**Into (10 words – using 10 [limit 660])**

Welcome everyone; our group members are Grufan, Prerna, and Fernando.

**Problem Statement (20 words – using 21 – total 31)**

For our project we use Natural Language Processing summarization algorithms to reconstruct headnotes using court case opinions as our training dataset.

**EDA, Filters, and Data Prep (90 – 188 – 219)**

We use a json dataset of cases from North Carolina provided by the Harvard Law School Library Case Law Access Project. There are ninety- seven thousand six hundred cases dating back to seventeen seventy- eight. There are thirty- two columns and we focused on the case body data, specifically the opinions and headnotes. Cases can have multiple or no opinions and headnotes. We observed up to six opinions in some cases.

We selected cases since two- thousand eight with headnotes having a length of more that one hundred fifty and with majority opinions longer than the headnotes, resulting in around thirty- seven hundred cases.

We removed the return characters, extracted, and tokenized the opinions and headnotes from the case body data. During which we utilized the pre-processing steps from each model to label the opinions sentences. Finally, we split the data into training, validation, and test sets.

For labeling, we compared sentences from each opinion with their headnotes, extracting those opinion sentences that increased the Rouge score to create the extractive summary. We then labeled the sentences that were included in the summary with one and zero otherwise.

**Overview of Research Done (60 – 60 - 279)**

We searched the NLP-Progress Summarization webpage for existing models that generate summaries based on:

The models’ ability to generate extractive summaries, since they usually generate semantically and grammatically correct sentences and compute relatively faster.

Their ability to summarize long documents and performs well as measured by the ROGUE score.

Most importantly, they can be implemented within the given time constraint.

**Describe Focus Areas & PreSumm & MatchSum (120 – 206 - 485)**

Our first model, PreSumm, was from the paper Text Summarization with Pretrained Encoders by Yang Liu and Mirella Lapata. They use combine text extractive and abstractive summarization.

PreSumm’s extractive model is a novel document-level encoder based on BERT stacked by several inter-sentence Transformer layers to capture document-level features for extracting sentences.

The abstractive model adopts an encoder-decoder architecture, combining the same pre-trained BERT encoder with a randomly-initialized Transformer decoder. It uses a new fine-tuning schedule as a means of alleviating the mismatch between the two.

The second model, MatchSum, was from the paper Extractive Summarization as Text Matching by Ming Zhong and others. They formulate an extractive summarization task as a semantic text-matching problem, in which a source document and candidate summaries will be (extracted from the original text) matched in a semantic space.

MatchSum uses a Siamese-BERT architecture, leveraging the pre-trained BERT to derive semantically meaningful text embeddings, to compute the similarity between several candidate summaries to the source document and selects the best candidate summary.

The Siamese networks consist of two identical neural networks, each taking one of the two inputs. The last layers of the two networks are then fed to a contrastive loss function, which calculates the similarity between the two inputs.

**Describe Result Metric &** [**Rouge Score**](https://rxnlp.com/how-rouge-works-for-evaluation-of-summarization-tasks/#.Xra4Ey33lE4) **(120 – 173 - 658)**

The models use the three Rouge scores, which are used for evaluating the summarization of texts. It works by comparing generated summaries against reference summaries.

Recall is how much of the reference summary is the generated summary capturing.

However, generated summaries could get too long, capturing all words in the reference summary with many additional useless words. Precision is used to prevent this.

Assigned by equal importance of recall and precision, i.e. alpha=0.5, is how the f-measure is computed.

The models use three Rouge scores: ROUGE-1, ROUGE-2, and ROUGE-L. ROUGE-1 measures the overlap of words, ROUGE-2 measures the overlap of two consecutive words, and ROUGE-L measures longest matching sequence of words.

**Show Example of What Worked and What Didn’t (120 – 226 - )**

Here we share an actual and MatchSum generated headnote. We see three segments that are related. The first is about the prevention of the defendant from working for all of plaintiff's current or recent clients, regardless of location. Next, the generated summary references the extreme scope. Lastly, it references the relatively small number of clients with whom defendant worked with.

In this PreSumm headnote sample we see two similar segments. The first is about how the trial court lacked jurisdiction to extend defendants period of probation verbatim. The second is referencing the arrest judgment and to vacate the order modifying probation and imposing sentence.

We can see the abstractive quality of the PreSumm generated headnote by the fragmented sentences in the initial words.

The table shows the Rogue scores for the various models. Our models do very well compared to the pre-trained Rouge scores. The MatchSum RoBERTa produced the highest scores.

Due to time constraints, we were not able to train the MatchSum hence the missing pre-trained Rouge score.

**Future Work, Learning, etc. (90 – 91 - )**

These models limit the maximum sentence length to 512 and we may want to alter the length to improve our results.

Due to time constraints, we were not able to utilize our full data, including the various opinions, to train the models. For future work, using all the available cases and opinions to train the models could improve the performance of generating headnotes.

**Closing (10 – 10 - 862)**

In conclusion, we were able to achieve high ROUGE scores with the models we chose.

Thank you for your undivided attention.