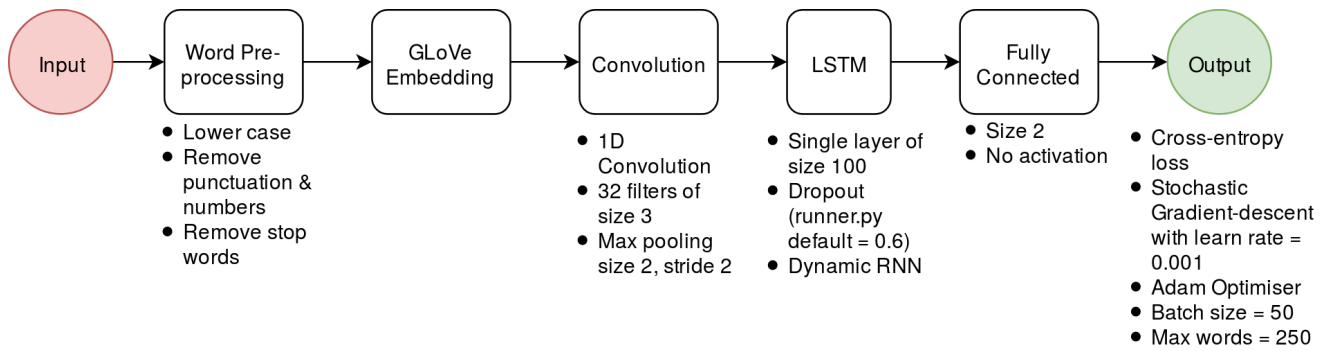


Final Model Structure



Input sentences were converted into lower case letters. Punctuation and numbers were removed so that only words remain. Stop words were removed to ensure a higher proportion of usable words. GLoVe embedding converts these words into 50-dimensional word vectors.

Initial model attempts were made using a LSTM layer and 1 fully connected layer. In these initial models, experiments with `BATCH_SIZE` and `MAX_WORDS_IN_REVIEW` found that smaller batch sizes sped up training time, and a higher number of words improved model accuracy faster. Final values were set to

`BATCH_SIZE = 50`

`MAX_WORDS_IN_REVIEW = 250`

A *1D convolution layer* was added to gain context from surrounding words. This layer includes 32 convolution filters, each of size 3. These filters are numerous enough to pick up a large number of features, without overfitting the data. The small filter size ensures that the layer only convolves around nearby words and does not lose context by considering words that are further away. *Max pooling* with size 2 and stride 2 is used to downsample the data, with only a small size to constrict the loss of information.

A *LSTM layer* was included so that the architecture can exhibit a memory of previous cycles. This layer was of size 100, which allowed for enough features to be learned without overfitting. A dropout layer was added to the LSTM as a regularisation technique, helping to avoid overfitting. Dropout rate is a placeholder variable, with the default set to 0.6 in `runner.py`. The layer was specified as a dynamic RNN to improve graph creation time and allow for variable size batches if need be.

A *fully connected layer* of size 2 was added to consolidate the features and structure the output. For simplicity, no activation function was included. Softmax activation could have been used instead to output a probability distribution.

Cross-entropy loss was used as the expected outputs take on the form of a 0/1 flag. The learning rate for gradient descent was set to 0.001 to ensure quick convergence without overshooting the minimum loss. The Adam Optimiser was chosen so that the learning rates are not necessarily constant across all trained variables.

Early stopping was used to avoid overfitting. The model was saved every 1000 steps, and chosen to maximise the validation set. This occurred at step 9000, with a validation accuracy of 84.6%.