HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION
TECHNOLOGY FACULTY OF COMPUTER SCIENCE

PROJECT II: ABSTRACTIVE LONG TEXT SUMMARIZATION

Course: Project II - IT3930E
Class Code: 738755
Supervisor: Associate Prof Nguyen Thi Kim Anh
Student: Vu Lam Anh
Student Code: 20214876

Contents

1 INTRODUCTION

Text summarization is one of the main tasks in Natural Language Processing and has been applied in many areas. It aims to produce the shorter version of a long text or document while preserving the key information and meaning from the original document.

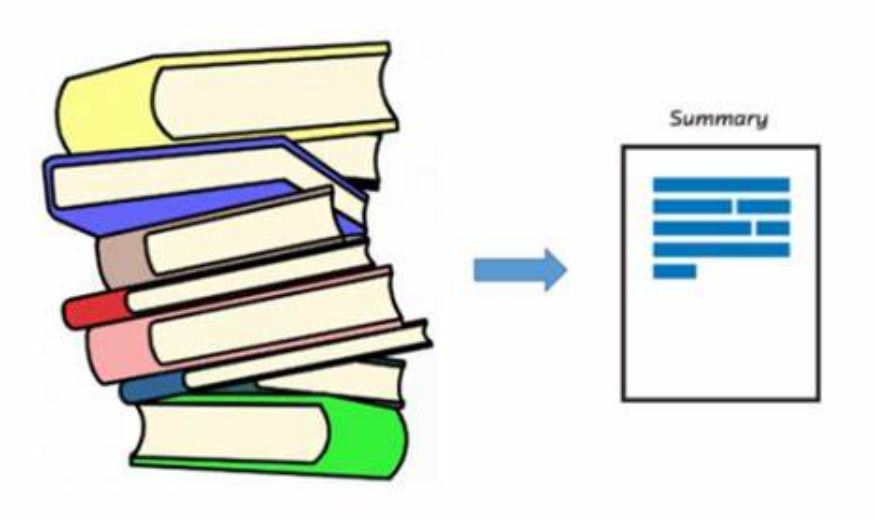

Figure 1: Summarize the whole book content

Specially, in an age of abundant information, summarizing lengthy texts efficiently is essential. Traditional extractive summarization methods, which select key sentences from the source text, often fail to capture the full essence and overarching themes of complex documents. This highlights the need for long abstractive text summarization, which creates concise, coherent summaries by rephrasing and synthesizing the core ideas. However, this task presents several challenges. Two main challenges among all of that are how to accurately capturing the whole context and the computing resources will increase exponential with the length of document.

In this project, we want to consider various approaches that can deal efficiently with long text and try to compare the efficient of each approach based on evaluating on the specific long document dataset.

2 DATASETS

2.1 PubMed overview

PubMed is a free resource supporting the search and retrieval of biomedical and life sciences literature with the aim of improving health–both globally and personally.

The PubMed database contains more than 37 million citations and abstracts of biomedical literature. It does not include full text journal articles; however, links to the full text are often

present when available from other sources, such as the publisher's website or PubMed Central (PMC).

Available to the public online since 1996, PubMed was developed and is maintained by the National Center for Biotechnology Information (NCBI), at the U.S. National Library of Medicine (NLM), located at the National Institutes of Health (NIH).

2.2 PubMed datasets

PubMed Datasets is a widely used dataset based on scientific documents. It consists of long documents of length ranging from several thousands of words to over ten thousands words. Each document in PubMed is a scientific article, collected from PubMed.com, and the reference summary is the associated abstract.

Citations in PubMed primarily stem from the bio-medicine and health fields, and related disciplines such as life sciences, behavioral sciences, chemical sciences, and bio-engineering.

Next, we will introduce some statistic information of PubMed dataset as the image below.

Dataset Split	Number of Instances	Avg. tokens
Train	119,924	3043 / 215
Validation	6,633	3111 / 216
Test	6,658	3092 / 219

Figure 2: General statistic information of datasets

PubMed dataset take the whole content of paper as original document and consider the abstractive as the summary of the summary of document. With the average tokens higher than 3000 tokens, PubMed document is considered "Long document" and also abstractive since it use the abstract part as the summary.Therefore, in this project, we will use this dataset to illustrate various approaches of dealing with abstractive long text summarization.

3 APPROACH

In general, to deal with the long text summarization with transformer-based model, there are three common ways:

- **End-to-End model with efficient attention:** The model has the standard transformer architecture but instead of using full attention, it use attention mechanism which has the memory complexity O(n) with n is the length of documents.
- **Retrieve-then-summarize:** Like its name, this model will have two stages: content selection and abstractive summarization. With this model, instead of summarizing full

text, it try to select the most important content of document then summarize based on that contents.

- **Divide-And-Conquer:** This model assume that the document has some specific structure and it try to divide the full documents into set of smaller documents based on their structure. Then it will feed each document subset into the model and in the end, the model will concatenate all the summary of each subset into one overall summary of the whole document.

In this report, we focus on the two approaches: End-to-End model with efficient attention and Retrieve-then-summarize while leaving the third ones for exploring in the future.

3.1 Efficient attention

In this original paper [1], the Transformer model has a full self-attention component with $O(n^2)$ time and memory complexity where n is the input sequence length. Therefore, when the sequence length increase so much, it is infeasible to train model due to the limitation of time and memory. To address this challenge, we sparsify the full self-attention matrix according to an "attention pattern" [2] which will scales linearly with the input sequence, making it efficient for longer sequences.

There are three options to design for attention pattern: **Sliding window**, **Dilated Sliding Window**, **Global Attention**

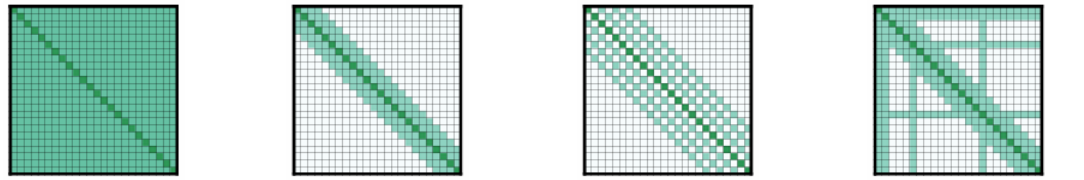

Figure 3: The full attention and various options of sparse attention

1. **Sliding window**

   This option employs a fixed-size window attention surrounding each token. Instead of attending to full sequence length n, it only attend on the fixed size window w. Given the fixed size window w, each token attends to $\frac{1}{2}w$ on each side. Since the size window w is pretty smaller than the length sequence n, the complexity scales linearly. Same as the convolution neural network, in a transformer with l layers, the receptive field size at the layer l_i is $l \times w$. Therefore, the more in the top layer, it capture the farther information in sequence.

2. **Dilated Sliding Window**

   In this attention variant, to further increase the receptive field without increasing computation, the sliding window can be "dilated" Fig. 3c. Assuming a fixed d and w for all layers, the receptive field is $l \times d \times w$, which can reach tens of thousands of tokens even for small values of d.By setting with different dilation configurations per head, it will help improve performance by allowing some heads without dilation to focus on local context, while others with dilation focus on longer context.

3. **Global Attention**

   However, the windowed and dilated attention are not flexible enough to learn task-specific representations. Therefore, it's good to add "global attention" on few pre-selected input locations.Particularly, we specify some tokens with a global attention attends to all tokens across the sequence, and all tokens in the sequence attend to them. Fig. 3d shows an example of a sliding window attention with global attention at a few tokens at custom locations. Applied to the text summarization task, it's easy for people who want to add the inductive bias to model's attention, so when training model, those important chosen token will go with global attention for capturing information better. Moreover, Since the number of such tokens is small relative to and independent of n the complexity of the combined local and global attention is still $O(n)$.

By combining all the options for efficient attention above, we can deal better with the long document.

3.2 Longformer Encoder Decoder

As we know the pre-trained encoder-decoder Transformer models such as BART have achieved the quite strong results in the text summarization. Yet, such model can't efficiently scale when dealing with long sequence. Therefore, based on the options for efficient attention we discuss above, Longformer Encoder Decoder [2] which effectively applies the efficient attention into the encoder-decoder Transformer models that will help us to cope with the long text better.

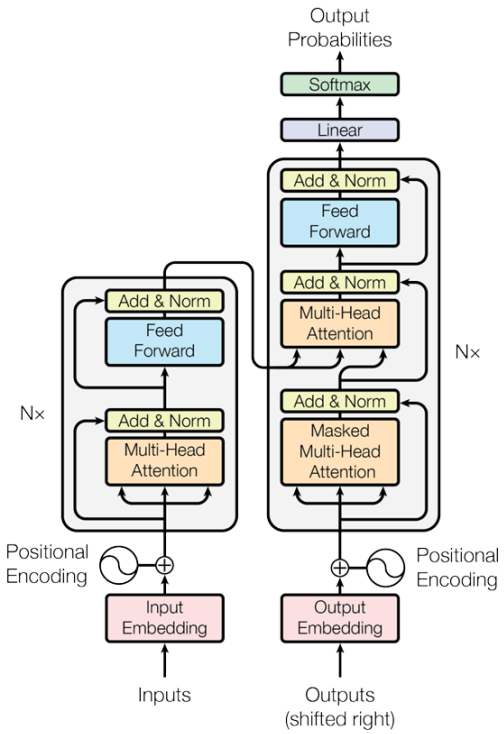

Figure 4: Transformer standard architecture [1]

Particularly, LED uses the efficient local+global attention pattern with window size is 1024 tokens and the global attention on the first $< s >$ token in the Longformer encoder. The decoder uses the full self-attention to the entire encoded tokens and to previously decoded locations. In the original paper, LED initialize parameters from the pre-trained BART, and follow BART's exact architecture in terms of number of layers and hidden sizes. The only differences is that to process longer inputs, LED extend position embedding to 16K tokens by repeatedly copying BART's 1K position embeddings 16 times. Same as BART, LED also has two versions: LED-base and LED-large, which respectively have 6 and 12 layers in both encoder and decoder stacks.

3.3 Retrieve-then-summarize approach

In this section, we will discuss about the Retrieve-then-summarize approach. In this approach, we need to select the most salient contents in the document first, after that, we will make summary based on that contents. There are two general approach to the content selection are:

1. **Ground-truth based (model-free)**: which is also referred to as oracle. With the ground-truth based approach, we will use the metrics between each sentence in the document and the reference summary to choose the most important sentence in the document

2. **Model-based**: This approach involve in extracting the importance information by using extraction model.

Each approach has its own weakness: Oracle method is only possible in the training time when we have the reference summary. With the model-based, it's feasible in the inference time, but in the training time, we don't have the labeled data for training.

Based by the paper "Long-Span Summarization via Local Attention and Content Selection" [4], we will address the problems in two above approach by combining them. The basic idea is that we will use the chosen the most salient information by the oracle approach for training extraction model and then we can use that model to extract the most important content in the inference time as image shown below.

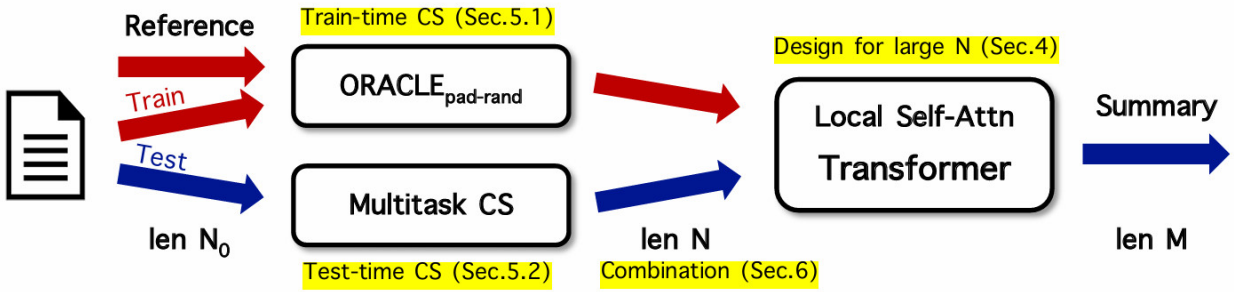

Figure 5: The whole process both training phase and testing phase [4]

In the next, we will discuss details about each component.

3.3.1 ORACLE

In this approach, we will some kinds of similarity measures d between each sentence in the document and the reference summary to rank the sentences in the document. Given the ground-truth summary y and similarity measure d, we compute the ranking of sentences as $r_k = \{i \in N : d(x, y)\}$. In this report we choose d is the ROUGE-2 recall. The ROUGE-2 recall measure by dividing the number of the overlap of bigrams between the predicted and reference summaries by the total number of bigrams in references summaries. After computing the ranking score, we can re-rank the sentence with maintaining the original sentence order as follow:

$$X = \{x_{r1}, x_{r2}, ..., x_{rN}\}$$

$$X^{CS} = SortOrig(TruncateN(X))$$

where r_i is the index of the sentence of rank i, the $TruncateN$ operation filters X such that the total of number of words is less than N, and $SortOrig$ retains the original sentence order.

However,the no-padding oracle method (ORC_{no-pad}) like above is highly aggressive, potentially preventing the downstream summarizer from learning complex abstraction. Therefore, there are proposed variant of oracle method is $ORC_{pad-rand}$. With $ORC_{pad-rand}$, instead of compute ranking score with only the sentence individually, it pad by random unselected sentences and keep the original sentence order then compute the score.

3.3.2 MULTITASK CONTENT SELECTION

In the model-based methods, there are two types are treating content selection as extractive labelling or using encoder decoder attention mechanism. To utilize both types, the multitask content selection method proposes to train the hierarchical encoder decoder with attention mechanism and a classification layer on top of the encoder.

1. **Training Stage**

   First, the model is trained on seq2seq RNN hierarchical abstractive summarization objective with loss function as follow:

   $$L_{seq2seq} = -\sum_{m-1}^{M} logP(y_m|Y < m, X)$$

   Second, we create binary labels as follows: for sentence i, the label z_i is 1 if $d(xi, y) > 0$; else z_i is 0, and d is the ROUGE-2 recall measure. The extractive labelling task objective is:

   $$L_{label} = -\sum_{i=1}^{N}(z_i log\hat{z}_i + (1 - z_i)log(1 - \hat{z}_i))$$

   $$z_i = sigmoid(W_{cls}^T h_i + b_{cls})$$

   where h_i is the sentence-level encoder output associated with sentence i, and $W_{cls}b_{cls}$ are the parameters of the classification layer.

   The MCS training loss is as follow:

   $$L_{MCS} = \gamma * L_{seq2seq} + (1 - \gamma) * L_{label}$$

2. **Inference Stage**

   For sentence i, the scores are computed as follow:

   $$score_{i,label} = \hat{z}_i$$

   $$score_{i,seq2seq} = \sum_{m=1}^{M} \alpha_{m,i}^{s}$$

where $\alpha^s_{m,i}$ is the sentence-level attention weight at decoder step m over input sentence i. The MCS inference score is:

$$score_{MCS} = n\_score_{i,label} + n\_scorei, seq2seq$$

where n_score is the normalized score

Based on that MCS score, we can rank the most important content and select from the highest to lowest score until we reach the maximum length. After that we can feed that shorter sequence into the abstractive summarization model.

4 EXPERIMENT AND RESULTS

4.1 Experiment

4.1.1 Data Pre-processing

To load and use PubMed dataset easily, we load the Pubmed dataset section from the **armanc/scientific_papers**[5] Huggingface hub. This PubMed dataset has the average 102 sentences in the document and approximately 7 sentences in the summary. Length of each sentences are around 30 tokens. This dataset has the three subsections which are train dataset, validation dataset, test dataset with the number of samples correspondingly: 119924 samples, 6633 samples, 6658 samples.

The average number of tokens for each document in the PubMed dataset is about over 3000 tokens. But in fact, not almost of documents have the number of tokens like that since there are many document seem very short and some are much longer than 3000 tokens. Due to purpose of this project which is deal with long document, we filter out all the document which have the number of tokens that are lower than 4096 tokens and only keep those has higher than that. Therefore, it will notify the effectiveness of the mentioned method above.

Moreover, because of the limitation of computing resources, we only use 20000 samples for training and about 1500 samples for validating and testing.

4.1.2 BART experiment setup

Although BART model is not our main points we want to consider in this project, but we want to use it as the baseline and can show clearly the effectiveness of LED over BART and BART with content selection compared to BART with only truncation. To adapt that reason, we will fine-tuned BART model in two versions: No Content Selection (Truncation)

and with Content Selection. For simplicity, we will fine-tune the pre-trained model from **facebook/bart-large-cnn** HuggingFace hub.

4.1.3 LED experiment setup

We utilize the pre-trained LED model by loading the model from **pszemraj/led-base-book-summary** Hugging face hub. This pre-trained model is the fine-tuned version of original LED based model on the booksum dataset. The window size of the model is the same from the original paper is 1024 tokens and also global attention is on the first $< s >$ token. We use the base model of LED so that it has 6 layers in both encoder and decoder.

We fine-tune two version for LED model : One is the model with maximum number tokens of input is 4096 tokens and other is 8192 tokens.

Belows are the hyperparameter settings for fine-tuning model:

- learning_rate: 5e-05
- train_batch_size: 2
- eval_batch_size: 2
- seed: 42
- gradient_accumulation_steps: 8
- total_train_batch_size: 16
- optimizer: Adam with betas=(0.9,0.999) and epsilon=1e-08
- lr_scheduler_type: linear
- lr_scheduler_warmup_steps: 700
- num_epochs: 2
- label_smoothing_factor: 0.1

4.1.4 Content Selection experiment setup

In the training stage, we use the $ORC_{padrand}$ method with the similarity measure which are ROUGE-2 recall to select the most important sentences with maintaining the order based on raking score until total tokens in the document exceeds 4096 tokens for LED model and 1024 tokens for BART model. After that, we use that chosen sentence for training the end-to-end abstractive model: LED, BART. The hyperparameter settings for this training are the same as the one we write in the LED experiment section.

During the training stage, we also train the hierarchical encoder decoder with attention mechanism with based on the abstractive summary and a classification layer on top of the encoder based on the content which are chosen by the $ORC_{padrand}$ method above.

The hyperparameter settings for the training are below:

- embedding_dim: 256
- rnn_hidden_size: 512
- num_layers_enc : 2
- num_layers_dec : 1
- learning_rate: 0.001
- train_batch_size: 2
- eval_batch_size: 2
- optimizer: Adam with betas=(0.9,0.999) and epsilon=1e-08
- lr_scheduler_type: linear
- lr_scheduler_warmup_steps: 1500
- num_epochs: 2
- dropout : 0.1

In the inference stage, we use the hierarchical model above to compute the MCS score and based on that we can rank the most important content and select from the highest to lowest score until we reach the maximum length : 4096 tokens for LED and 1024 tokens for BART.

4.2 Results

We evaluate all the model with the ROUGE score. Below are the results of all models:

System	R1	R2	RL	RL-Sum
BART- 1024	33.58	11.37	18.96	30.21
BART-1024-CS	35.67	12.72	19.96	32.20
LED - 4096	42.72	13.86	21.06	37.39
LED - 4096 - CS	43.68	14.25	21.42	38.14
LED - 8192	**44.67**	**15.49**	**22.56**	**39.30**

5 CONCLUSION

In this project, we consider two approaches: End-to-End model with efficient attention and Retrieve-then-summarize with LED and BART. With discussing about theory, we also try to implement each approach and evaluate the efficient of each approach on the PubMed dataset based on ROUGE family score.

Our experimental results demonstrate that effectiveness of model with efficient attention over the original BART model in dealing with long document. The content selection also show its advantage when the model with content selection perform better than ones without content selection.But, in overall the best model is LED with number of input tokens is 8192 tokens, it shows that the content of document in this dataset are distributed over the whole text, not in some specific parts and they are closely related to each other. Therefore, the content selection in this case will not maximize its efficiency, while increasing the number input tokens for capturing much information is more appropriate.

6 FUTURE WORK

In the future work we want to explore the third method we mentioned above: the "divide and conquer" method, which involves splitting the text into smaller sections, summarizing each independently, and then combining the summaries for coherence. Moreover, for the document that has the structure like document in PubMed, some methods of incorporating structural information, such as headings and subheadings, can prioritize important content and create hierarchical summaries that reflect the document's organization.Although the metrics score are quite good but the actual result is bad when there are many repetition of words, broken of coreference,... so apply some post-processing techniques, like enhancing text coherence, correcting grammar, eliminating redundancies, can further refine the summaries for better readability and quality.

7 REFERENCE

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin(2017): Attention Is All You Need

[2] Iz Beltagy, Matthew E. Peters, Arman Cohan(2020): Longformer: The Long-Document Transformer

[3] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, Luke Zettlemoyer(2019): BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension

[4] Potsawee Manakul, Mark J. F. Gales(2021): Long-Span Summarization via Local Attention and Content Selection

[5] PubMed dataset: armanc_scientific_papers

[6] Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, Nazli Goharian(2018): A Discourse-Aware Attention Model for Abstractive Summarization of Long Documents