BTP-2

ML Summarization - Digitalee

Ishita Vohra, 20171054

Overview

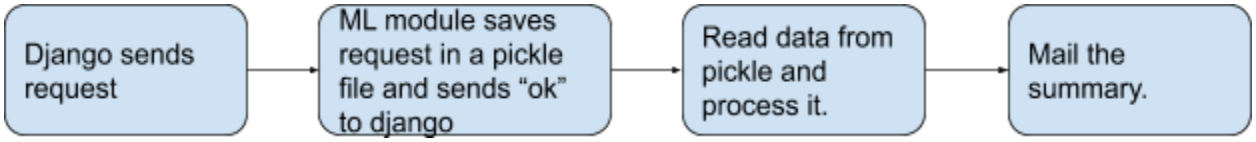
Since ML summary generated a good quality of summaries, this semester was focused on integrating ML summary with the system. I also explored various strategies of how we can further improve the ML summary and how we can improve our metrics of measuring the quality of the summary.

Workflow

The usual way of sending a request, processing it by generating summaries, and then sending it back didn't work for ML summary since it was computationally time expensive. The request consisted of the filename, keywords, data, and email of the person. The data is the raw text which we get after pdf parsing. Hence we tried various ways of resolving this issue. We are now saving the request in a pickle file. We send "ok" as a response to the Django server. Now we will read data from the pickle file, generate a summary and we will email the summary to the user.

Alternative techniques

We thought of other ways as well. Instead of storing it in a pickle way, we can store it in a blob file. We can also store the request and its corresponding summary in a database. Whenever a new user will request the summary for the same file and same keywords we can just return the summary from our database. This involves changing the whole system hence we didn't implement it.



Documentation

Apart from this, we had also explored the various new metrics to compare the performance of the summary. We found LSA based approaches for which we don't need manual summaries. Paper Link.

We had also explored various modules like GPT-2, Transformers, and BERT. This would be abstractive summarization and they will improve summarization quality. However, they consume a lot of resources and are time-expensive. Doc link

Future Work

- Currently, we only run the IREL summarizer on split pdf. We should also try running ML summarizer on split-pdf.
- Try out abstractive summarization since they will give better results. However, the trade-off needs to be considered.
- I looked at various ways of improving the ML summary. We can adopt the following measures
    - Changing sentence embeddings from skip thoughts to BERT embeddings
    - Since we have access to a very less number, proper evaluation of the algorithm for our needs might not be possible. Some algorithms which perform extremely well on some documents might perform very poorly on others. So we can consider ensembling outputs from different models using average scores or some other ranking criteria. This will improve baseline performance across all documents.

- Try training skip thoughts or any embeddings which we are using on Financial Corpus(We can try Enron Corpus for this).