**Deep Learning Challenge – Group 6**

Jose Orozco Becerra snr 2048765

Marloes Evers snr: 2051360

Sathya Krishna Sharma Jagannatha snr:2033987

**Codelab account name : sathyakrishna\_sharma**

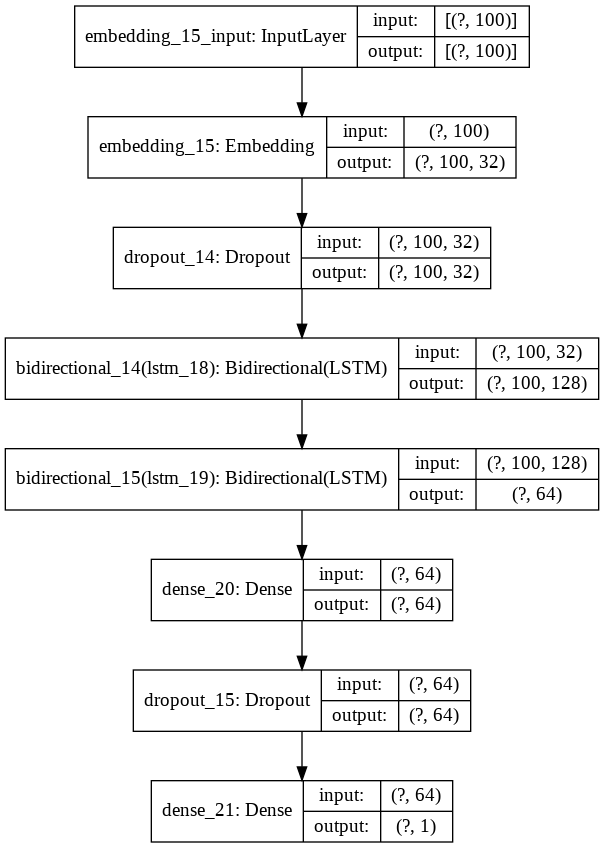


Figure1: Architecture of our model.

*• Explanation of the model*

This task has looked into the classification problem of classifying whether a sentence was an original English text or a human translation from French into English. The steps taken were:

The features and their labels were first loaded in and after this splitted into a training and a validation set. Thereafter, all sentences were transformed into sequences of integers using tokenization. Using post padding, all sequences were then transformed to arrays with the same dimensions. One-hot encoding was applied to change the labels into the numeric dummy variables 0 and 1.

The final sequential model consisted of 8 layers, as can be seen in figure 1, which shows the architecture of the built model. No predefined embedding layer was used, but a self-learning embedding layer was added to the model. Also two bidirectional LSTM layers and two dense layers were added, even as two drop-out layers, to avoid overfitting of the model.

Adam was chosen as optimizer for the model. Overall, this optimizer performs well on (most) classification problems, also proving that in this research. Several learning rates were tested varying from 0.1 and 0.0001. Eventually 0.001 was chosen, giving the best results. Because of the nature of this task, binary classification, the sigmoid activation was used in the last layer.

An early-stopping criterion was used to keep track of the validation loss during training. Patience=3 was chosen, automatically stopping the training when the model would encounter an increase in the validation for 3 epochs in a row. Accuracy was chosen to measure the performance of the model, as we were interested in finding out what part of the data our model would classify correctly.

*Justification of the model*

Because our task is to classify sentences and not words into classes, we chose bidirectional LSTM.

The LSTM model was chosen due to the fact that it can deal with long term dependencies (a problem that it may be found in sentences of translations) ( Graves et al., 2005). The architecture of the algorithm allows it to run straight down the entire chain, so the information can flow very easily without any changes. Also, it has the ability to remove or add information. The sigmoid layer outputs with the value of 0 “let nothing through to pass” while a value of 1 means “let everything through”. As stated before, for this project bidirectional LSTM was used. Bidirectional LSTM trains two layers on the input sequence. One LSTM layer on the input sequence and the second LSTM layer on the reversed copy of the input sequence provides more context for learning sequences which gives better results for NLP tasks.

*Accuracy results encountered*

|  |  |
| --- | --- |
| Evaluations | Accuracy |
| Training data | 79.8% |
| Evaluation data (validation) | 75.2% |
| Test data | 75% (codalab: sathyakrishna\_sharma ) |

References:

Graves, A., Fernández, S., & Schmidhuber, J. (2005, September). Bidirectional LSTM networks for improved phoneme classification and recognition. In *International Conference on Artificial Neural Networks* (pp. 799-804). Springer, Berlin, Heidelberg.

<https://medium.com/@vishwajeetkumar_85368/news-classification-using-bidirectional-lstm-and-attention-a67aa803ca74>

Chakravarthy,S. (2020). Tokenization for Natural Language processing. Retrieved from:

<https://towardsdatascience.com/tokenization-for-natural-language-processing-a179a891bad4>

Kumar, V. (2020). New Classification using bidirectional lstm and attention

<https://medium.com/@vishwajeetkumar_85368/news-classification-using-bidirectional-lstm-and-attention-a67aa803ca74>

Djajafer (2020). Multiclass text classification with keras and lstm. Retrieved from: <https://medium.com/@djajafer/multi-class-text-classification-with-keras-and-lstm-4c5525bef592>

Brownlee, J. (2017, October). How to Use Word Embedding Layers for Deep Learning with Keras. Retrieved from: <https://machinelearningmastery.com/use-word-embedding-layers-deep-learning-keras/>

Admin. (2020, February). Keras LSTM tutorial - How to easily build a powerful deep learning language model. Retrieved from: <https://adventuresinmachinelearning.com/keras-lstm-tutorial/>

tommytracey [username]. (2018, August). Machine Translation. Retrieved from: <https://github.com/tommytracey/AIND-Capstone/blob/master/machine_translation.ipynb>

Tensorflow documentation (n.d.). Tokenizer. Retrieved from:

<https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/text/Tokenizer>