Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2019) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

4th International Conference on Computer Science and Computational Intelligence 2019 (ICCSCI), 12-13 September 2019

Text Generator Application In Assisting Mental Health Care

Kevin Lotan[a], Franky Desmarets[b], David Christopher[c]

[a]Computer Science Department, School of Computer Science, Bina Nusantara University, Jl. K. H. Syahdan No. 9, DKI Jakarta, 11480, Indonesia
[b]Computer Science Department, School of Computer Science, Bina Nusantara University, Jl. K. H. Syahdan No. 9, DKI Jakarta, 11480, Indonesia
[c]Computer Science Department, School of Computer Science, Bina Nusantara University, Jl. K. H. Syahdan No. 9, DKI Jakarta, 11480, Indonesia

Abstract

Text Generator Application in Assisting Mental Health Care, this application is built by using Convolution Neural Network as its emotion classifier, and then using random output selector from the prepared data to give the output to the user. This application was built to help people to stabilize their mental healthiness by 'tweeting' messages to the generator and then they can get a direct reply from it which is quite helpful to ease their ominous feeling.

Keywords: Convolution Neural Network; Text Generator; Mental Health;

1. Introduction

Mental health problems are serious in Indonesia. There has been previous research by Riset Kesehatan Dasar, that in East Java alone, they got 2.2 per mil prevalence of severe mental illness like schizophrenia and other psychotic disorders. This number is significantly higher than the national prevalence in Indonesia, which is 1.7 per mil. [1]

Mental illness can occur from shock, and mostly people in Indonesia will call people that have mental illness issue crazy people, and the people who have mental illness issues can be more damaged. We make this application to help the people that have mental illness issues to solve their problems, by telling them stories, and we give them some kind of motivational words to enlighten their mood, because people with mental illness issues need support to help them fight with their mental illness issues.

* Corresponding author. Tel.: +62-8779-2388-855;
E-mail address: kevin.lotan@binus.ac.id

People with mental illness are struggling to overcome their illness. Since the number of the counselors is limited, we are trying to help people that struggle with their mental illness to feel that they have someone that they can talk to and get replies with some words that they want to see. We want to lighten their mood by giving them a text or some picture with quotes.

2. Previous Work

There has been previous work from [2] about creating a chatbot by extracting keywords using NLP. But their work is about a simple chatbot to ask several questions to clear the command text from the users. Here we are creating a more advanced chatbot that can text and reply based on the mental condition of the users. For extracting the insight from text to help to detect mental issues, [3] using Natural Language Processing to get the dataset from social media. They used De Choudhury et al. 's (2013b) built a classifier for determining whether a Twitter post indicates depression. Then trained a Support Vector Machine (SVM) to classify the tweets into the four categories (happy, no distress, low distress, and high distress) and found F1-measures of 0.4–0.6 (the harmonic mean of precision and recall). While (Jackson et al., 2017) in his paper used Natural Language Processing to extract information from clinical records to process large quantities of unstructured text and return its meaning. [5] tried to take the most important keyword in a text using TF-IDF (Term Frequency-Inverse Document Frequency), to then predict the best counseling message using Hidden Markov Model.

In understanding the meaning of the text, we have taken some examples of how to do that. The most popular way is by using Artificial Neural Network, [6] use Convolutional Neural Network to classify the text into thirteen conditions: depression, bipolar disorder, psychosis, general anxiety disorder (GAD), panic disorder, anxiety spectrum disorders, obsessive-compulsive disorders (OCD), obsessive-compulsive spectrum disorder (OCSD), attention deficit hyperactivity disorder (ADHD), post-traumatic stress disorder (PTSD), eating disorders, dementia, and complicated grief. Then we found a journal that uses GA-ANN by [7] that might be helpful in classifying the problem of mental issues.

To do the text generation, we can use some LSTM-based network, which is shown by [8] in their comparing journal. The other way which is more preferred by us is by using GPT-2 that is being used by [9] in their patent claim text generation or [10] work in creating a task-oriented dialogue system.

Based on current state-of-the-art of NLP models, there is GPT-2 [11], which is a Transformer based model with the largest version contains 1.5 Billion parameters that achieves state-of-the-art results on 7 out of 8 tested datasets in a zero-shot setting and showed proficiency in text generation test. Besides GPT-2 there is also the BERT model [12]. BERT is a bidirectional encoder representation from transformers. BERT achieves state-of-the-art results on 11 NLP tasks, with a high F1 score on SQuAD, which is a question answering dataset. This shows that BERT is very good at answering tasks.

GPT-2 [11] and BERT [12] is based on Transformer architecture [13], which is an architecture that relies entirely on an attention mechanism to draw global dependencies between input and output. Attention can be described as mapping query, keys, and value to an output, where the output is a weighted sum of the value and the weight assigned to each value is computed by the compatibility function of the query with the corresponding key. From research, it can be seen that GPT-2 is the best model to do text generation tasks. But, the GPT-2 model is very large and not practical to be used for building applications intended for mobile use.

DistillBERT [14] is a model based on BERT [12], created by distilling [15] the base version of BERT. Distillation is the process of compressing the knowledge in an ensemble into a single model which can reduce size and improve performance of machine learning algorithms.

DistillBERT [14] is significantly smaller and faster that original BERT [12] while still maintaining 97% of language understanding capabilities of original BERT.

There is also DistillGPT2 which is a GPT2 version of DistillBERT that can have high proficiency of text generation which can be run effectively on mobile devices.

3. Methodology

3.1. Emotion Extraction

3.1.1. Preprocessing

Natural Language Processing has been the front most task in Artificial Intelligence. Natural Language Processing is a branch of artificial intelligence that deals with the interaction between humans and computers using the natural language. Since we are creating a chat bot, we will need the computers to first understand what the user is currently feeling, through the message they are texting. Preprocessing in Natural Language Processing will be used here to help computers gain insight from the text. Gaining insight from text is called Natural Language Understanding. This can be achieved by first preprocessing the text input to reduce the ambiguity in a sentence and then we will feed the model with richer information.

Preprocessing is done in the following way:

- Case Folding
- Noise Removal
- Stemming
- Tokenization
- Removing Stop Words

3.1.2. Convolution Neural Network Model

The model we used will be CNN (Convolution Neural Network) to gain emotion from the preprocessed text by using NLP. CNN is Convolution Neural Network, CNN has been proved to reduce classification error from 26% to 15%. CNN uses ConvNet instead of a feed-forward pass. The ConvNet is able to successfully capture the spatial and temporal dependencies in an image through sets of relevant filters. The architecture of ConvNet performs a better fitting to the image datasets due to the reduction in the number of parameters involved and the reusability of weights. In this model we will create a Convolution Neural Network that consist of an embedding layer, a convolution layer, a dropout 20% layer, 2 dense layers of 128 and 64 neurons and then a dense layer of 14 layers with softmax activation function, each for an output class. The convolution neural network is created from 256 neurons, with activation function relu, strides number 1, kernel size 3 and then a max pooling layer with pool size of 2. The learning rate is 0.00075 with 10 epochs.

```
[9]: model = tf.keras.Sequential([
         tf.keras.layers.Embedding(max_features, embedding_dims, input_length=max_text_length),
         tf.keras.layers.Conv1D(256, kernel_size=(3), padding='same', activation='relu', strides=1),
         tf.keras.layers.MaxPooling1D(pool_size=(2)),
         tf.keras.layers.Dropout(0.2),
         tf.keras.layers.Flatten(),
         tf.keras.layers.Dense(128, activation='relu'),
         tf.keras.layers.Dense(64, activation='relu'),
         tf.keras.layers.Dense(len(list_of_classes), activation='softmax')
     ])
```

```
[10]: optim = tf.keras.optimizers.Adam(
          learning_rate=0.00075,
          beta_1=0.9,
          beta_2=0.999,
          epsilon=1e-07,
          amsgrad=False,
          name="Adam"
      )
```

Fig. 1. Model written in python

3.2. Text Generator

After we get the insight from the text, there are 14 classes of emotion. The output from sentiment classifier from the text will output one from these classes. Each of classes will have 3 sentence of output, we will randomize the output from them.

4. Experiments Result

In this experiment, we used data from [16]. This data is basically scrapes of twitter's comments and post. They got labeled and then will be used in training and testing the model.

4.1. First Model Experiment

In this experiment, we tried putting this data below into the CNN model.

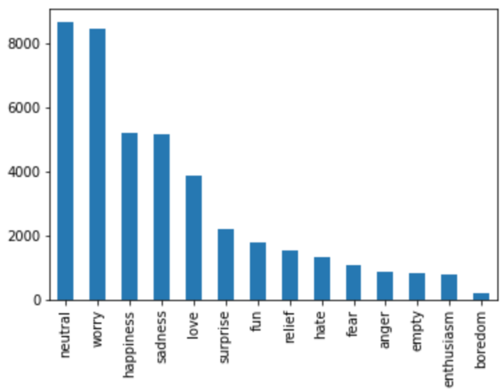

Fig. 2. Data 1

After the 10 epoch training process, The CNN model has 87% accuracy on the training data and then we tried to check the prediction output manually with some anger speech like for example 'Who did this?', this should output anger but the biggest number in the output still pointing at neutral, this model got over fit and the anger category is in the third biggest. Every text inputted will get either worry or neutral as the first output.

```
[193]: text = ['Who did this?']

[194]: tokenizer.fit_on_texts(list(text))
       x_tokenized = tokenizer.texts_to_sequences(text)
       text = sequence.pad_sequences(x_tokenized, maxlen=max_text_length)

[195]: y_pred = model.predict(text)

[196]: def pred(y_pred):
           y_pred = np.array(y_pred)
           global list_of_classes
           o = -1
           o2 = -1
           o3 = -1
           k = 0

           for i in range(len(list_of_classes)):
               if y_pred[0][i] > k:
                   o3 = o2
                   o2 = o
                   o = i
                   k = y_pred[0][i]

           return [list_of_classes[o], list_of_classes[o2], list_of_classes[o3]]

       print(pred(y_pred))
       ['neutral', 'empty', 'anger']
```

Fig. 3. Prediction using Trained Model 1

4.2. Second Model Experiment

To reduce over fitting, we tried to reduce the number of oversized data like neutral, worry, etc to at most 1000 data into the same model.

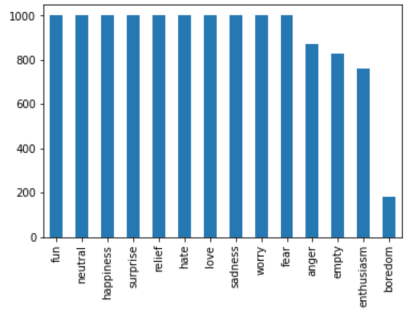

Fig. 4. Data 2

In this experiment, we split the training data into 80% training data and 20% testing data to help us compute the accuracy. After another 10 epoch training, the new model got 95% accuracy on training data. To test the prediction from the model, we used the test data to create a confusion matrix.

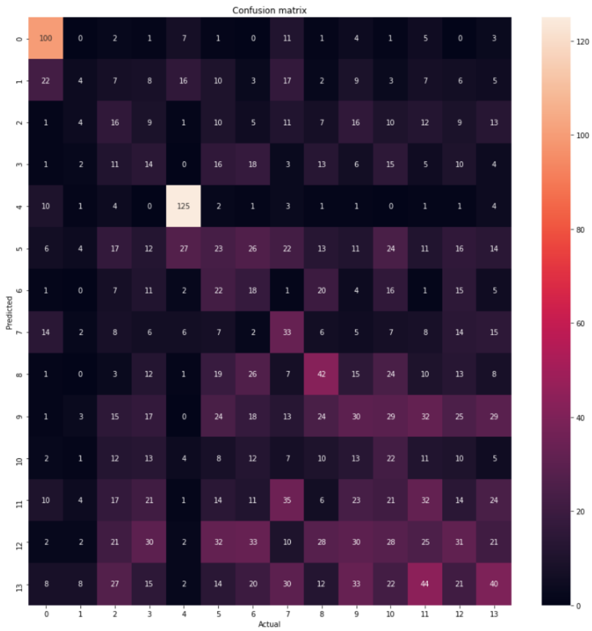

Fig. 5. Confusion Matrix on Second Trained Model

From the test data, we got 20.9% accuracy. We will continue to use this model as our prediction model to generate text because this sentiment analysis will take a lot of time to perfect.

4.3. User Interface for Text Generator

We created a simple GUI by using tkinter from python library. This GUI has a text box for input, a generate button and another text box for the generated output.

Fig. 6. Graphical User Interface

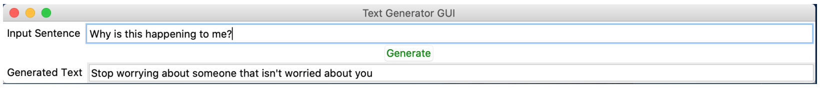

Fig. 7. Graphical User Interface 2

5. Conclusion and Future Works

This Text Generator is a great idea to help people talk to someone to help ease their ominous mental state or maybe learning to reply someone. But in terms of this machine accuracy, this still produces bad result. First the emotion classifier only has 20% accuracy, this model is prone to over fitting because its accuracy on training data is pretty high and then the text generator is lazily built by randomizing the prepared text based on the class. In the future, we will try to increase the classifier accuracy and then maybe using machine learning to generate the text, so that we wont see the same output over and over again. This will increase the machine accuracy overall and can really help people to gain experience in talking with someone.

References

[1] Hartini, N., Fardana, N. A., Ariana, A. D., Wardana, N. D. (2018). Stigma toward people with mental health problems in Indonesia. Psychology Research and Behavior Management, 11, 535–541. https://doi.org/10.2147/PRBM.S175251

[2] Baby, C. J., Khan, F. A., Swathi, J. N. (2017). Home automation using IoT and a chatbot using natural language processing. 2017 Innovations in Power and Advanced Computing Technologies, i-PACT 2017, 2017-Janua, 1–6. https://doi.org/10.1109/IPACT.2017.8245185

[3] Calvo, R. A., Milne, D. N., Hussain, M. S., Christensen, H. (2017). Natural language processing in mental health applications using non-clinical texts. Natural Language Engineering, 23(5), 649–685. https://doi.org/10.1017/S1351324916000383

[4] Jackson, R. G., Patel, R., Jayatilleke, N., Kolliakou, A., Ball, M., Gorrell, G., Roberts, A., Dobson, R. J., Stewart, R. (2017). Natural language processing to extract symptoms of severe mental illness from clinical text: The Clinical Record Interactive Search Comprehensive Data Extraction (CRIS-CODE) project. BMJ Open, 7(1), 1–10. https://doi.org/10.1136/bmjopen-2016-012012

[5] Althoff, T., Clark, K., Leskovec, J. (2016). Large-scale Analysis of Counseling Conversations: An Application of Natural Language Processing to Mental Health. Transactions of the Association for Computational Linguistics, 4, 463–476. https://doi.org/10.1162/tacl_a_00111

[6] Tran, T., Kavuluru, R. (2017). Predicting mental conditions based on "history of present illness" in psychiatric notes with deep neural networks. Journal of Biomedical Informatics, 75, S138–S148. https://doi.org/10.1016/j.jbi.2017.06.010

[7] Jahed Armaghani, D., Hasanipanah, M., Mahdiyar, A., Abd Majid, M. Z., Bakhshandeh Amnieh, H., Tahir, M. M. D. (2018). Airblast prediction through a hybrid genetic algorithm-ANN model. Neural Computing and Applications, 29(9), 619–629. https://doi.org/10.1007/s00521-016-2598-8

[8] Luppescu, G., Romero, F. (2017). Comparing Deep Learning and Conventional Machine Learning for Authorship Attribution and Text Generation. Stanford: Stanford University Press, 1–9.

[9] Lee, J.-S., Hsiang, J. (2019). Patent Claim Generation by Fine-Tuning OpenAI GPT-2. 1–11. http://arxiv.org/abs/1907.02052

[10] Budzianowski, P., Vulić, I. (2019). Hello, It's GPT-2 - How Can I Help You? Towards the Use of Pretrained Language Models for Task-Oriented Dialogue Systems. 15–22. https://doi.org/10.18653/v1/d19-5602

[11] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I. (2018). Language Models are Unsupervised Multitask Learners.

[12] Devlin, J., Chang, M. W., Lee, K., Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, 1(Mlm), 4171–4186.

[13] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 2017-December(Nips), 5999–6009.

[14] Sanh, V., Debut, L., Chaumond, J., Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. 2–6. http://arxiv.org/abs/1910.01108

[15] Hinton, G., Vinyals, O., Dean, J. (2015). Distilling the Knowledge in a Neural Network. 1–9. http://arxiv.org/abs/1503.02531

[16] Mankutimma, text-emotion-classification, 2018, Retrived 11 June 2020 from https://github.com/mankutimma/text-emotion-classification/tree/master/data