**Working with .ipynb Files**

To ensure efficient processing of large .ipynb files, it is recommended to run Jupyter Notebook with the following command: `jupyter notebook --NotebookApp.iopub\_data\_rate\_limit=1.0e10`. This command increases the reading speed, which is essential when working with enormous file sizes, surpassing the default settings.

**Dataset**

For this task, I am utilizing the 'Dailymail stories' dataset available at https://cs.nyu.edu/~kcho/DMQA/. Since the original dataset size exceeds 200k+, I have selected a smaller subset to work with due to limitations on my system's capacity.

**Data Pre-processing**

To extract the relevant information from the files, I iterated through them and retrieved the text contents. For the summary, I employed the search term "@highlight" and retrieved the corresponding partition of the text. Subsequently, I employed gensim for basic data pre-processing, specifically for data cleaning purposes. After splitting the data into test and training sets, I employed a Tokenizer to convert the text into the required format.

**Model Building**

The model architecture consists of three Attention layers followed by an encoder with an RNN. The RNN component incorporates LSTM layers with a 20% dropout to mitigate the issue of vanishing gradients. Next, a decoder is included in the model, and subsequently, the model is compiled.

**Model Fitting and Testing**

The model was fitted with 10 epochs, considering the time-consuming nature of each epoch. Following the training process, accuracy and loss graphs were plotted to compare the performance between the training and test sets. Finally, the model was evaluated by conducting a self-test, which involved calculating different Bleu scores.