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

PROJECT II: ABSTRACT 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 TO TEXT SUMMARIZATION

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. With the exponential growth of digital data, summarization techniques have become indispensable for efficiently extracting valuable insights from large volumes of text so that helping people work faster and more efficiently.

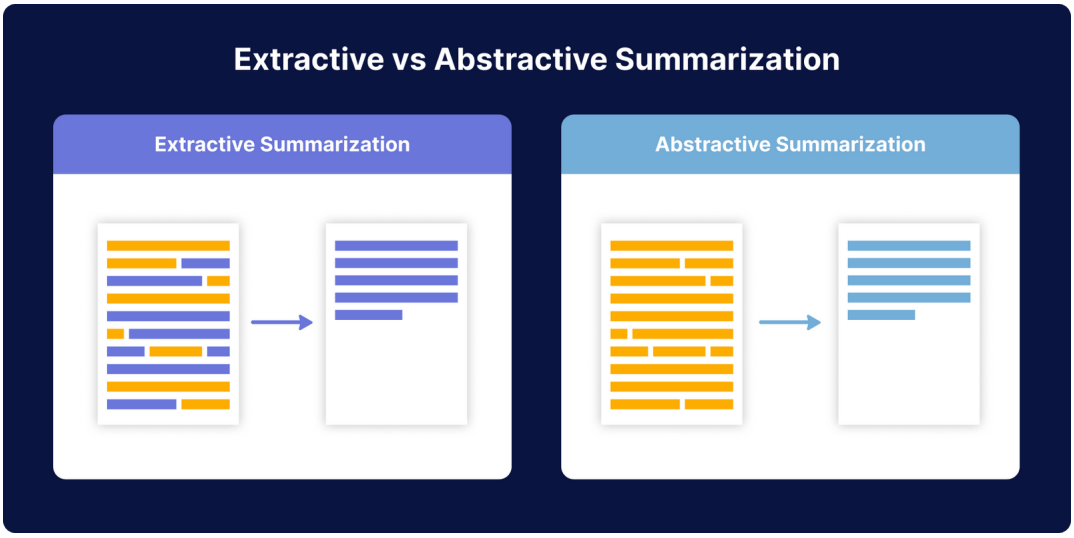

Figure 1: Extractive and Abstrace Summariation (Source: https://www.abstractivehealth.com/extractive-vs-abstractive-summarization-in-healthcare)

Text summarization can categorized into many types based on vary different conditions, but it can be broadly divided into two many types: **Extractive summarization** and **Abstractive summarization**. The extractive summarization focuses on extracting the set of sentences which are important in the original text. Instead, the abstractive summarization summaries the original text by understanding the meaning of the whole text and based on that, it can produce the summary by its words. Therefore, the abstractive summarization is more likely similar to how one person summaries the document in real life. In this report, we want to focus only on the abstractive summarization which we find that are more interesting.

2 DATASETS

In this report, we use the **NEWS SUMMARY** from the Kaggle platform (Source: [1]) to illustrate the abstractive summarization. It contains the summarized news from Inshorts and only scraped the news articles from Hindu, Indian times and Guardian. Time period ranges from february to august 2017.

This dataset consists of 4515 examples and contains Author_name, Headlines, Url of Article, Short text, Complete Article. But we only focus on the two features which are **Short text** and **Complete Article**. Complete Article contains the whole text from original article and short text is the text summaries the information from that article. Our aims is to build the model can input the whole article and then generate the summary of that article.

3 APPROACH TO THE PROBLEM

3.1 General approach

In general, the abstractive summarization task can be divided into two intersection subtasks which are **NLU(Naltural Languge Understanding)** and **NLG(Naltural Language Generation)**. NLU focus on techniques help us to understand the semantics of text and represent it in the specific form which computer can understand (vector,...). Based on semantics vector we found in NLU, NLG focus on the method which can generate the text. Not only in summarization, many other tasks in NLP also contain those tasks. For specific, below is the general workflow of abstractive summerization model:

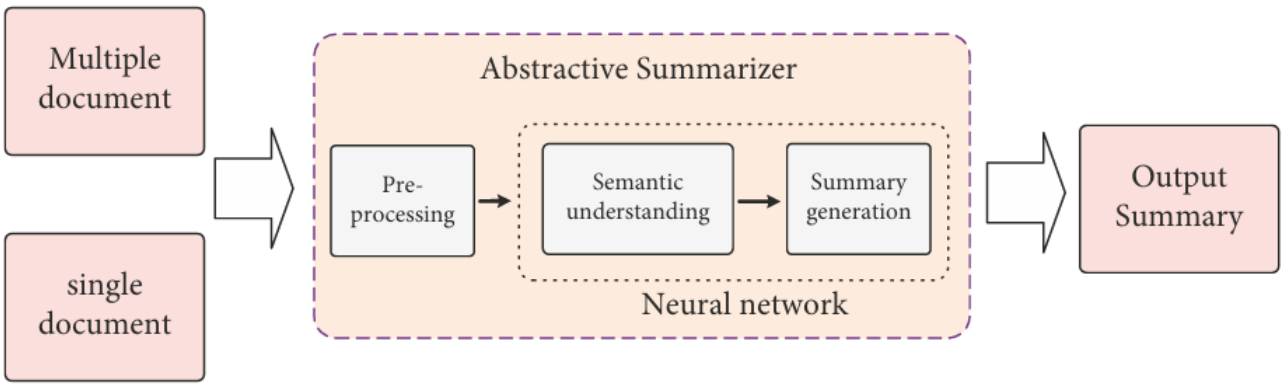

Figure 2: General workflow of abstractive summerization model. Source: [2]

In above workflow, the **Processing step** contains the linguistic technology to structure the input text(Sentence segmentation, word tokenization, stop-word removal,...). The **Semantic understanding** use the neural network to recognize and represent the deep semantics of the input text through fusion vector. Finally, the **Summary Generating** will map the fusion vector to the vocabulary space, then generate the summary.

From the general scheme, many researchers invent models which are constructed as the encoder and decoder architecture. The model we use following also has that architecture.

3.2 Bart model

Bart model [4] is the model which pre-trains a model combining Bidirectional and Auto-Regressive Transformers. Bart contains BERT model as the encoder and GPT as the decoder. It uses the standard sequence-to-sequence Transformer architecture from (Vaswani et al., 2017) [3], except, following GPT, that we modify ReLU activation functions to GeLUs. It pre-trained the task of reconstructing the denoising text go back to the original text.

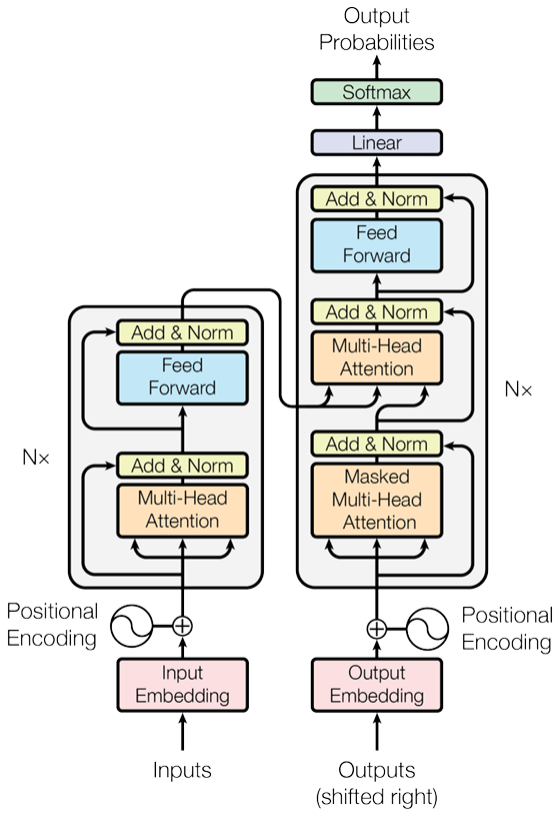

Figure 3: Transformer Architecture. Source : [3]

There are several transformations to corrupt text: Token Masking, Sentence Permutation, Document Rotation, Token Deletion and Text Infilling. The task in the bart model is to use seq2seq model to reconstruct the original text. From pre-training on the above task, bart

model will have general information about the structure of document, sentence or word and we can fine-tune model further for the downstream task.

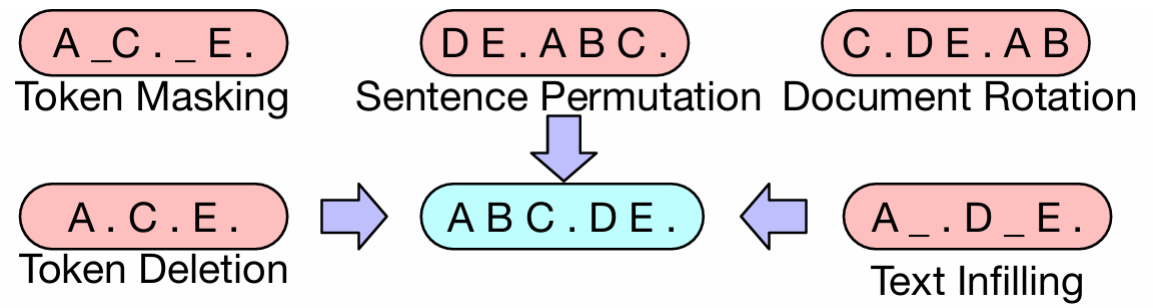

Figure 4: Transformations for noising the text. Source: [4]

In this report we choose bart model for abstractive text summarization since bart model has autoregressive decoder, it can be directly fine-tuned for text generation task such as text summarization. Moreover, with the task of the summarization, we can think the full text as the corrupted version summary text, so we need to reconstruct original output text, then it's very similar to the pre-trained task in bart.

3.3 Fine-tuning model

Fine-tuning is one of the most used method in machine learning nowaday. Instead of training from scratch as before, fine-tuning will adapt a pre-trained model for specific tasks. The model will be pre-trained on the very large data and be published public by organizations or person. Fine-tuning will use the weights of a pre-trained model as a starting point for further training on a smaller dataset of examples that more directly reflect the specific tasks and use cases the model will be utilized for.

There are many ways to fine-tune such as: Full fine-tuning, Parameter efficient fine-tuning(PEFT),... Full fine-tuning refers to train the entire network and update all the parameter in the model.Alternatively, PEFT reduce the number of updated parameter and will decrease the computational resources and usually be more stable than full fine-tuning. In this report, we want to try all the methods to fine-tune and compare result to each others.

4 EVALUATION

In general, the text generation task or specifically, in the text summarization, it is very hard to evaluate the accuracy of the text which is generated by the model since the naltural language is diversity and no specific rule to tell which text is wrong or correct unless reading it. Reading is maybe the easy task for human but for computer, it's impossible for it to read word by word. In the following section, we will introduce the statistics method to measure the generated text and it maybe no the exactly accurate to evaluate but it's good reference for us to tell which generated text is good or not.

4.1 Family of ROUGE Score

ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation. It is essentially of a set of metrics for evaluating automatic summarization of texts. It works by comparing an automatically produced summary or translation against a set of reference summaries.

$$\frac{number\_of\_overlapping\_words}{total\_words\_in\_reference\_summary}$$

(a) Recall formula

$$\frac{number\_of\_overlapping\_words}{total\_words\_in\_system\_summary}$$

(b) Precision formula

ROUGE computes the overlapping words between the generated summary and reference summary with different granularity corresponding to different ROUGE scores: ROUGE-N, ROUGE-L, ROUGE-S. In all kinds of ROUGE score, they compute the precision, recall and f1-score using overlapping. Recall in the context of ROUGE simply ROUGE simply means how much of the reference summary is the generated summary recovering or capturing? It is computed by dividing the number of overlapping words in total words in reference summary. However, it does not tell you the other side of the story. A machine generated summary can be extremely long, capturing all words in the reference summary. But, much of the words in the system summary may be useless, making the summary unnecessarily verbose. This is where precision comes into play. In terms of precision, what you are essentially measuring is, how much of the system summary was in fact relevant or needed? Precision is measured as: dividing the number of overlapping words in total words in generated summary. Each will tell the different story so to harmonize both precision and recall, we compute the f1-score.

Now we will tell specifically what things does the kind of ROUGE score compute? ROUGE-N measures unigram - ROUGE-1, bigram - ROUGE-2, trigram - ROUGE-3 and higher order n-gram overlap. We compute the precision, recall and f1-score of overlapping of single word or pairs of words,... Alternatively, the ROUGE-L measures the overlapping of longest matching

sequence of words using Longest common sequence (LCS). An advantage of using LCS is that it does not require consecutive matches but in-sequence matches that reflect sentence level word order.Finally, ROUGE-S measures any pair of word in a sentence in order, allowing for arbitrary gaps. This can also be called skip-gram co-ocurrence.

In this report, we will only use the common ones in ROUGE: ROUGE-1, ROUGE-2, ROUGE-LSUM, ROUGE-L to evaluate the quality of the generated summary.

4.2 BERT Score

In the ROUGE score, while this provides a simple and general measure, it fails to account for meaning-preserving lexical and compositional diversity. The BERTSCORE introduced in [7], which is a language generation evaluation metric based on pre-trained BERT contextual embeddings. BERTSCORE computes the similarity of two sentences as a sum of cosine similarities between their tokens' embeddings.

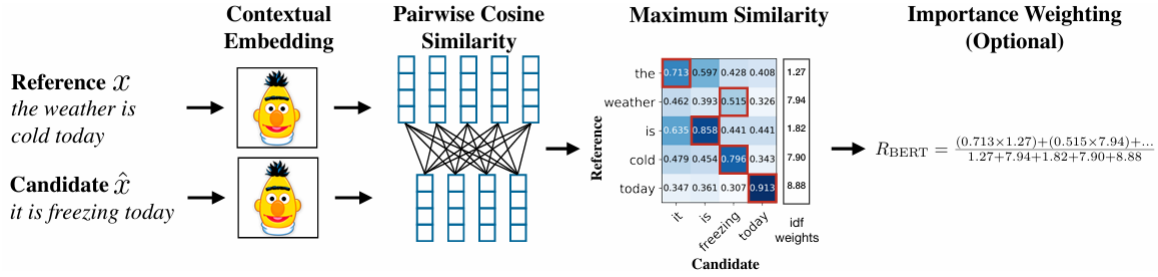

Figure 6: The workflow to compute recall bert score

Unlike the above method, with BERTSCORE we can measure the semantics between the generated summary and reference summary based in the fact that with bert contextual embeddings, the vector represent for words(token) also contains semantics information. In the original paper they use greedy matching to maximize the matching similarity score, where each token is matched to the most similar token in the other sentence.Then they combine precision and recall to compute an F1 measure.

$$R_{\text{BERT}} = \frac{1}{|x|}\sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} \mathbf{x}_i^\top \hat{\mathbf{x}}_j \quad , \quad P_{\text{BERT}} = \frac{1}{|\hat{x}|}\sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} \mathbf{x}_i^\top \hat{\mathbf{x}}_j \quad , \quad F_{\text{BERT}} = 2\frac{P_{\text{BERT}} \cdot R_{\text{BERT}}}{P_{\text{BERT}} + R_{\text{BERT}}} \quad .$$

Figure 7: The way to compute recall, precision and f1-score

Optionally, BERTSCORE also enable us to incorporate importance weighting. In their paper, they experiment with inverse document frequency (idf) scores computed from the test

corpus. Given M reference sentences $\{x^i\}_{i=1}^{M}$. Then the idf score of token w is computed as:

$$idf(w) = -log(\frac{1}{M}\sum_{i=1}^{M} I[w \in x^{(i)}])$$

Now the recall BERTSCORE can be recomputed as:

$$R_{BERT} = \frac{\sum_{x_i \in x} idf(x_i) max_{\hat{x}_j \in \hat{x}} x_i \hat{x_j}}{\sum_{x_i \in x} idf(x_i)}$$

There are many variants of pre-trained BERT contextual embeddings to choose and in this report, we choose model **microsoft/deberta-xlarge-mnli**,which is the model have the best correlation with human evaluation[8].

5 EXPERIMENT AND RESULTS

5.1 Experiment

To easily access the News Summary datasets and utilize the free GPU resources, we use notebook in Kaggle website to fine-tune model. In this report, we will choose the pre-trained model: **sshleifer/distilbart-xsum-12-3**[9] because it has less paramater than the original bart but still work efficiently and it has been already fine-tuned on the Extreme Summarization so it also has the general information about summarization, that help the our fine-tuning process be faster.

We use the Trainer API from transformers hugging face library to fine-tune model easier and more efficiently. We are fine-tuning with following configurations:

- Train and validation batch size: 4
- Learning_rate: 3e-05
- Optimizer: Adam with betas=(0.9,0.999) and epsilon=1e08
- Lr_scheduler_type: linear
- Lr_scheduler_warmup_steps: 500
- Num_epochs: 5
- Label_smoothing_factor: 0.1

  Our evaluation strategy is compute Cross-entropy loss on the test data and compute the rouge score of the generated summary for test data at the end of each epoch.

5.2 Results

Below is the training loss, validation loss and rouge score in each epoch:

Epoch	Training Loss	Validation Loss	Rouge-1	Rouge-2	Rouge-L	Rouge_LSum
1	3.4812	3.3209	47.7226	26.3282	35.5063	42.5426
2	3.2269	3.1838	50.4271	27.7047	37.2638	45.1897
3	2.9504	3.1401	50.6362	28.2773	37.6	45.4901
4	2.8014	3.1346	51.2942	28.4684	38.0877	46.0386
5	2.71	3.1426	51.2701	28.3575	37.9263	45.8934

Below is the actual performance of the model on the article from the source: Ronaldo threatens to punch referee after getting sent off :

"Portuguese striker Cristiano Ronaldo was sent off for the 12th time in his career after his elbow hit Al Nassr's defender Ali Al-Bulaihi's neck in the 86th minute of the Saudi Super Cup on Monday night. Ronaldo raised his fist towards referee Mohammed Al-Hoaish and intended to punch him. The referee gave him a red card and sent him off."

6 FUTURE WORK

Although model work quite good but in many articles, performance of model is very bad, especially the long article. Moreover, the hyper-parameter we specify in the fine-tuning process is just random, so our next aims is to find the best sets of hyper-parameter for fine-tuning. Next, about the evaluation, as we mentions above Rouge score actually can't evaluate the semantics information of the generated summary, so we will find more other metrics to evaluate model in many perspectives. Additionally, we want to find the evaluation method that doesn't need the reference summary since we want to evaluate our generation with the text which does't have the reference summary.

7 REFERENCE

References

[1] Kondalarao vonteru : NEWS SUMMARY

[2] Mengli Zhang,Gang Zhou,Wanting Yu,Ningbo Huang,and Wenfen Liu(2021): A Comprehensive Survey of Abstractive Text Summarization Based on Deep Learning

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

[4] 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

[5] Kavita Ganesan: What is ROUGE and how it works for evaluation of summaries?

[6] Chin-Yew Lin(2004): ROUGE: A Package for Automatic Evaluation of Summaries

[7] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, Yoav Artzi(2020): BERTScore: Evaluating Text Generation with BERT

[8] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, Yoav Artzi(2020): BERT score for text generation

[9] sshleifer/distilbart-xsum-12-3