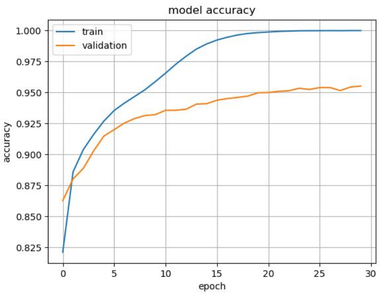
Text Based Information Retrieval

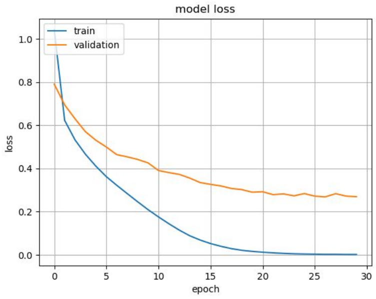
KU Leuven, Project 2017/18

Semantic Embeddings in Information Retrieval **Matija Kljun** r0725870
Part II - QA with visual features **Tomás Pereira** r0725869

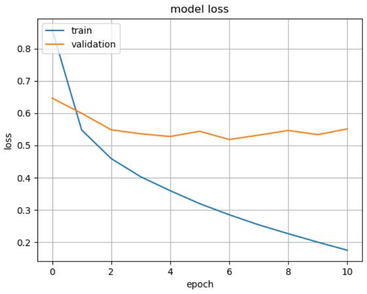
The objective for this second part of the project was to take what had been learnt during the first one, to improve it and to finally successfully implement a visual question answering model.

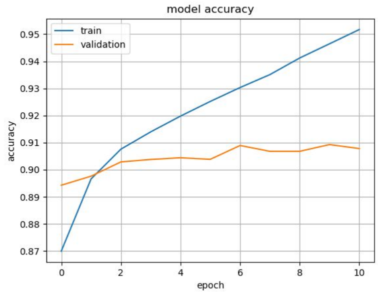
As previously noted, for the **question representation** task the group created a stateful LSTM autoencoder model with embedding, with an inference cycle for prediction, i.e. when in prediction mode, the decoder is initialized with a start token ('<go>') and the output sequence is continually generated until a maximum number of words is reached or the end token is predicted ('<eos>'). With this approach we trained the network with 512 units for 30 epochs (minimum validation loss of 0.2702, automatically detected with an early stopping mechanism), an accuracy of 0.3587 was reached for testing data.





For the **question answering** model, different ways of including the visual features in the textual model from the first part were tried. On the final model, a dense layer was applied to the output of stateful LSTM autoencoder and concatenated with the output of the dense layer on the image features. The output was then passed to another *tanh* dense layer and then to a softmax layer. To avoid overfitting on the training data, we used early stopping and dropout on the LSTM layer inputs. Running this model with 512 units for 30 epochs, using some data processing tweaks such as reversing the input sequence (as suggested in [1]), resulted in early stopping at the 11th and the following results were obtained:



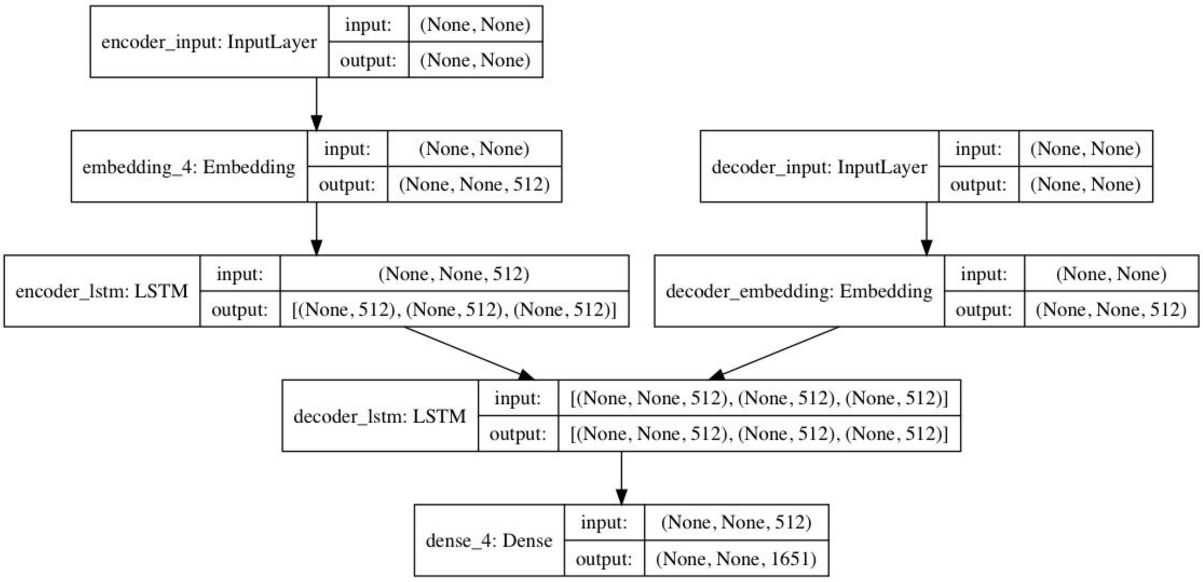


The resulted accuracy on the test data was 0.1604, with a WUPS score of 0.2404. This is an improvement over the accuracy of 0.1388 and the WUPS score of 0.2167 obtained with the question answering model without the use of visual features.

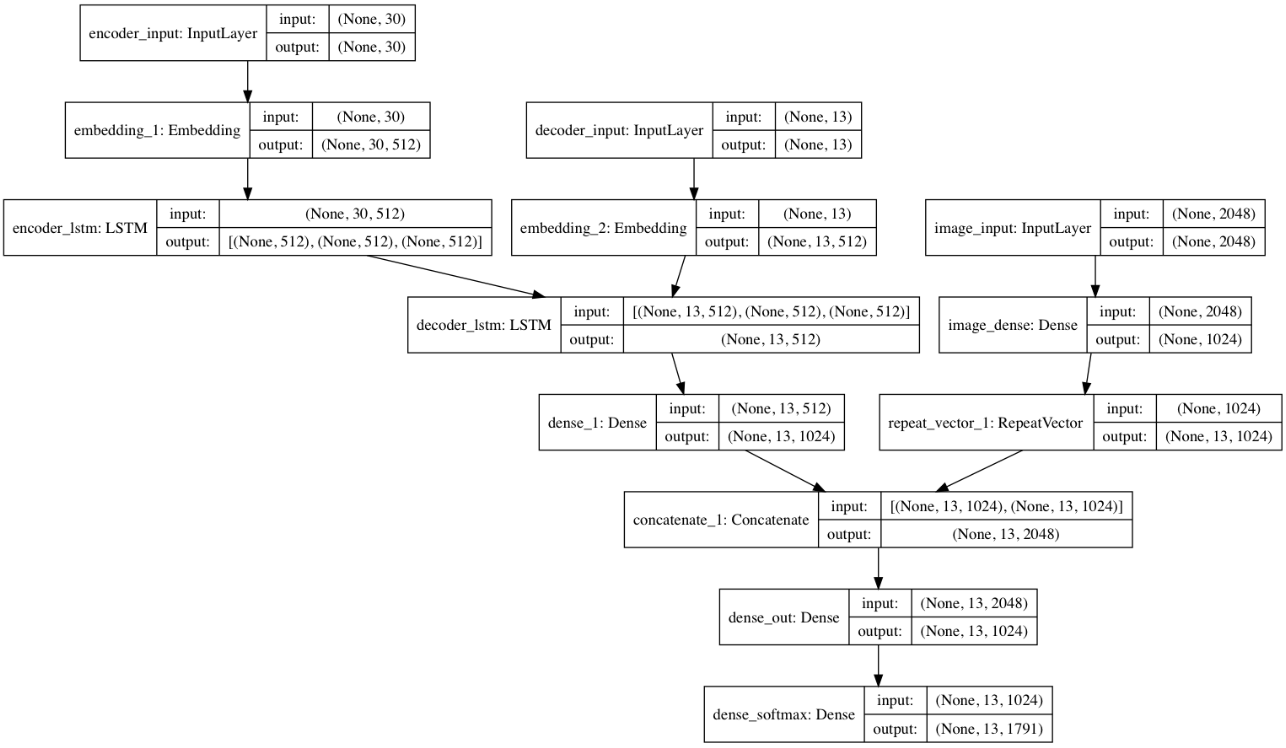
[1] Sequence to Sequence Learning with Neural Networks, Ilya Sutskever, Oriol Vinyals, Quoc V. Le https://arxiv.org/abs/1409.3215

Model structures

Question representation model structure


encoder_input: InputLayer
input: (None, None)
output: (None, None)
embedding_4: Embedding
input: (None, None)
output: (None, None, 512)
decoder_input: InputLayer
input: (None, None)
output: (None, None)
encoder_lstm: LSTM
input: (None, None, 512)
output: [(None, 512), (None, 512), (None, 512)]
decoder_embedding: Embedding
input: (None, None)
output: (None, None, 512)
decoder_lstm: LSTM
input: [(None, None, 512), (None, 512), (None, 512)]
output: [(None, None, 512), (None, 512), (None, 512)]
dense_4: Dense
input: (None, None, 512)
output: (None, None, 1651)

Visual question answering model structure


encoder_input: InputLayer
input: (None, 30)
output: (None, 30)
embedding_1: Embedding
input: (None, 30)
output: (None, 30, 512)
decoder_input: InputLayer
input: (None, 13)
output: (None, 13)
encoder_lstm: LSTM
input: (None, 30, 512)
output: [(None, 512), (None, 512), (None, 512)]
embedding_2: Embedding
input: (None, 13)
output: (None, 13, 512)
image_input: InputLayer
input: (None, 2048)
output: (None, 2048)
decoder_lstm: LSTM
input: [(None, 13, 512), (None, 512), (None, 512)]
output: (None, 13, 512)
image_dense: Dense
input: (None, 2048)
output: (None, 1024)
dense_1: Dense
input: (None, 13, 512)
output: (None, 13, 1024)
repeat_vector_1: RepeatVector
input: (None, 1024)
output: (None, 13, 1024)
concatenate_1: Concatenate
input: [(None, 13, 1024), (None, 13, 1024)]
output: (None, 13, 2048)
dense_out: Dense
input: (None, 13, 2048)
output: (None, 13, 1024)
dense_softmax: Dense
input: (None, 13, 1024)
output: (None, 13, 1791)