-Too Long Don’t Read TLDR: Text summarization from basic to advanced approaches

-So what is summarization? Text summarization is the task of producing a concise and fluent summary of texts or documents while preserving key information and overall meaning.

There are two main strategies for summarization. The first is summarization by extraction, which consists of concatenating source sentences into a summary. And second is summarization by abstraction, which involves generating novel sentences for the summary

The goal of this project is to implement different Text summarization techniques on the same data set and compare them based on the same evaluation metrics.

-Text summarization is becoming an important and useful task in several applications such as business analysis.

With this technique, we can gain the key information from articles and documents without reading through every word.

Here is a scenario that your boss wants you to write a summary of the report to save his time. Instead of reading the whole report word by word, you can use summarization model to generate a summary and give that to your boss. It is quite convenient and efficient.

-We use CNN and Daily Mail data set, which consists of online news articles paired with multiple summaries. This data is shown in the table.

The articles have an average of 781 tokens while the summaries have an average of 56 tokens.

For pre-processing step, we remove punctuation, number, and noisy words that are not related to news articles. We also lower every letter in all words.

And to save computation time, we use only the first two sentence of stories together with 2 summaries.

- For our approaches,

First, our baseline model is a vanilla LSTM sequence-to-sequence model. Generally, we use LSTM with an encoder-decoder architecture inspired by Neural Machine Translation and summaries are generated from the decoder, using target vocabulary. This model provides us a baseline of the training time and accuracy

For our second model, we finetune the BART-base model. BART-base is a pre-trained transformer model trained on XSum which is a news dataset. BART uses a standard sequence-to-sequence architecture with a bidirectional encoder similar to BERT and a left-to-right decoder similar to GPT.

For BART-large, we directly use the model without finetuning.

For T5, it is an encoder-decoder model pre-trained on a multi-task mixture of unsupervised and supervised tasks. It is trained on C4 dataset which is a cleaned version of Common Crawl’s web crawl corpus. The goal of this model is to reframe all NLP tasks into a unified text-to-text-format where the input and output are always text strings. T5 uses the same Masked Language Model as BERT but it is different from BERT-based models that can only output either a class label or a span of the input. The advantage of this model is that it allows us to use the same model, loss function, and hyperparameters on any NLP task

-To evaluate the generated summaries, we use F1 from the ROUGE metric which is commonly used in summarization tasks.

We select only three specific ROUGE metrics namely: ROUGE-1 and ROUGE-2 by measuring unigram and bigram similarity of reference summary and model-generated one. On the other hand, ROUGE-L measures the longest common subsequence between the two summaries.

In this example, we can see that there are total 7 words in the system summary, and 6 of them are overlapping words so the precision score should be 6 divided by 7 and recall score is 6 divided by 6. And these two numbers will be used for calculating F1 score for ROUGE-1

-Here is our result.

From the table, we can see that vanilla LSTM model that was trained from scratch has a relatively disappointing performance in all ROUGE metrics compared with other models.

The performance of Transformer-based models like BART-large model and all three T5 models almost doubles the LSTM model. Though, they have very similar performance to each other in all metrics regardless of size of models and underlying datasets.

On the other hand, our finetuned Transformer-based BART-base model outperforms LSTM but still cannot compete with other Transformer models. This may be because of how we perform finetuning or data processing.

Overall, the accuracy of the best model in our experiment is not high compared with the current state-of-the-art models. Nonetheless, in other experiments from different researchers using this same dataset, their best models hardly have an accuracy larger than 0.4. So, there is still a long way to go for text summarization.

-Here are the examples of our model outputs compared with a reference summary.

We can see that the summary output from LSTM model is incorrect as seen in the word like Manchester United repeating itself in both subject and object.

In contrast, the outputs of other models are correct and understandable in terms of meaning and grammar.

There are two key points in the reference summary.

The first is Manchester united have made Schweinsteiger as their top target. BART base model and all T5 models identified this point successfully.

The other point is Van Gall eyes Scheweinsteiger reunion at Old Trafford. And for this point, only BART large model can capture this detail.

To summarize, almost all the models successfully summarize one key point from the text except the LSTM model.

And the sentences they generate are readable and concise.

Although none of them can point out both of two key points, we think that their performance is acceptable and helpful.

-From our analysis, we conclude that

The accuracy of text summarization models can be increased by using transformer model.

Our current models can generate a correct and readable summary sentence in some cases.

Text summarization is still a challenging topic.

The overall accuracy of our models is still relatively low; and in other experiments, researchers can hardly generate a model with the accuracy higher than 0.4.

For Future work, we can incorporate Named-entity recognition to our model since large portion of summaries involve named entity.