precision, f-measure or f1 score. The following is a confusion matrix of training data and test data shown in Figure 9

```
              precision    recall  f1-score   support

     NEGATIF       0.79      0.80      0.80       189
     POSITIF       0.67      0.65      0.66       114

    accuracy                           0.75       303
   macro avg       0.73      0.73      0.73       303
weighted avg       0.74      0.75      0.75       303

======================================================
              precision    recall  f1-score   support

     NEGATIF       0.92      0.91      0.91       758
     POSITIF       0.85      0.86      0.85       454

    accuracy                           0.89      1212
   macro avg       0.88      0.88      0.88      1212
weighted avg       0.89      0.89      0.89      1212
```

Figure 9. Classification Report

4. Conclusion

The emotional dataset of students during online learning create by collecting conversation history logs that are stored on the server. Only the message column will be processed into an emotional dataset. The Natural Language Processing (NLP) method for emotional classification with RNN reports after 50 iterations we achive test accuracy of 75%

Acknowledgement
This research was funded by Direktorat Riset dan Pengabdian Masyarakat, Direktorat Jendral Penguatan Riset dan Pengembangan which has funded this research.

References
[1] Eko Yulianto, Putri Dwi Cahyani, and Sofia Silvianita, “Perbandingan Kehadiran Sosial dalam Pembelajaran Daring Menggunakan Whatsapp groupdan Webinar Zoom Berdasarkan Sudut Pandang Pembelajar Pada Masa Pandemic COVID-19,” *JARTIKA J. Ris. Teknol. dan Inov. Pendidik.*, vol. 3, no. 2, pp. 331–341, 2020, doi: 10.36765/jartika.v3i2.277.
[2] A. Muhson, “Pengembangan Media Pembelajaran Berbasis Teknologi Informasi,” *J. Pendidik. Akunt. Indones.*, vol. 8, no. 2, 2010, doi: 10.21831/jpai.v8i2.949.
[3] N. E. Putri and D. Iskandar, “Analisis Preferensi Konsumen Dalam Penggunaan Social Messenger Di Kota Bandung Tahun 2014 (Studi Kasus : Line, Kakaotalk, Wechat, Whatsapp),” *J. Manaj. Indones.*, vol. 14, no. 2, p. 110, 2017, doi: 10.25124/jmi.v14i2.356.
[4] S. Dina, “Pemerintah ungkap tantangan pembangunan infrastruktur internet,” *kominfo.go.id*, 2020. .
[5] N. Girdharwal, “Emotional intelligence and happiness,” *Indian J. Public Heal. Res. Dev.*, vol. 10, no. 10, pp. 88–92, 2019, doi: 10.5958/0976-5506.2019.02774.8.
[6] R. Habibi, D. B. Setyohadi, and E. Wati, “Analisis Sentimen Pada Twitter Mahasiswa Menggunakan Metode Backpropagation,” *J. Inform.*, vol. 12, no. 1, pp. 103–109, 2016, doi: 10.21460/inf.2016.121.462.
[7] A. Chatterjee, U. Gupta, M. K. Chinnakotla, R. Srikanth, M. Galley, and P. Agrawal, “Understanding Emotions in Text Using Deep Learning and Big Data,” *Comput. Human Behav.*, vol. 93, pp. 309–317, 2019, doi: 10.1016/j.chb.2018.12.029.
[8] W. Van der Aalst, “Process mining: Data science in action,” *Process Min. Data Sci. Action*, no. April 2014, pp. 1–467, 2016, doi: 10.1007/978-3-662-49851-4.
[9] James F. Allen, “Natural language processing,” *Encycl. Comput. Sci.*, pp. 1218–1222, 2003.
[10] A. Yadollahi, A. G. Shahraki, and O. R. Zaiane, “Current state of text sentiment analysis from opinion to emotion mining,” *ACM Comput. Surv.*, vol. 50, no. 2, 2017, doi: 10.1145/3057270.
[11] H. Shi, M. Xu, and R. Li, “Deep Learning for Household Load Forecasting-A Novel Pooling Deep RNN,” *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 5271–5280, 2018, doi: 10.1109/TSG.2017.2686012.
[12] W. Yu, I. Y. Kim, and C. Mechefske, “An improved similarity-based prognostic algorithm for RUL estimation using an RNN autoencoder scheme,” *Reliab. Eng. Syst. Saf.*, vol. 199, no. February, p. 106926, 2020, doi: 10.1016/j.ress.2020.106926.

Authors

1st Author
Nisa Hanum Harani
Lecture of Politeknik Pos Indonesia,Bandung,Indonesia
nisahanum@poltekpos.ac.id

2nd Author
Cahyo Prianto
Lecture of Politeknik Pos Indonesia,Bandung,Indonesia
cahyoprianto@poltekpos.ac.id