The International Society of Data Scientists
Journal of Data Science and Artificial Intelligence
Vol. 1, No. 1. Online ISSN: 2831-4794
Available at https://www.isods.org/publication/

Application of deep-learning for medical text analysis: a comparison of different types of recurrent neural network

Cuong Do*[a], Lan Vu*[b]
[a]Department of Mathematics, University of Connecticut, 341 Mansfield Road U1009, Storrs, Connecticut 06269-1009.
[b]Department of Epidemiology, Hanoi University of Public Health, 1A Duc-Thang road, North Tu-Liem District, Hanoi, Vietnam

ABSTRACT

We investigate the application of Deep Learning in the healthcare settings to automatically categorize symptoms using patient transcriptions. Several types of networks are investigated, including using stacked unidirectional and bidirectional long short-term memory networks and gated recurrent unit networks, and combinations of these. The approach obtains high accuracy, precision, and recall. Specifically, the stacked LSTM model, stacked bidirectional GRU model, and the stacked bidirectional LSTM model perform equally well in term of accuracy, while that of the stacked GRU model is slightly lower. These well-performing models could be applied for automatic symptom categorization based on patient transcriptions.

Keywords: Recurrent Neural Networks, Long Short-Term Memory, Gated Recurrent Unit, TensorFlow

***First Authors,** E-mail: cuong.do@engineer.uconn.edu, vhl@huph.edu.vn
*Cuong Do and Lan Vu contributed equally to this manuscript

INTRODUCTION

Electronic health records (EHRs) contain useful information for providing better care for patients. Most of data in EHRs is unstructured text (such as physician notes, nursing notes, clinical protocols, or medical transcriptions). One of the key challenges for healthcare analytics is to deal with huge data volumes in the form of unstructured text[1]. Text Analytics, or Text Mining, is generally defined as the methodology followed to derive quality and actionable insights from textual data.[2] The use of efficiently automatic text analysis tools can provide many benefits for health organization in a wide range of applications. Machine learning can facilitate rapid development of medical text analysis tools by leveraging large amounts of text data. Previous studies had shown the possibility of applying recurrent neural network (RNN), one of deep learning methods, for auto-classification of diseases or auto-prediction of risk based on text analysis of patient health records. For instance, Liu Cheng et al., showed that a four-layer CNN could improve the performance of baselines for predicting chronic obstructive pulmonary disease and congestive heart failure significantly[3]. A deep dynamic network using RNNs with long short-term memory (LSTM) hidden units, word embedding, and pooling was used in DeepCare for predicting future medical outcomes[4]. Using a three-layer Stacked Denoising Autoencoder (SDA), Miotto et al., proposed deriving deep patient representations from EHRs before a random forest classifier. It was found that the approach produced significantly better predictions than raw EHRs or conventional approaches (e.g. PCA, k-means)[5]. Using a deep architecture based on RNNs, it was also possible to automatically de-identify free-text summary of patients by removing protected health information from clinical notes[6].

RNNs are a class of neural networks that are powerful in modeling sequence data. Because of their internal memory, RNN can provide a much deeper understanding of a sequence and its context compared to other algorithms. However, RNNs have two problem, short-term memory and vanishing gradients. These problem was addressed by two variants of RNNs, Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU). LSTM and GRU, which model the hidden state

with cells in order to determine what to keep in and what to erase from memory given the previous state, the current memory, and the input value.[7,8] Natural Language Processing applications have greatly benefited from these variants as they capture long-term dependencies reliably. This study compares the use of different variants of RNNs, including stacked gated recurrent units (GRU) model, long short-term memory (LSTM), as well as their bidirectional versions on auto-clasification of medical unstructured data.

RECURRENT NEURAL NETWORKS (RNN), LSTM, AND GRU

Recurrent neural networks (RNN) are widely used for solving tasks related to involving temporal sequence, including speech, text, audio, or time series[7]. RNNs are characterized by the backward connections in a hidden layer, allowing the network with short-term memory capability.

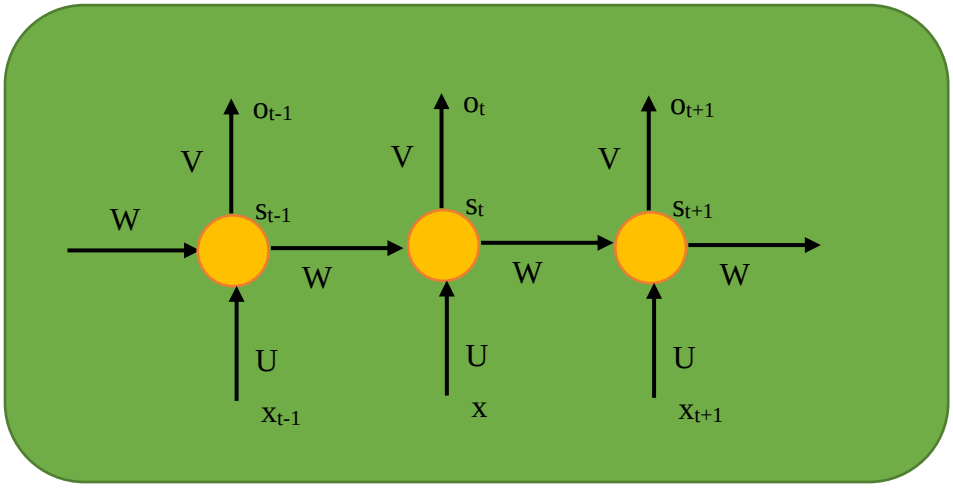


Figure 1: RNN Architecture

In the RNN architecture as shown in the figure, the input sequence $\{x_1, x_2, \ldots, x_n\}$. The initial state vector is s_0. After observing inputs, the state vectors an ordered list of state vectors $\{s_1,s_2 \ldots s_n\}$ with its corresponding output vectors $\{o_1,o_2 \ldots o_i\}$ are returned. V is output layer weight vector, and U is hidden layer weight vector. W is a shared weight vector across all time steps. The previous state, s_{t-1}, contributes to the current hidden state s_t as follows:

$$s_t = f(U * x_t + W * s_{t-1} + b) \quad (1)$$

The output sequence, o_t, as the function of the current state, where each element is dependent on all previous x_{t-1}

$$o(s_t) = softmax(V * s_t) \quad (2)$$

RNN has attained satisfactory performance in many different tasks, including language modeling and machine translation. However, there are several issues such as vanishing gradient and inefficiency to manipulate long sequences, which leads to different variants of RNN for improvement. Long Short-Term Memory (LSTM) overcomes vanishing gradient problem in classical RNNs, in which the gradients grow or decrease exponentially when being introduced with long input sequences[9]. Figure 2 presents the architecture of LSTM gates.

There are three additional gates compared to a vanilla RNN: The update gate u (also called the input gate), forget gate f and output gate o. Gates are neurons using a sigmoid activation, multiplied with the output of other neurons. At each time step, previous cell state c_{t-1} and previous hidden state h_{t-1} are used to calculate the current cell state c_t, and hidden state h_t as follows:

$$u_t = \sigma(W_u x_t + U_u h_{t-1} + b_u) \quad (3)$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (4)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (5)$$

$$\tilde{c}_t = tanh(W_c\, x_t + U_c\, h_{t-1} + b_c) \quad (6)$$

$$c_t = u_t * \tilde{c}_t + f_t * c_{t-1} \quad (7)$$

$$h_t = o_t * \tanh(c_t) \quad (8)$$

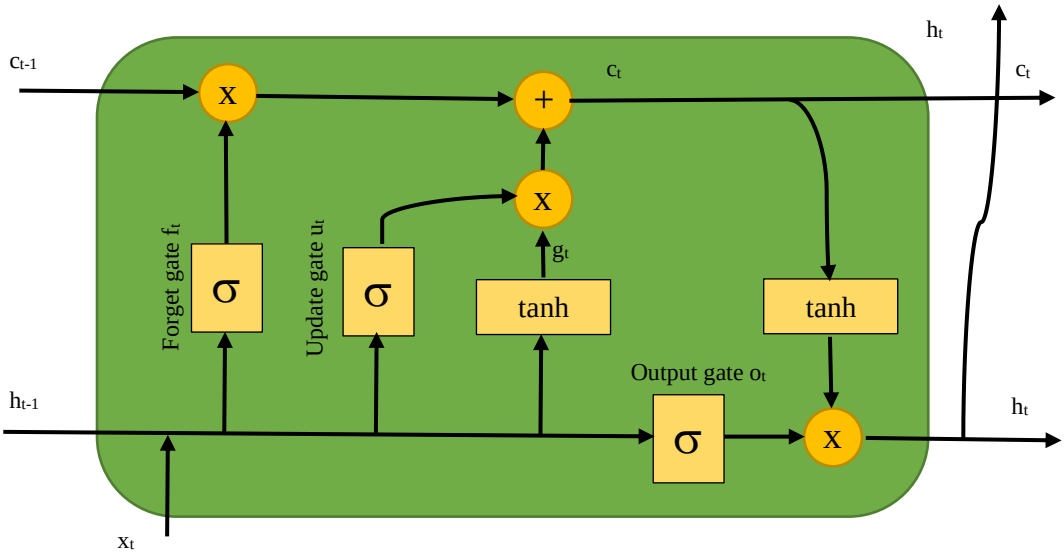


Figure 2: LSTM Gates

Where * is element-wise multiplication, σ is sigmoid function, W, U, and b are the parameters to be trained. Memory components c_t represents the current state at time-step t. The values of gates are based on the previous h_{t-1} and the current input x_t. The candidate value $\tilde{c}_t$ is used to update the memory c_t by the amount controlled by the update gate. The forget gate f and update gate u are used such that f is used to choose the amount of information to be held, and u is to choose the amount of information to be updated. The current hidden state h_t is calculated by from c_t with a *tanh* activation, controlled by the output gate o_t.

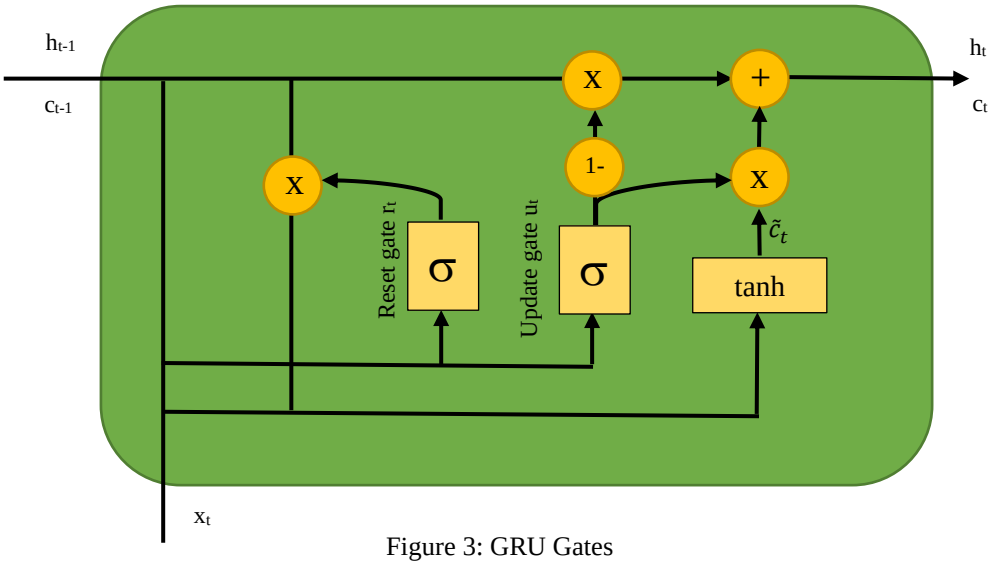


Figure 3: GRU Gates

Vanilla RNNs do not work well with long sequences due to the vanishing gradient problem. With the gating mechanism, the gradient is kept high for long time-steps, and thus overcome the problem of vanishing gradient. Gated recurrent units (GRU) are like LSTMs with a forget gate however have fewer parameters as there is no output gate, so it is faster (Figure 3). In additional to the vanishing gradient problem, there is exploding gradient problem, which could be solved using gradient clipping[8].

The GRU equations are given as follows:

$$u_t = \sigma(W_u\, x_t + U_u\, h_{t-1} + b_u) \tag{9}$$

$$r_t = \sigma(W_r\, x_t + U_r\, h_{t-1} + b_r) \tag{10}$$

$$\tilde{c}_t = tanh(W_c\, x_t + U_c\, (r_t * c_{t-1}) + b_c) \tag{11}$$

$$c_t = u_t * \tilde{c}_t + (1 - u_t) * c_{t-1} \tag{12}$$

There are several different architectures used in RNNs, including one-to-one, one -to-many, many-to-one, and many-to-many. A sentiment analysis model uses the one-to-one architecture, while the one-to-many may be used for music generation. A many-to-many architecture is used in machine translation, which includes an encoder and a decoder. The encoder encodes the input sequence into an output representation and feed it to the decoder to generate the outputs.

Bidirectional RNNs with LSTM, or GRU are commonly used to overcome the limitation of for not being able to learn from elements later in the sequence. The disadvantage is that the entire sequence is needed before it can be processed. In simple models, often times one layer of RNNs suffices. In many other problems, it is common seeing stacks of several RNN layers to make a deeper network, and seeing combination of layer types, such as 1 dimensional convolutional layer, and dense layers. Sequence models can be augmented using an attention mechanism, allowing the model to understand where the attention should be given a sequence of inputs.

DATASET

The dataset is developed by Appen containing text phrases that describe symptoms, and symptom prompts, as well as audio associated with the text phrases[10]. In this paper we use the text data of the dataset for modeling.

Figure 4 below shows the word cloud of the phrases, in which stop words are removed. The word sizes represent different frequencies. It could be seen that the words that are related to the body part, feelings, pain are more commonly visual than others. Figure 5 shows the top common words from the phrase text.

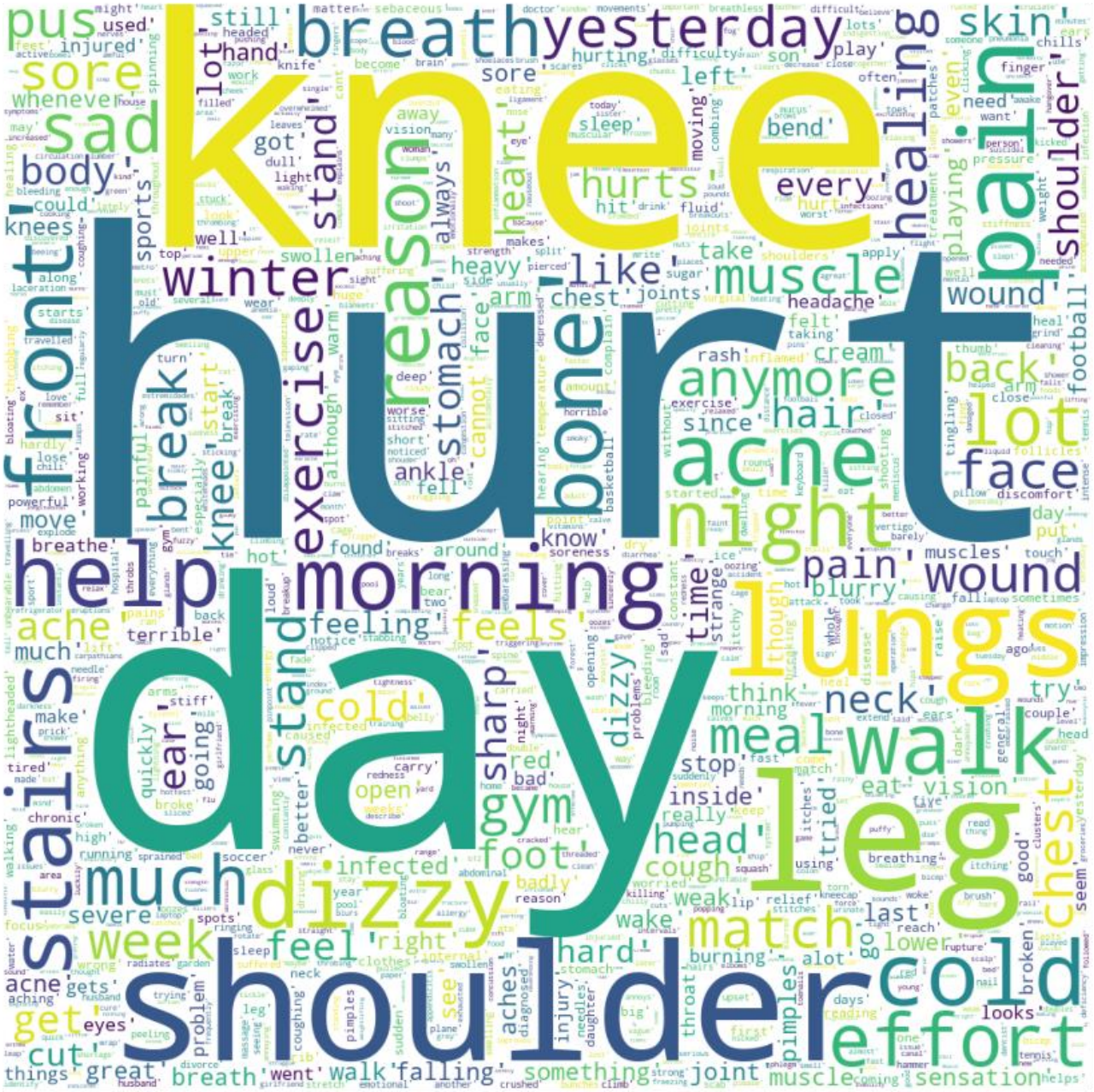


Figure 4. The word cloud of the phrase text in the dataset

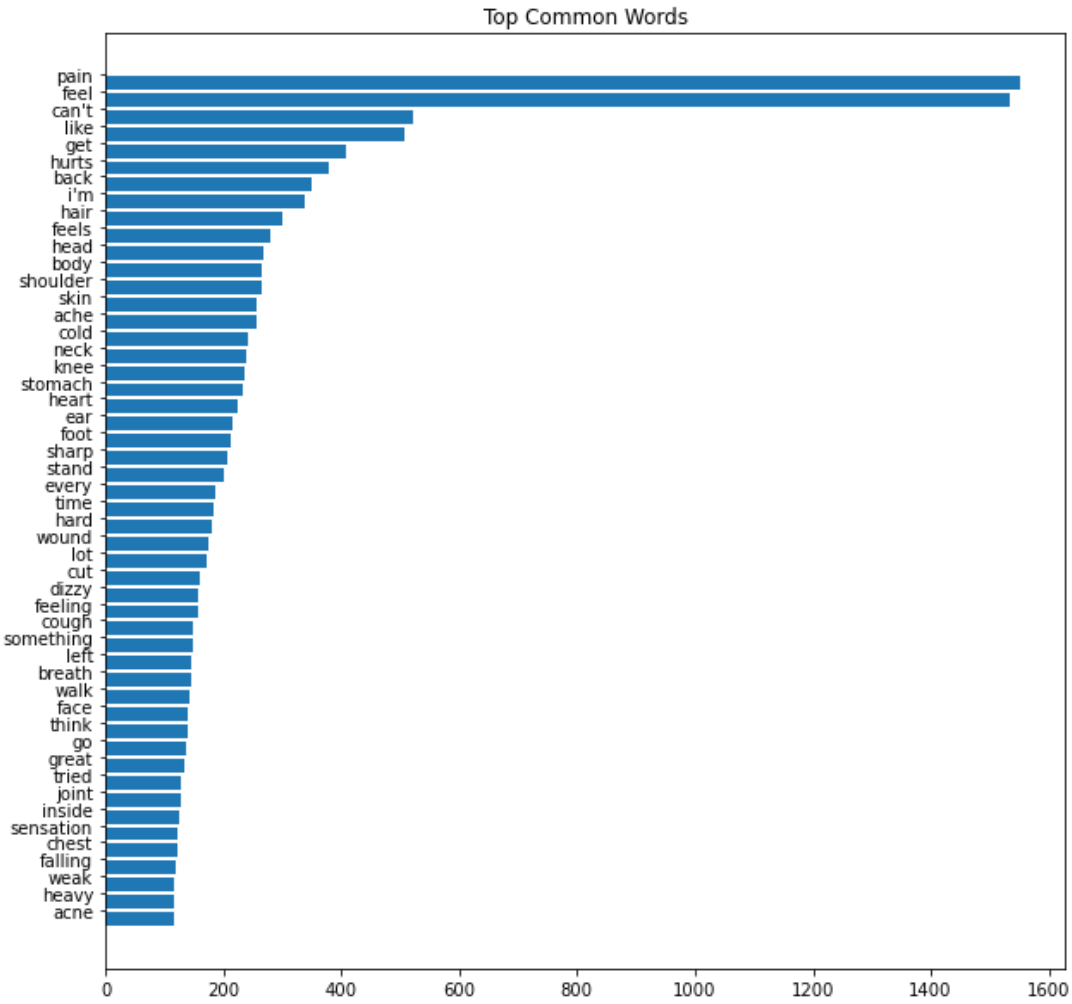


Figure 5. The top common words of the phrase text in the dataset

MODDELING APPROACH

The transcription phrase text is tokenized and converted to sequence of integers and padded to a fixed length. Data is split such that 20% of data is set aside for testing the trained model, while the rest of 80% is then further split such that 80% is used for training, and 20% is used to validate the model during training.

A vocabulary size of 2000, and a vector space of 64 dimensions are selected in the embedding layer, followed by two layers of GRU or LSTM of 128 units stacked on top of each other, followed by a dense layer with softmax activation and 25 outputs, which are the number of classes.

Bayesian optimization is applied to direct a search, which builds a probabilistic model of the function mapping from hyperparameter values to the objective evaluated on a validation set. The model is fit on the training data and validated on the validation data using the best hyperparameter set and stop early callback to find the best number of epochs. The model is finally refit using the best hyperparameter set on all the available training and validation data for the given best number of epochs.

RESULTS AND DISCUSSION

From Table 1, it could be seen that the stacked GRU model obtains a high accuracy 99.55% with hyperparameter tuning. The stacked LSTM model however provides a higher accuracy of 99.62%. The stacked bidirectional LSTM and stacked bidirectional GRU models reach the same high accuracy of 99.62% as the stacked LSTM model. Table 2 provides the details for precision, recall, and F1-score. We can observe that these metrics are high for all the classes, at least 98%.

Table 1. Model accuracy

Models	Accuracy
Stacked GRU	99.55%
Stacked LSTM	99.62%
Bidirectional Stacked GRU	99.62%
Bidirectional Stacked LSTM	99.62%

Table 2. Model precision, recall, and F1-score of the stacked GRU, stacked LSTM, stacked bidirectional GRU, and stacked bidirectional LSTM models

	precision	recall	f1-score	support
Acne	0.99	1.00	0.99	67
Back pain	0.98	1.00	0.99	46
Blurry vision	1.00	1.00	1.00	45
Body feels weak	1.00	0.98	0.99	43
Cough	1.00	1.00	1.00	51
Ear ache	1.00	1.00	1.00	64
Emotional pain	0.98	1.00	0.99	49
Feeling cold	1.00	1.00	1.00	52
Feeling dizzy	1.00	1.00	1.00	63
Foot ache	1.00	0.98	0.99	42
Hair falling out	0.98	0.98	0.98	61
Hard to breath	1.00	1.00	1.00	54
Head ache	1.00	1.00	1.00	51
Heart hurts	1.00	0.98	0.99	45
Infected wound	1.00	1.00	1.00	70
Injury from sports	1.00	1.00	1.00	39
Internal pain	1.00	1.00	1.00	47
Joint pain	1.00	1.00	1.00	61
Knee pain	1.00	1.00	1.00	64
Muscle pain	0.98	1.00	0.99	52
Neck pain	1.00	1.00	1.00	51
Open wound	1.00	1.00	1.00	40
Shoulder pain	1.00	0.98	0.99	63
Skin issue	0.98	0.98	0.98	61
Stomach ache	1.00	1.00	1.00	52

	precision	recall	f1-score	support
Acne	1.00	1.00	1.00	67
Back pain	0.98	1.00	0.99	46
Blurry vision	1.00	1.00	1.00	45
Body feels weak	1.00	0.98	0.99	43
Cough	0.98	1.00	0.99	51
Ear ache	1.00	1.00	1.00	64
Emotional pain	1.00	1.00	1.00	49
Feeling cold	1.00	1.00	1.00	52
Feeling dizzy	1.00	1.00	1.00	63
Foot ache	1.00	0.98	0.99	42
Hair falling out	0.98	0.98	0.98	61
Hard to breath	1.00	1.00	1.00	54
Head ache	1.00	1.00	1.00	51
Heart hurts	1.00	0.98	0.99	45
Infected wound	1.00	1.00	1.00	70
Injury from sports	1.00	1.00	1.00	39
Internal pain	1.00	1.00	1.00	47
Joint pain	1.00	1.00	1.00	61
Knee pain	1.00	1.00	1.00	64
Muscle pain	0.98	1.00	0.99	52
Neck pain	1.00	1.00	1.00	51
Open wound	1.00	1.00	1.00	40
Shoulder pain	1.00	0.98	0.99	63
Skin issue	0.98	1.00	0.99	61
Stomach ache	1.00	1.00	1.00	52

	precision	recall	f1-score	support
Acne	1.00	1.00	1.00	67
Back pain	0.98	1.00	0.99	46
Blurry vision	1.00	1.00	1.00	45
Body feels weak	1.00	0.98	0.99	43
Cough	1.00	1.00	1.00	51
Ear ache	1.00	1.00	1.00	64
Emotional pain	0.98	1.00	0.99	49
Feeling cold	1.00	1.00	1.00	52
Feeling dizzy	1.00	1.00	1.00	63
Foot ache	1.00	0.98	0.99	42
Hair falling out	0.98	0.98	0.98	61
Hard to breath	1.00	1.00	1.00	54
Head ache	1.00	1.00	1.00	51
Heart hurts	1.00	0.98	0.99	45
Infected wound	1.00	1.00	1.00	70
Injury from sports	1.00	1.00	1.00	39
Internal pain	1.00	1.00	1.00	47
Joint pain	1.00	1.00	1.00	61
Knee pain	1.00	1.00	1.00	64
Muscle pain	0.98	1.00	0.99	52
Neck pain	1.00	1.00	1.00	51
Open wound	1.00	1.00	1.00	40
Shoulder pain	1.00	0.98	0.99	63
Skin issue	0.98	1.00	0.99	61
Stomach ache	1.00	1.00	1.00	52

	precision	recall	f1-score	support
Acne	1.00	1.00	1.00	67
Back pain	0.98	1.00	0.99	46
Blurry vision	1.00	1.00	1.00	45
Body feels weak	0.98	0.98	0.98	43
Cough	1.00	1.00	1.00	51
Ear ache	1.00	1.00	1.00	64
Emotional pain	1.00	1.00	1.00	49
Feeling cold	1.00	1.00	1.00	52
Feeling dizzy	1.00	1.00	1.00	63
Foot ache	1.00	0.98	0.99	42
Hair falling out	0.98	0.98	0.98	61
Hard to breath	1.00	1.00	1.00	54
Head ache	1.00	1.00	1.00	51
Heart hurts	1.00	0.98	0.99	45
Infected wound	1.00	1.00	1.00	70
Injury from sports	1.00	1.00	1.00	39
Internal pain	1.00	1.00	1.00	47
Joint pain	1.00	1.00	1.00	61
Knee pain	1.00	1.00	1.00	64
Muscle pain	0.98	1.00	0.99	52
Neck pain	1.00	1.00	1.00	51
Open wound	1.00	1.00	1.00	40
Shoulder pain	1.00	0.98	0.99	63
Skin issue	0.98	1.00	0.99	61
Stomach ache	1.00	1.00	1.00	52

In Figure 6(a) on the left, observing the confusion matrix, it could be seen that the stacked GRU model works well as most the non-zero counts are on the diagonal. We could observe the six misclassifications, where “body feels weak” was misclassified as “hair falling out”, “foot ache” was misclassified as “skin issue”, “hair falling out” was misclassified as “back pain”, “heart hurts” was misclassified as “cough”, “shoulder pain” was misclassified as “muscle pain”, and “skin issue” was misclassified as “acne”. In Figure 6(b) on the right, the stacked LSTM model was able to correctly classify “skin issue” compared to the stacked GRU model.

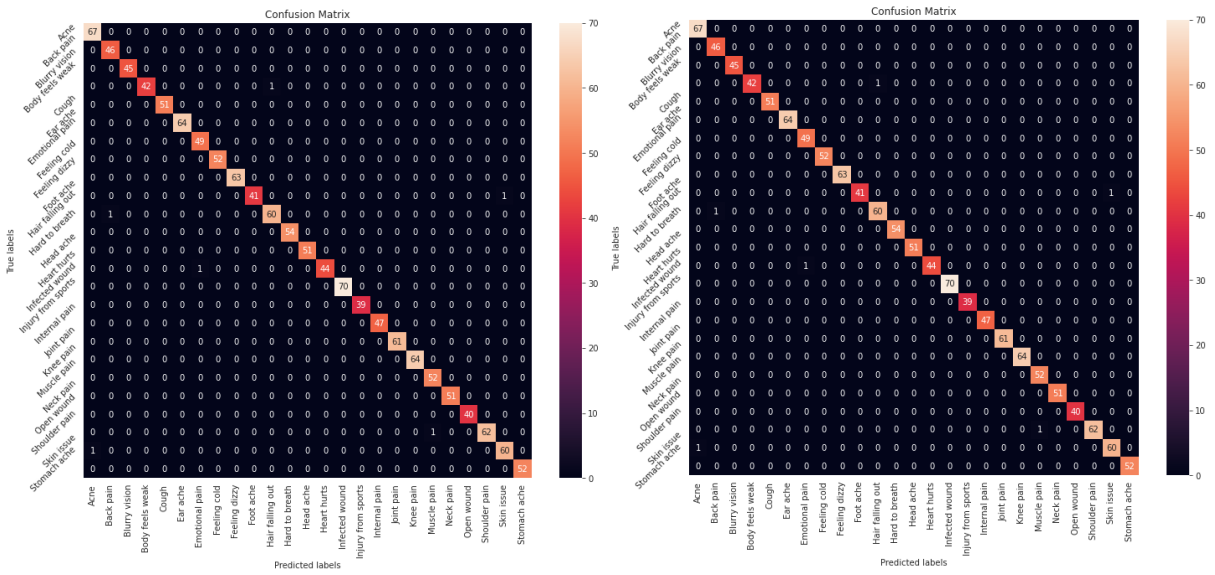


Figure 6. Confusion matrix for (a) stacked GRU model and (b) stacked LSTM model

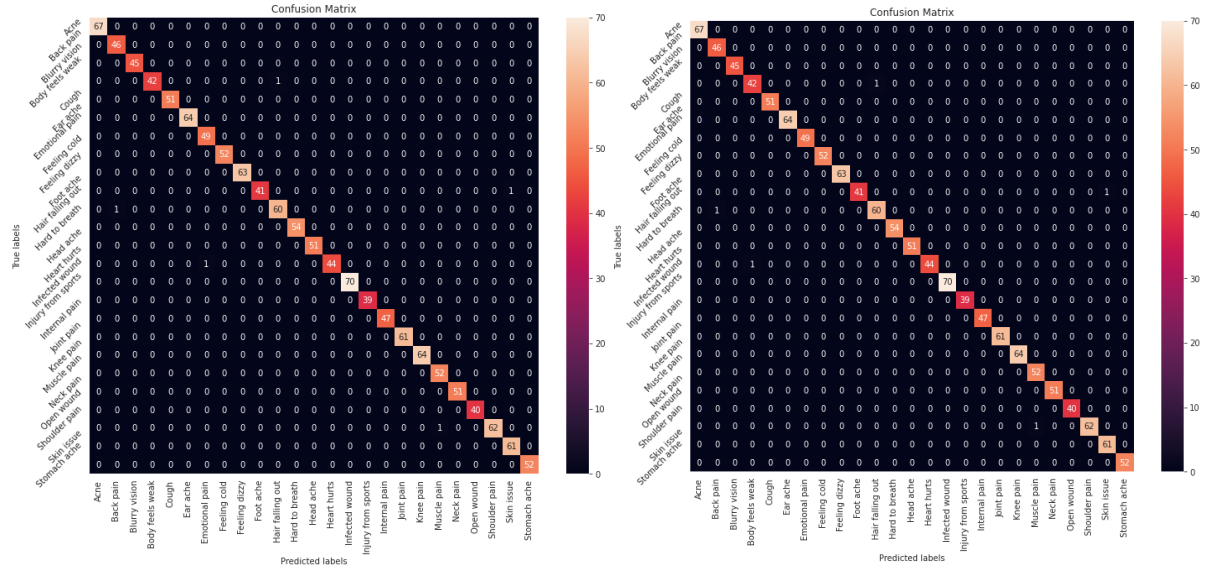


Figure 7. Confusion matrix for (a) stacked bidirectional GRU model and (b) stacked bidirectional LSTM model

In Figure 7, similarly from the corresponding confusion matrix, it could be seen that both the stacked bidirectional LSTM model and stacked bidirectional GRU model work well with most of the non-zero counts on the diagonal. The only difference was that the same “heart hurts” case that was misclassified as “emotional pain” by the stacked bidirectional GRU model but was misclassified by the stacked bidirectional LSTM model as “body feels weak”.

CONCLUSION

To conclude, we have demonstrated the application of Deep Learning in the healthcare settings using a stacked gated recurrent units (GRU) model, and a stacked long short-term memory (LSTM), as well as their bidirectional versions. The accuracy, precision, and recall of these models on the test data are high. Specifically, the stacked LSTM model, stacked bidirectional GRU model, and the stacked bidirectional LSTM model perform equally well in term of accuracy, while that of the stacked GRU model is slightly lower. The application could be used where automatic classification is needed to quickly categorize symptoms in associated categories for faster and more efficient healthcare operations.

REFERENCES

[1] "Elbattah, M., Arnaud, É., Gignon, M. and Dequen, G. The Role of Text Analytics in Healthcare: A Review of Recent Developments and Applications. DOI: 10.5220/0010414508250832 In proceedings of the 14th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2021) - Volume 5: HEALTHINF, pages 825-832.", .

[2] Sarkar, D. (2019). Text analytics with Python: a practitioner's guide to natural language processing. Apress., .

[3] Cheng, Y., Wang, F., Zhang, P. and Hu, J., "Risk Prediction with Electronic Health Records: A Deep Learning Approach," SDM (2016).

[4] Pham, T., Tran, T., Phung, D. and Venkatesh, S., "DeepCare: A Deep Dynamic Memory Model for Predictive Medicine," arXiv:1602.00357 [cs, stat] (2017).

[5] Miotto, R., Li, L., Kidd, B. A. and Dudley, J. T., "Deep Patient: An Unsupervised Representation to Predict the Future of Patients from the Electronic Health Records," 1, Sci Rep **6**(1), 26094 (2016).

[6] Dernoncourt, F., Lee, J. Y., Uzuner, O. and Szolovits, P., "De-identification of Patient Notes with Recurrent Neural Networks," arXiv:1606.03475 [cs, stat] (2016).

[7] Sherstinsky, A., "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network," Physica D: Nonlinear Phenomena **404**, 132306 (2020).

[8] Cho, K., van Merrienboer, B., Bahdanau, D. and Bengio, Y., "On the Properties of Neural Machine Translation: Encoder-Decoder Approaches," arXiv:1409.1259 [cs, stat] (2014).

[9] "Sepp Hochreiter; Jürgen Schmidhuber (1997). 'Long short-term memory'. Neural Computation. 9 (8): 1735–1780. doi:10.1162/neco.1997.9.8.1735. PMID 9377276. S2CID 1915014.", .

[10] "Appen, Medical Speech, Transcription, and Intent (English), https://appen.com/datasets/audio-recording-and-transcription-for-medical-scenarios/.", .