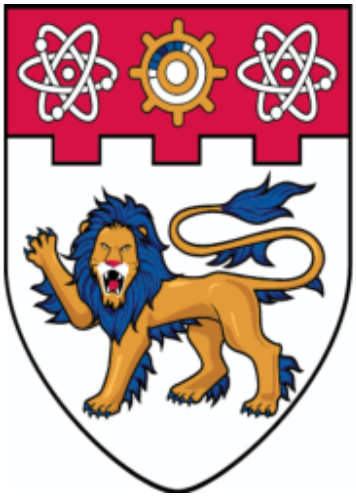

NANYANG TECHNOLOGICAL UNIVERSITY

SINGAPORE

Nanyang Technological University

CZ4045 Natural Language Processing Project

Lim Zhi Qing

Jared Chan

Mak Qin Xiang

Content Page

2. Sentence-Level Categorization: Question Classification

2.a Classes Used

The 5 classes used are 1, 2, 3, 4 and OTHERS.

2.b Aggregation Methods

We tested global max pooling and global average pooling as our aggregation methods. We adopted the global max pooling as it gave us a better accuracy as compared to global average pooling.

2.c Neural Network Architecture

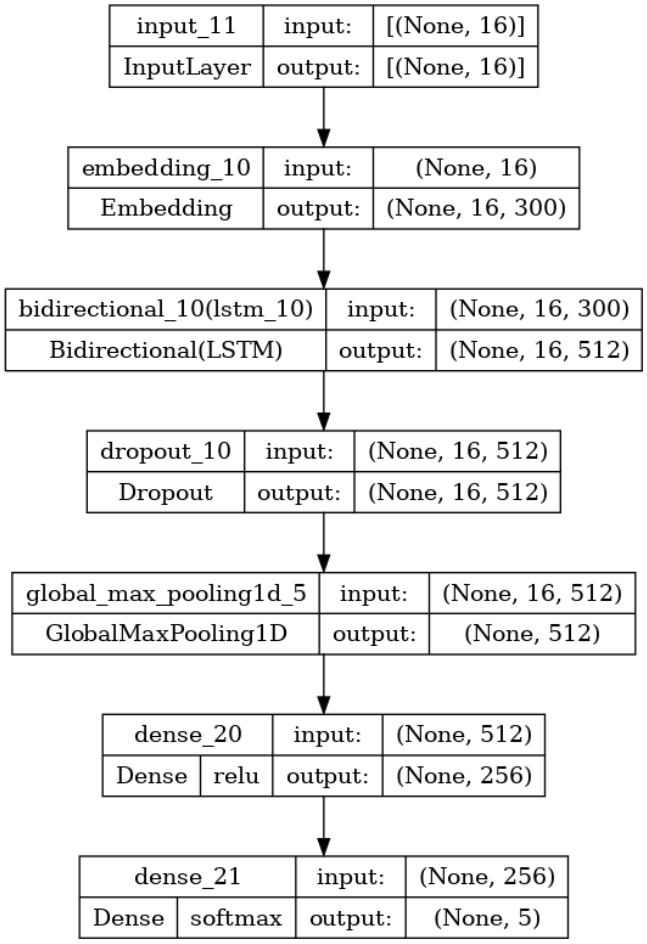

Figure 1: Neural Network Architecture.

Layer	Parameter	Shape
bidirectional (forward)	Weight	(300, 1024)
bidirectional (forward)	Bias	(1024,)
bidirectional (backwards)	Weight	(256, 1024)
bidirectional (backwards)	Bias	(1024,)
dense	Weight	(512, 256)
dense	Bias	(256,)
output	Weight	(256, 5)
output	Bias	(5,)

Figure 2: Parameters updated at each layer.

Embedding Layer

The first layer in the model is the embedding layer. The input is a 16 word sequence of tokenized text. Each word is converted into word embeddings by indexing the pre-trained Word2Vec embedding matrix. The resulting output is a sequence of 16 words, each with 300 features, giving the embedding layer 2400600 untrained parameters.

Bi-directional LSTM Layer

The bi-directional LSTM layer of 256 cells processes the embedded sequences in both the forwards and backwards directions, capturing dependencies from both past and future contexts. The output is a tensor of sequence length 16 with hidden dimensions 512. The dimensions of the hidden states is double the number of LSTM cells due to the bidirectional nature of the LSTM. In total, the Bi-directional LSTM layer has 1140736 trainable parameters. The weights and biases being updated can be seen in figure 2.

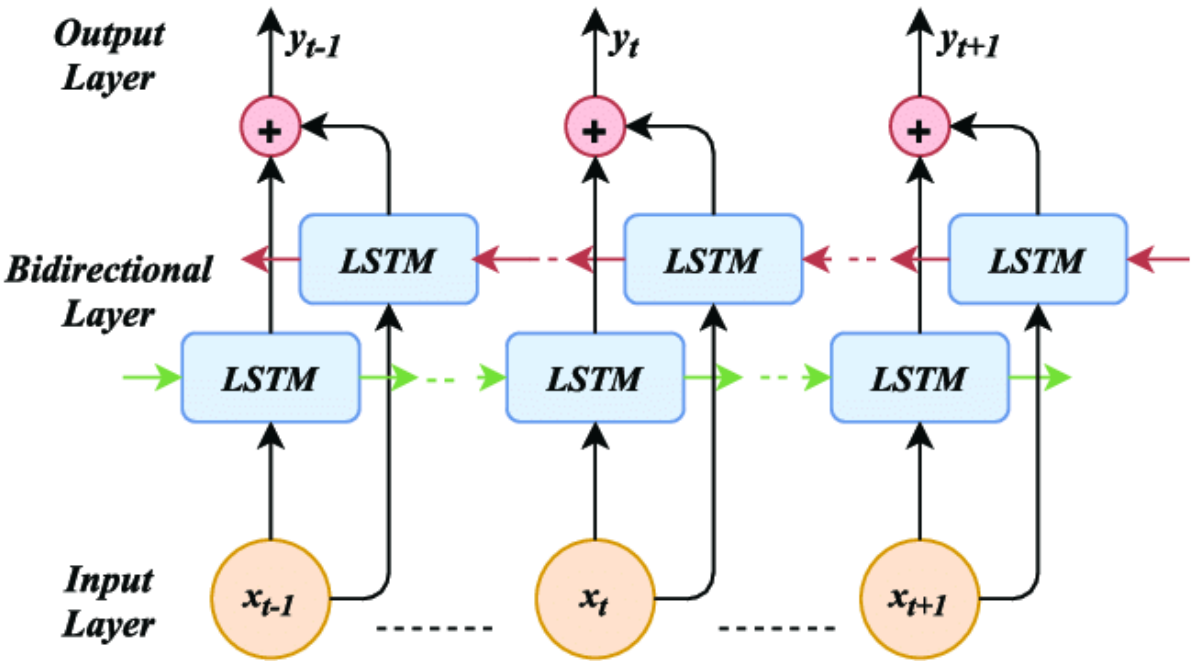

Figure 3: Bi-directional LSTM Model [1].

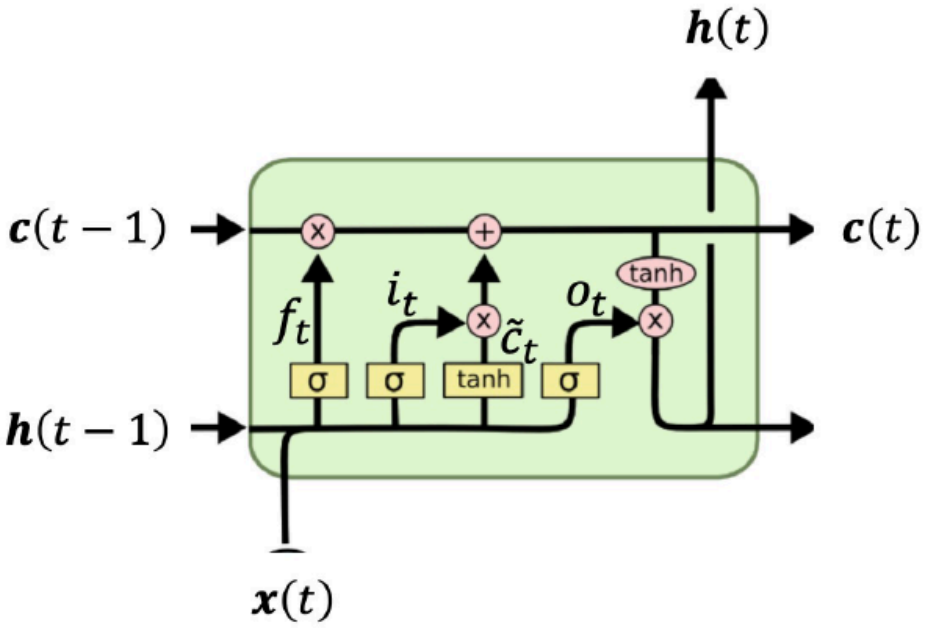

Figure 4: LSTM cell

In LSTM networks, the core idea revolves around the cell state,which allows information to flow along it with some controlled modifications through gates. LSTM cells have three main gates,

which are the forget gate, input gate and output gate. The following paragraphs on LSTM were taken from the CZ4042 Neural Network lecture notes on RNN.

The forget gate can modulate the memory cell's self-recurrent connection,allowing the cell to remember or forget its previous state, as needed. It is the f(t) as shown in figure 3, and it determines if c(t-1) is to be remembered or not.
Mathematical function: $\boldsymbol{f}(t) = \sigma\left(\boldsymbol{U}_{\boldsymbol{f}}^{\top}\boldsymbol{x}(t) + \boldsymbol{W}_{\boldsymbol{f}}^{\top}\boldsymbol{h}(t-1) + \boldsymbol{b}_{\boldsymbol{f}}\right)$

The input gate can allow incoming signals to alter the state of the memory cell or block it. It decides what new information to store in the cell stage. This has two parts: A sigmoid input gate layer decides which values to update and a tanh layer that creates a vector of new candidate values c̃(t) that could be added to the state.
Mathematical functions: $\boldsymbol{i}(t) = \sigma\left(\boldsymbol{U}_{\boldsymbol{i}}^{\top}\boldsymbol{x}(t) + \boldsymbol{W}_{\boldsymbol{i}}^{\top}\boldsymbol{h}(t-1) + \boldsymbol{b}_{\boldsymbol{i}}\right)$

$$\tilde{\boldsymbol{c}}(t) = \phi\left(\boldsymbol{U}_{\boldsymbol{c}}^{\top}\boldsymbol{x}(t) + \boldsymbol{W}_{\boldsymbol{c}}^{\top}\boldsymbol{h}(t-1) + \boldsymbol{b}_{\boldsymbol{c}}\right)$$

Cell state mathematical function: $\boldsymbol{c}(t) = \tilde{\boldsymbol{c}}(t) \odot \boldsymbol{i}(t) + \boldsymbol{c}(t-1) \odot \boldsymbol{f}(t)$

The output gate can allow the state of the memory cell to have an effect on other neurons or prevent it.
Mathematical functions: $\boldsymbol{o}(t) = \sigma\left(\boldsymbol{U}_{\boldsymbol{o}}^{\top}\boldsymbol{x}(t) + \boldsymbol{W}_{\boldsymbol{o}}^{\top}\boldsymbol{h}(t-1) + \boldsymbol{b}_{\boldsymbol{o}}\right)$

$$\boldsymbol{h}(t) = \phi\left(\boldsymbol{c}(t)\right) \odot \boldsymbol{o}(t)$$

Dropout Layer

The dropout layer prevents overfitting during training by randomly setting a fraction of input units to zero, helping with regularisation and generalisation of the model.

Pooling Layer

The pooling layer performs global max pooling on the output of the dropout layer, computing the maximum value of each feature over the sequence length, capturing the most important features from the input sequence. This layer takes in an input of dimension (16, 512) and reduces it to an output of dimension (512).

Densing Layer

The dense layer is a fully connected layer that takes the output of the pooling layer as input. The ReLU (Rectified Linear Unit) activation function is applied element-wise to each neuron's output. Negative values are replaced with zero while positive values are unchanged. This layer consists of 131328 trainable parameters, and the weights and biases being updated can be seen in figure 2. The resulting output is of length 256, which is the number of neurons in the dense layer. This output is the final vector representation of all the words in the input text, which is fed to the softmax classifier in the output layer.

Output Layer

Finally, the output layer is another dense layer consisting of 1285 trainable parameters, and the weights and biases being updated can be seen in figure 2. This layer contains 5 neurons, equal to the number of classification labels. Each neuron represents a class label. The softmax activation function takes the vector output of the previous layer as input and produces a probability distribution over the classes. Each class is assigned a probability and the class with the highest probability is predicted as the model's output.

2.d Epochs Used

In total the training took 7.6781 seconds, with 8 epochs.

2.e Accuracy

The accuracy on the development set for each training epoch are as follows.

Epoch	Training Loss	Training Accuracy	Validation Loss	Validation Accuracy
1	0.1982	0.9376	0.3981	0.856
2	0.1236	0.9548	0.3691	0.884
3	0.0461	0.9838	0.3638	0.900
4	0.0165	0.9966	0.4411	0.892
5	0.0079	0.9978	0.4575	0.882
6	0.0137	0.9956	0.4784	0.884
7	0.0058	0.9986	0.4671	0.890
8	0.0233	0.9919	0.5868	0.850

Figure 5: Table of Accuracy on Training and and Validation Set

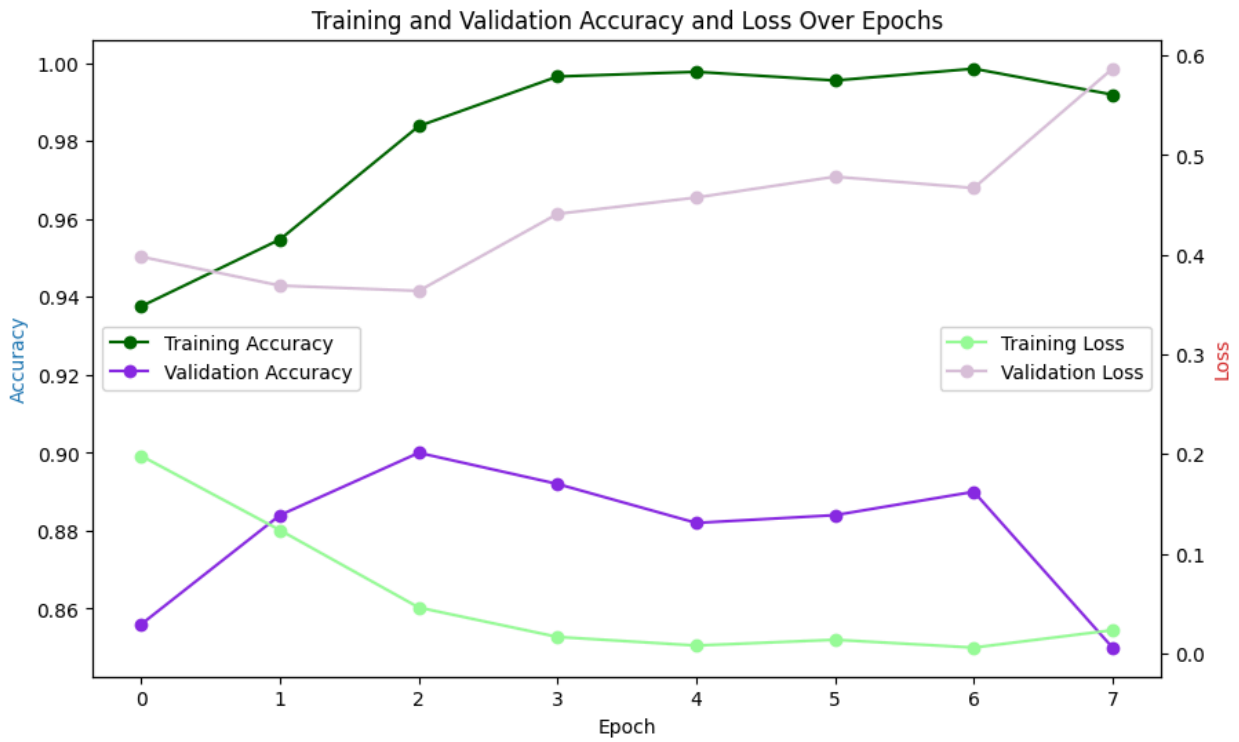

Figure 6: Graph of Accuracy on Training and Validation Set

The accuracy on the test set is 0.9040.

References

[1] I. K. Ihianle, A. O. Nwajana, S. H. Ebenuwa, R. I. Otuka, K. Owa, and M. O. Orisatoki, “A deep learning approach for human activities recognition from multimodal sensing devices,” *IEEE Access*, vol. 8, pp. 179028–179038, Jan. 2020, doi: 10.1109/access.2020.3027979.

Readme

Question 2:
Create a new notebook on Kaggle.
(https://www.kaggle.com/)
Import the notebook.
Import the TREC dataset.
(https://www.kaggle.com/datasets/thedevastator/the-trec-question-classification-dataset-a-longi)
Run all cells in the notebook.

model_maxpool.summary() this gives us the model architecture