

Day 113: NLP Papers Summary – On Extractive And Abstractive Neural Document Summarization With Transformer Language Models

Objective and Contribution

Proposed an abstractive summarisation method on long documents. This is achieved through a two-step process of extractive and abstractive summarisation. The output of the extractive step is used to train the abstractive transformer language model. This extractive step has been shown to be very important towards the end summarisation results. In addition, the generated abstractive summaries are more abstractive than previous work that employed copy mechanism and also yielded higher ROUGE scores. The contributions are:

1. Demonstrated the effectiveness of transformer language models in summarising long scientific articles, outperforming Seq2Seq models
2. The proposed model was able to produce more abstractive summaries than previous work and still achieve higher ROUGE scores

THE HUMAN SUMMARISATION PROCESS

1. Read and understand the source document
2. Select the most important parts of the source document
3. Paraphrase the key concepts in these important parts
4. Generate a coherent and fluent output summaries

Datasets

There are four different long document summarisation datasets:

1. arXiv
2. PubMed
3. bigPatent
4. Newsroom

Table 1: Statistics from (Sharma, Li, and Wang 2019) for the datasets used in this work - The number of document/summary pairs, the ratio of the number of words in the document to the abstract and the number of words in the summary and document.

Dataset	#Documents	Comp Ratio	Sum Len	Doc Len
arXiv	215,913	39.8	292.8	6,913.8
PubMed	133,215	16.2	214.4	3,224.4
Newsroom	1,212,726	43.0	30.4	750.9
BigPatent	1,341,362	36.4	116.5	3,572.8

Framework

The proposed framework is broken into two independent components:

1. *Extractive summarisation*. A hierarchical document model that either copy or classify sentences in the document to build extractive summary



2. *Abstractive summarisation*. The extractive summary as well as the document is used to condition the transformer language model

EXTRACTIVE SUMMARISATION


The extractive step involves sentence extraction using two different hierarchical document models: hierarchical seq2seq sentence pointer and sentence classifier. The goal is to filter out noisy sentences and extract important sentences to better train our transformer language model. The hierarchical seq2seq sentence pointer has an encoder-decoder architecture:

1. The encoder is a bidirectional LSTM at both the word and sentence level (hierarchical)
2. The decoder is an autoregressive LSTM

The hierarchical encoder combines both the word and sentence-level directional LSTM. The token-level biLSTM encodes each sentence in the document to obtain the sentence embeddings. The sentence-level biLSTM encodes these sentence embeddings to obtain document representations. The decoder is an autoregressive LSTM that takes in the hidden state of the previously extracted sentence as input and predict the next sentence to be extract.

Similar to the pointer network, the sentence classifier uses a hierarchical LSTM to encode the document and produce a sequence of sentence embeddings. The final document representation is the average of these sentence embeddings. The final document representation is concatenated to each sentence embedding and feed into a neural network with a sigmoid function to obtain the probability of each sentence to be included in the extractive summary.

ABSTRACTIVE SUMMARISATION

We trained a single transformer language model from scratch using “formatted” data. The transformer language model is GPT-2. Language models are trained by factorising joint distribution of words autoregressively. This inspires us to organise the training data in certain format where we put the ground-truth summary after the information the model would normally use to generate summaries. In this way, we model the joint distribution of document and summary during training and use the conditional distribution (given the document) to generate summary at inference. Therefore, the training data is formatted in 4 different ways:  ons:

1. *Paper Introduction*. Assumption that introduction should contain enough to generate the abstract
2. *Extracted summary (from extractive summarisation)*
3. *Abstract (ground-truth summary)*
4. *Rest of the paper*. Serve to train language model to understand domain language

For some datasets, the introduction section would be the entire document as there are no rest of the paper section. Figure below showcase the overall framework.



Results and Analysis




Table 2 and 4 showcase that our extractive models outperformed all previous extractive baselines on both arXiv and PubMed datasets. On the Newsroom dataset (table 6), our TLM outperformed the other abstractive model, Seq2Seq, by a massive margin and also outperformed the pointer-generator network. However, the Exconsumm model dominates the extractive and mixed results.





The best performing TLM (TLM-I+E (G,M)) has outperformed previous abstractive results on most ROUGE scores metrics except on ROUGE-L. We believe this might be due to the fact that we don't have a copy mechanism in place, making it very challenging to get exact matches on large n-grams. The figure below supports this hypothesis as the copy mechanism of the discourse-aware model can copy up to 25-grams from the source document. In addition, the figure below also showcase that our TLM has generated more abstractive summ



previous work by the low percentage of n-grams overlap between generated summaries and source documents.

We also measure the upper bound performance of our TLM (TLM-I+E (G,G)) by including the ground-truth extracted sentences in both training and testing. Lastly, the figure below showcase a qualitative results of summaries generated by our TLM.



Conclusion and Future Work

The fluency and coherency of the generated summaries are of strong level. However, there remains the problem of abstractive summaries generating imaginary / inaccurate content. Potential future work could focus more on factual correctness and coherency when evaluating summarisation models.

Source: <https://arxiv.org/pdf/1909.03186.pdf>





Ryan

Data Scientist

Previous Post

< [Day 112: NLP
Papers Summary -
A Challenge
Dataset and
Effective Models
for Aspect-Based
Sentiment Analysis](#)

Next Post

[Day 114: NLP
Papers Summary -
A Summarization
System for
Scientific
Documents](#)





[Data Science](#) [Natural Language Processing](#) [NLP Papers Summary](#)

Day 365: NLP Papers Summary – A Survey on Knowledge Graph Embedding

Ryan

30th December 2020





[Data Science](#) [Implementation](#) [Natural Language Processing](#)

Day 364: Ryan's PhD Journey – OpenKE-PyTorch Library Analysis + code snippets for 11 KE models

Ryan

29th December 2020





[Data Science](#) [Ryan's PhD Journey](#)

Day 363: Ryan's PhD Journey – Literature Review – Knowledge Acquisition – 1st Passes XIV

Ryan

28th December 2020

