Hanoi University of Science and Technology
School of Electrical and Electronic Engineering

Abstractive Text Summarization

Chu Quang Tung 20213594
Phung Minh Chien 20213565
Instructor: Ph.D Do Thi Ngoc Diep

December 13, 2024

ACKNOWLEDGEMENT

We would like to express our appreciation to Ph.D Do Thi Ngoc Diep who gave us the opportunity to work on this project. Your invaluable advice, insightful feedback, and thorough guidance have played a pivotal role in our learning journey. Through this project, we have gained a deeper understanding of preprocessing pipelines for text summarization, including data cleaning, tokenization, and GPU optimization. We have also improved our skills in managing large-scale datasets, implementing model training workflows, and evaluating performance metrics such as ROUGE scores. This work would not have been possible without your continuous support and mentorship, which have greatly enriched our knowledge and skills.

ABSTRACT

In the rapidly advancing field of natural language processing (NLP), the utility of pretrained large language models (LLMs) for specialized tasks has garnered significant interest. This project report delves into the application of models, MBart, ViT5 which are fine-tuned to perform the downstream task of text summarization. Leveraging their extensive pre-existing linguistic knowledge embedded during pretraining, these models are hypothesized to be highly effective for summarization tasks. We employ a robust training regime using our own scraping datasets, to fine-tune these models specifically for summarizing Vietnamese legal and criminal news texts. The performances of these above models are rigorously evaluated using the ROUGE score, which measures the overlap of n-grams between the generated summaries and reference summaries. This report not only presents a comparative analysis of the effectiveness of these models in generating concise and coherent summaries but also discusses the implications of our findings for the development of more sophisticated text summarization tools. Additionally, we explore the challenges encountered during the fine-tuning process and propose recommendations for future research endeavors in this area.

Contents

Chapter 1

Introduction

In today's digital age, there is a vast and ever-growing amount of text-based data available across various platforms such as websites, blogs, news articles, social media posts and online discussions. Additionally, extensive textual content can be found in articles, books, novels, legal documents, scientific papers and biomedical literature. This abundance of textual information has led to a significant problem known as information overload, where users are inundated with an overwhelming amount of data. Consequently, users often find themselves spending a considerable amount of time navigating through this sea of text, trying to sift through and extract relevant information, which severely impacts their productivity and efficiency. As a result, the urgent and fundamental challenge lies in efficiently locating and summarizing the necessary information from text resources. Manual summarization involves meticulously browsing through the entire content and condensing it, which is both time-consuming and prone to getting lost in the vast amount of data. Automatic text summarization (ATS) offers an effective solution to address this issue. ATS aims to automatically generate a concise and readable summary containing the core contents of the input text. It is becoming more and more important for solving how to obtain required information quickly, reliably, and efficiently. Generally, there are two prominent summarization systems based on the way the summaries are generated: extractive summarization and abstractive summarization. Abstractive text summarization is the task of generating a short and concise summary that captures the salient ideas of the source text. The generated summaries potentially contain new phrases and sentences that may not appear in the source text. Abstractive systems need to first understand the semantics of the text, and then employ the algorithm of natural language generation (NLG) to generate a more concise summary using paraphrase, synonymous substitution, sentence compression,... In this work, we aim to fine-tune models, which is pretrained on a data-rich task, on own scraping dataset for legal and criminal news. After that, we will use Rouge score to compare the performance of our model.

Chapter 2

Related Works

2.1 Transformer Architecture

The Transformer architecture, introduced in the seminal paper "Attention is All You Need" by Vaswani et al. in 2017, represents a major breakthrough in machine learning for processing sequences. Unlike traditional models that processed data sequentially (like RNNs and LSTMs), the Transformer uses a novel structure that processes data in parallel, significantly improving efficiency and performance.

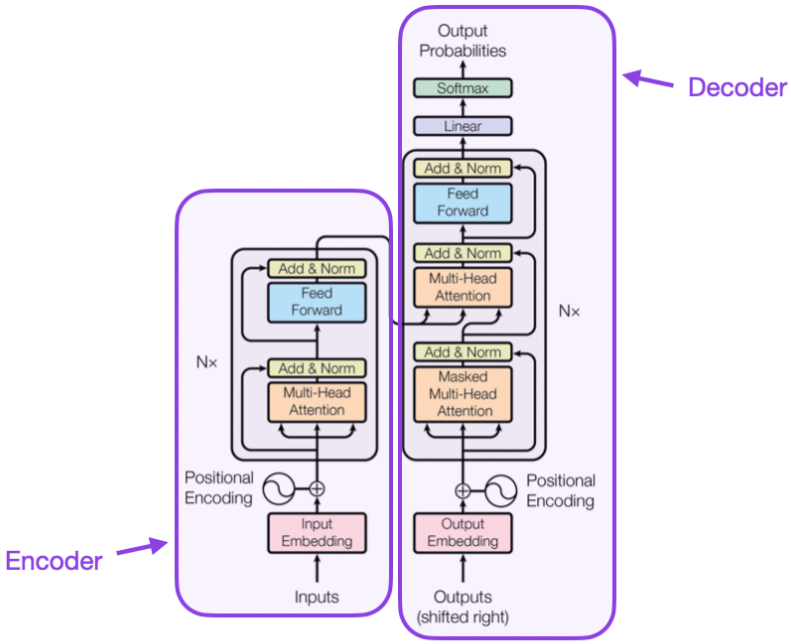

Figure 1: The Transformer - model architecture.

Figure 2.1: Transformer Architecture

- **Core Components of the Transformer**

  - Self-Attention Mechanism

  The self-attention mechanism is at the heart of the Transformer and allows the model to dynamically focus on different parts of the input sequence, assessing the importance of each word in relation to every other word. This mechanism calculates attention scores using query, key, and value vectors

derived from the input data, which help determine how much focus should be directed towards other parts of the input sequence during processing.

- **Multi-Head Attention**

This feature extends the self-attention mechanism by running it multiple times in parallel. Each "head" in the multi-head attention mechanism can focus on different parts of the input sequence, allowing the model to capture a richer understanding of the context. This is crucial for complex language understanding tasks where multiple relationships and meanings may exist simultaneously within the text.

- **Positional Encoding**

Since the Transformer does not inherently process sequential data in order, positional encodings are added to give the model some information about the relative or absolute positioning of the tokens in the sequence. This ensures that the order of words and their syntactic and semantic relationships are preserved and considered during model training and inference.

- **Layer Structure**

The Transformer model consists of an encoder and decoder, each comprising multiple layers of the same two core sub-layers: a multi-head attention layer followed by a position-wise fully connected feed-forward network. Each sub-layer has a residual connection around it followed by layer normalization, which helps in stabilizing the learning process and improving the convergence speed.

Transformers have revolutionized the processing of sequences in machine learning by introducing parallel processing capabilities, which allow for the entire sequence to be processed simultaneously, unlike Recurrent Neural Networks (RNNs) that handle data sequentially. This capability not only helps Transformers manage long-range dependencies more effectively by attending to all parts of the sequence concurrently, but it also avoids the vanishing gradient problem that plagues RNNs. The efficiency and speed of Transformers are further enhanced by their ability to reduce computational complexity and training times, which is especially advantageous when handling large datasets. This leads to quicker model iterations and enhancements. Furthermore, the self-attention mechanism within Transformers facilitates a dynamic, context-aware representation of data, enabling the model to process information from both directions of a sequence simultaneously, unlike RNNs that are limited to processing from left to right or vice versa. This ability to understand and generate contextually appropriate outputs has been extended beyond natural language processing (NLP) to tasks like image recognition and music generation, showcasing the model's scalability and flexibility. In performance benchmarks, Transformers consistently outperform RNNs, with architectures like BERT and GPT setting new standards for machine understanding of language across various tasks, including translation and text generation. The transformative impact of the Transformer architecture represents a significant shift in sequence processing, addressing the limitations of previous models and paving the way for advanced AI applications across diverse fields.

2.2 Vietnamese Text-to-Text Transfer Transformer Model (T5)

T5, or the "Text-to-Text Transfer Transformer," built on the innovative Transformer architecture, has marked a paradigm shift in how natural language processing (NLP) tasks are approached and executed. Unlike traditional models that relied on recurrent neural networks (RNNs), T5 employs a sophisticated encoder-decoder structure using self-attention layers that process data simultaneously rather than sequentially. This fundamental change drastically enhances the speed and efficiency of the model, enabling it to decode complex relationships and dependencies between words across a given text with unprecedented accuracy.

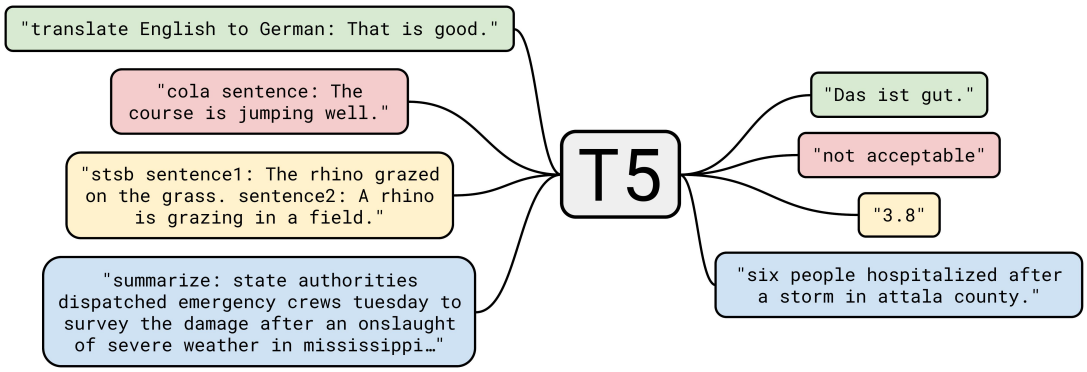

Figure 2.2: T5 Explained

One of the key innovations of T5 is its unified text-to-text framework, which simplifies the way NLP tasks are built and solved. By converting all tasks—whether translation, summarization, text classification, or question answering—into a unified format for converting input text to output text, T5 streamlines the process of training and deploying NLP models. This text-to-text framework has inspired many adaptations designed to target specific languagesand domains. Among these adaptations, ViT5, a variant of T5 tuned specifically for Vietnamese, stands out as a transformation model in the field of Vietnamese NLP.

ViT5 is built on the powerful architecture of T5 but is pre-trained and meticulously tuned using large Vietnamese datasets to handle the unique linguistic features of Vietnamese. Like its predecessor, ViT5 uses a time-varying task in pre-training, in which text segments are masked randomly and the model learns to predict these masked segments. However, the datasets used for ViT5 emphasize Vietnamese syntax, semantics, and vocabulary, allowing the model to gain a deep understanding of the structure and cultural context of the language.

ViT5 enables solving a wide range of Vietnamese NLP tasks, including summarization, translation, text classification, and question answering, with high efficiency and accuracy. During fine-tuning, ViT5 demonstrates remarkable adaptability to task-specific datasets, optimizing its performance for diverse Vietnamese NLP applications. This adaptability stems from the inherent flexibility of the T5 architecture, which can accommodate a variety of text tasks without significant architectural modifications. ViT5 extends this adaptability further by scaling across different configurations, from lightweight models suited for low-resource scenarios to larger models that leverage powerful computational resources to deliver cutting-edge

results.

Since its launch, ViT5 has had a profound impact on Vietnamese NLP, addressing challenges previously constrained by limited resources and the complexity of the language. By leveraging a powerful text-to-text model and adapting it to the nuances of Vietnamese, ViT5 has set a new benchmark for language-specific NLP models. It has proven to be a vital tool for both researchers and practitioners, enabling breakthroughs in machine translation, document summarization, and natural language understanding for Vietnamese texts.

2.3 Multilingual Bidirectional and Auto-Regressive Transformer Model (MBART)

mBART (Multilingual Bidirectional and Auto-Regressive Transformer) is a cutting-edge model in multilingual natural language processing (NLP), designed by Facebook. It extends the architecture of BART to support multilingual text, effectively combining the capabilities of bidirectional encoders and autoregressive decoders for multilingual understanding and generation. This integration makes mBART a versatile tool for handling diverse linguistic tasks across multiple languages.

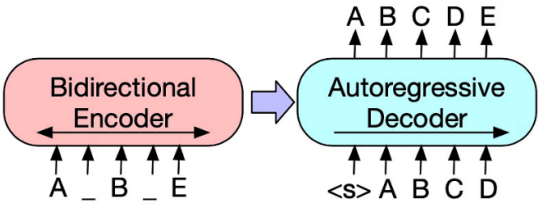

Figure 2.3: BART model

The architecture of mBART utilizes the self-attention mechanism intrinsic to transformer models, enabling it to evaluate the importance of various words in an input sequence while maintaining contextual integrity. By processing inputs bidirectionally through its encoder, mBART captures intricate dependencies and contextual relationships between words. Its autoregressive decoder, on the other hand, ensures the generated text is coherent, grammatically accurate, and contextually aligned with the input.
This multilingual adaptability allows mBART to excel in cross-lingual tasks, such as translating or summarizing texts from one language to another, with minimal fine-tuning. Its robust transformer-based design, enriched with multilingual pretraining, solidifies mBART's position as a pioneering model for multilingual NLP.
MBART employs a distinct pre-training regimen that sets it apart from traditional language models, using a combination of masked language modeling and next sentence prediction to enhance its linguistic capabilities. This pre-training involves deliberately corrupting text by replacing random spans of text, not just individual tokens, with a single [MASK] token—a technique that broadens the context required for predictions and encourages a deeper understanding of language structure. The model then learns to accurately predict these masked spans from the uncorrupted surrounding text, which helps it grasp the semantics, syntax, and narrative structure of the language.

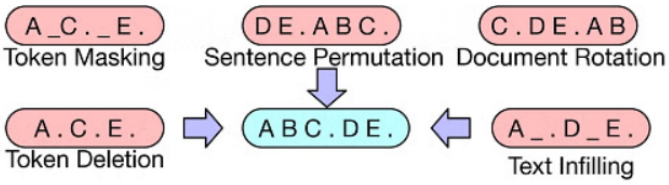

Figure 2.4: BART noising schemes

In addition to masked language modeling, MBART also integrates next sentence prediction into its pre-training phase. This involves presenting the model with pairs of sentences where it must predict if the second sentence in a pair is the actual succeeding sentence in the original text or a randomly chosen sentence. This training task compels MBART to develop a nuanced understanding of how ideas connect across sentences, enhancing its ability to follow and generate coherent narrative flows.

These combined strategies equip mBART with a robust set of capabilities for understanding context and generating text that logically follows from given inputs. The dual approach not only improves the model's accuracy in generating text but also its effectiveness in tasks that require a deep understanding of textual relationships, such as summarizing long documents, answering questions based on extended passages, or even generating cohesive text where maintaining narrative continuity is essential. By training mBART to predict both within and between sentences, the model develops a comprehensive proficiency in handling a wide range of complex language processing tasks, setting a new standard for versatility and performance in NLP models.

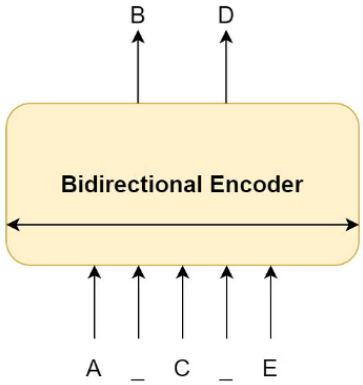

Figure 2.5: Bidirectional encoder

MBART can be effectively adapted for a wide variety of NLP tasks beyond its initial training through a process called fine-tuning. In fine-tuning, mBART is trained on task-specific datasets, allowing it to apply its pre-learned linguistic capabilities to new domains or challenges. For instance, to use mBART for abstractive text summarization, it would be fine-tuned on a dataset containing pairs of long texts and their corresponding summaries. This specialized training helps mBART learn the nuances of summarizing content, enabling it to generate concise, informative, and coherent summaries. Similarly, for tasks like sentiment analysis or question answering, mBART would be trained on relevant datasets that help it learn to classify emotional tones or respond to queries based on context provided. This flexibility to

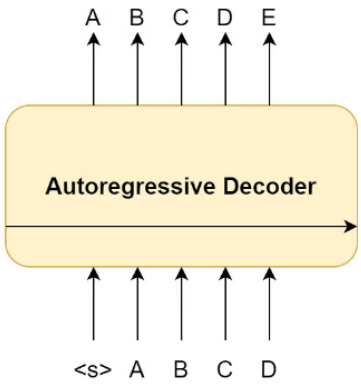

Figure 2.6: Autoregressive model

adapt through fine-tuning makes mBART a highly versatile tool, capable of handling everything from text generation to more structured tasks like classification or translation, effectively tailoring its broad foundational knowledge to specific applications.

Chapter 3

Dataset Description

3.1 Dataset Description

3.1.1 Introduction

- Source: CNN - DailyMail
- Size: 2.51 GB
- Directory structure:

  - train set: 287k rows
  - validation set: 13.4k rows
  - test set: 11.5k rows

- Source: VNN - VNEx
- Size: 136 MB
- Directory structure:

  - train set: 30970 rows
  - validation set: 4426 rows
  - test set: 8849 rows

The CNN/DailyMail dataset is a widely-used English-language dataset in the field of natural language processing and deep learning. It comprises a large collection of news articles and stories from two major news websites: CNN and DailyMail. The dataset covers diverse topics such as politics, business, sports, education, and more. Each item in the dataset typically includes a headline, the main body of the article, and a summary (if available). The dataset is commonly used for tasks such as extractive summarization and abstractive summarization, where the goal is to extract or generate concise summaries from the original articles. Researchers and practitioners leverage this dataset extensively for training and evaluating machine learning and deep learning models in natural language understanding tasks. Overall, the CNN/DailyMail

dataset serves as a valuable resource for the research community, providing a rich and diverse collection of text data for advancing the state-of-the-art in natural language processing and text summarization.
Our own dataset tries to mimic the same structure of CNN/DailyMail dataset, we scrap the data from 2 Vietnam major news website: VNExpress and Vietnamnet. However, our dataset only covers news related to laws and crimes.
Each article is accompanied by several highlight sentences. These highlights are curated and are used as reference summaries in summarization tasks.

article string · lengths	highlights string · lengths
1.64k 3.2… 32.…	14 752 99.…
LONDON, England (Reuters) -- Harry Potter star Daniel Radcliffe gains access to a reported £20 million ($41.1 million) fortune as he turns 18 on Monday, but he insists the money won't cast a spell on him. Daniel Radcliffe as Harry Potter in "Harry Potter and the Order of the Phoenix" To the disappointment of gossip columnists around the world, the young actor says he has no plans to fritter his cash away on fast cars, drink and celebrity parties. "I don't plan to be one of those people who, as soon as they turn 18, suddenly buy themselves a massive sports car collection or something similar," he told an Australian interviewer earlier this month. "I don't think I'll be particularly extravagant	Harry Potter star Daniel Radcliffe gets £20M fortune as he turns 18 Monday . Young actor says he has no plans to fritter his cash away . Radcliffe's earnings from first five Potter films have been held in trust fund .

Figure 3.1: Dataset CNN/DailyMail example

Summarize	Content
30963 unique values	30969 unique values
Người đưa, môi giới hoặc nhận "hối lộ tình dục", "đổi tình lấy chức"… bị xử lý hình sự như nhận tài ...	Người đưa, môi giới hoặc nhận "hối lộ tình dục", "đổi tình lấy chức"… bị xử lý hình sự như nhận tài ...
Nghĩ em vợ hỗn láo với mình, Bậc đã về nhà lấy một con dao bầu rồi truy sát Mạnh, kết quả là Mạnh đã...	- Nghĩ em vợ hỗn láo với mình, Bậc đã về nhà lấy một con dao bầu rồi truy sát Mạnh, kết quả là Mạnh...
Với thời_gian tiến_hành từ 8h sáng tới 17h chiều , buổi khám_xét nhà Bùi_Văn_Công có_lẽ sẽ đi vào kỷ...	Sáng nay 23/3 , Cơ_quan cảnh_sát điều_tra Công_an tỉnh Điện_Biên phối_hợp cùng VKSND tỉnh đã tổ_chức...
Giáo_hội Phật_giáo Việt_Nam cho_biết sẽ triển_khai nội_dung đề_nghị bỏ_tục đốt vàng_mã sâu_rộng hơn ...	Sau Công_văn 31 của Trung_ương Giáo_hội Phật_giáo Việt_Nam , thông_tin từ Hoà_thượng Thích_Huệ_Thông...

Figure 3.2: Dataset VNN/VNEx example

3.1.2 Dataset Challenges

However, NLP datasets present various challenges that make the task of summarization and other NLP tasks complex and demanding. Here are some of these challenges:

- **Length Variability**
  - **Inconsistent Context Lengths:** Long content texts require the model to understand and sum-

marize information from a much larger context window, which can exceed the input length limit of many models (e.g., 512 or 1024 tokens for transformer-based architectures).
- **Truncation Risks:** When content is longer than the model's maximum input length, key information may be lost during truncation, leading to incomplete or irrelevant summaries.
- **Summary Length Control:** The dataset may contain summaries (Summarize) of inconsistent lengths, which can cause the model to produce excessively long or overly brief summaries that do not adhere to the required summary style.
- **Imbalanced Representation::** Shorter samples dominate the dataset, potentially biasing the model toward shorter summaries rather than addressing complex long-text summarization.

- **Complexity of Language and Vocabulary/Syntax**
  Vietnamese presents unique linguistic challenges due to its vocabulary and syntax, which impact both input encoding and output generation: - **Diverse Writing Styles:** News articles exhibit a wide range of writing styles, from formal and technical to narrative and conversational. This diversity requires models to be flexible and adaptive in understanding and generating text across different styles.
  - **Tonal Language:** Vietnamese uses diacritical marks extensively, and small changes in tone or accent marks can significantly alter the meaning of a word. For example, the words "ma," "mã," and "má" have entirely different meanings.
  - **Morphological Complexity:** Vietnamese does not use morphological inflections like some Western languages (e.g., no conjugations), but compound words and phrasing are highly variable, creating complex sentence structures.
  - **Vocabulary Richness:** The dataset may contain a wide range of domain-specific terms, idiomatic expressions, and colloquialisms, which can be difficult for the model to learn, especially if there are rare or low-frequency tokens.
  - **Lack of Delimiters:** Unlike English, where sentences are often separated by periods or commas, Vietnamese relies heavily on context and phrase structure, making sentence boundary detection more difficult.
  - **Code-Mixing and Abbreviations:** There may also be instances of abbreviations, slang, or mixed usage of formal and informal language, adding further noise to the dataset.

To address these challenges, several approaches can be employed:

- **Advanced Architectures:** Using advanced neural network architectures, such as transformers (e.g., BARTPho, MBART, ViT5), which can handle long-range dependencies and complex syntactic structures more effectively.
- **Pre-training and Fine-tuning:** Leveraging large-scale pre-training on diverse corpora followed by fine-tuning on specific tasks can help models learn robust representations that generalize well across different styles and domains.

- **Data Augmentation:** Employing data augmentation techniques to create more diverse training examples can help models become more robust to variations in vocabulary and syntax.

3.2 Data Scraping

3.2.1 Website Selection: Vietnamnet and VN Express

- **Vietnamnet**
  - Popularity and Credibility: Vietnamnet is a highly reputable online news platform that covers a wide range of topics, including detailed and well-researched legal news.
  - Comprehensive Coverage: The platform provides rich and detailed legal reports, articles on court cases, and updates on legislative changes.
  - Consistent Formatting: Vietnamnet maintains a relatively consistent structure for its articles, which facilitates effective web scraping.
- **VNExpress**
  - High Readership and Authority: VNExpress is one of the most-read Vietnamese online news portals, with extensive coverage of national and international news.
  - Focus on Timely Legal Updates: VNExpress often publishes legal updates and reports on crime-related cases, making it an ideal source for obtaining comprehensive legal news.
  - Structured News Sections: Similar to Vietnamnet, VNExpress has a structured and categorized format that simplifies the data extraction process.

  By combining data from these two platforms, the dataset will cover a wide range of legal and crime-related content, leading to a more diverse and robust dataset for abstractive text summarization.

3.2.2 Rationale for Choosing the Law and Crime News Section

- Rich in Context and Detail: Law and crime news articles are typically detailed, providing a comprehensive narrative that is ideal for abstractive summarization.
- Need for Concise Summaries: These articles are often long and dense, making them prime candidates for generating concise and informative summaries.
- Consistency in Structure: Most legal and crime news articles follow a clear structure, making it easier to extract meaningful summaries.
- Consistency in Structure: Most legal and crime news articles follow a clear structure, making it easier to extract meaningful summaries.
- Potential Applications: Summarizing legal news can have practical applications in legal research, public information services, and judicial case reviews.

3.2.3 Data Scraping Methodology

The data collection process involves extracting information from websites and converting that information into a structured format for further analysis or storage. The provided script focuses on collecting legal news articles from the VietnamNet website, using a systematic approach to ensure accurate and efficient data collection.

- **Target Website Structure:** The script targets the VietnamNet "Pháp luật" section, utilizing paginated URLs (`https://vietnamnet.vn/phap-luat-page{page_num}`) to collect articles from multiple pages.
- **Data Extraction:**
  - Articles Identification: Headlines and article links are extracted from `<h2>` or `<h3>` tags with the `vnn-title` CSS class.
  - Content Retrieval: For each article, the script fetches the full content, including the summary (from `content-detail-sapo` class) and detailed text (from `maincontent main-content` class).
- **Error Handling:**
  - The script uses `try-except` blocks to manage potential errors in network requests (e.g., timeouts or invalid URLs) using the `requests` library.
  - Missing or malformed content (such as empty summaries) is handled by skipping invalid entries.
- **Character Escaping:** Non-XML-compliant characters are escaped to ensure compatibility with `Excel's OpenXML` format, preventing errors during data export.
- **Server Etiquette:** A `time.sleep(1)` delay between requests is implemented to avoid overwhelming the server, following responsible scraping practices.
- **Data Storage:**
  - Extracted data is organized into an Excel file using the `openpyxl` library, with each article's summary, content, and link placed in separate columns.
  - The script appends data for each page to the worksheet, making it easy to process large datasets.
- **Scalability and Customization:** The script is scalable to handle multiple pages and can be customized to target different sections or modify the range of data collected.
- **Ethical Considerations:** The program adheres to ethical scraping standards, minimizing the impact on server performance and ensuring compliance with web scraping best practices.

3.3 Data Preprocessing

3.3.1 Exploratory Data Analysis (EDA)

First, we will perform a comprehensive exploratory analysis of our dataset. These below figures show the length distribution of articles and highlights of train, validation and test set respectively.

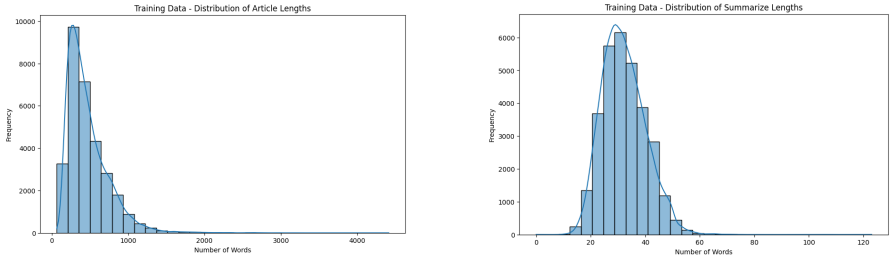

Figure 3.3: Length distribution of articles and highlights in train set

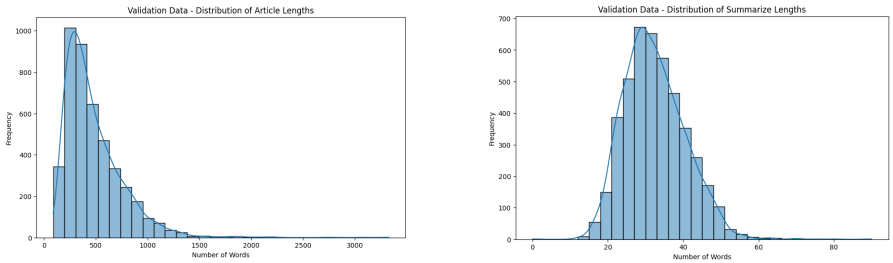

Figure 3.4: Length distribution of articles and highlights in validation set

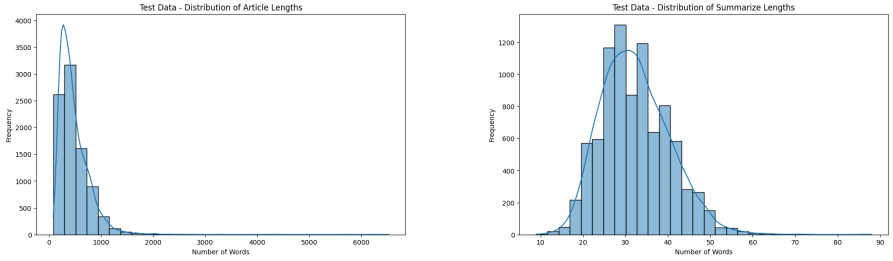

Figure 3.5: Length distribution of articles and highlights in test set

3.3.2 Preprocessing Pipeline

The preprocessing pipeline for the mBART model training ensures that the input data is cleaned, structured, and ready for training and evaluation. This process involves handling missing data, filtering invalid samples, tokenizing the data, and converting it into a format suitable for PyTorch-based training.

Data Cleaning

- **Loading CSV Files:** The datasets (`train.csv, val.csv, and test.csv`) are loaded using `pandas.read`$_csv()$.`Handling Missing Values:`$Anyrowswhere Content or Summarize is NaN(missing$ $[\text{"}Content\text{"},\text{"}Summarize\text{"}])to avoid issues during training.$
- **Handling Short Cells:** Before tokenization, the data is filtered to remove rows with content length smaller than 500 characters. This ensures that only data with sufficient information is used for training, helping the model learn from more meaningful and complete examples.

Dataset Conversion for Training

After preprocessing, the cleaned lists are converted into datasets.Dataset format for compatibility with the Hugging Face Trainer:

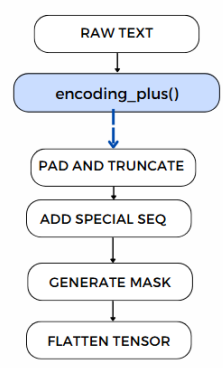

Figure 3.6: Data preprocessing pipeline

The preprocessing step outputs flattened tensors in PyTorch format, which seamlessly integrate with the model's requirements for efficient and accurate processing. These tensors include:

- **input_ids:** This tensor contains the tokenized representation of the input text (i.e., the summary prompt and the content).
- **attention_mask:** This binary mask (comprising 0s and 1s) indicates which tokens should be attended to by the model (1 for real tokens, 0 for padding tokens).
- **labels:** This tensor contains the tokenized version of the summary text, which is used as the ground truth for training.
- **labels_attention_mask:** This binary mask (comprising 0s and 1s) serves a similar role as the input `attention_mask`, ensuring the model focuses on valid tokens in the summary while ignoring padding tokens.

```
Input IDs: tensor([[ 101, 7592, 1010, 2088,  999,  102,    0,    0,    0,    0,    0,    0,
            0,    0,    0,    0,    0,    0,    0,    0]])
Attention Mask: tensor([[1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]])
```

Figure 3.7: text_input_ids and text_attention_mask example

Chapter 4

Implementation Details

In our project, we utilize the PyTorch Lightning framework to fine-tune the ViT5 and mBART model for the task of text summarization.

- **Model Configuration:**

  We initialize the text summarization system with `facebook/mbart-large-50` and `VietAI/vit5-base`, pre-trained multilingual models capable of handling diverse linguistic structures. These models serve as the foundation, leveraging their pre-trained knowledge to adapt to the specific task of generating summaries for input text.

  The forward method is defined to handle the forwarding of inputs through the model, applying the appropriate attention masks and optionally including labels for calculating the loss during training phases. This method returns the loss and the logits, which are essential for both optimizing the model and evaluating its performance.

- **Training Details:**

  Optimization and Learning Rate: The model is fine-tuned using the AdamW optimizer, selected for its robust handling of sparse gradients and effective regularization via weight decay. A learning rate of `5e-5` is used, which strikes a balance between making meaningful adjustments to the pre-trained weights and preserving the knowledge already encoded in the model. This conservative rate ensures that the model remains stable while learning task-specific patterns.

- **Batch Processing:** The training loop processes batches sequentially, where input text is tokenized with a maximum length of 512 tokens and target summaries with a maximum length of 128 tokens. These limits prevent memory overflow and standardize input dimensions. At each training step:
  - The loss is computed and logged, offering immediate feedback on the model's performance.
  - Logging every 10 steps ensures consistent monitoring of metrics, enabling the early detection of issues such as overfitting or underfitting.
  - Continuous evaluation is performed at the end of each epoch to assess the model's generalization to unseen data. This helps determine when the model has achieved optimal performance.

- **Hyperparameter Tuning:**

The fine-tuning process places significant emphasis on selecting appropriate hyperparameters to maximize model performance:

- **Learning Rate:** The chosen value of 5e-5 is based on empirical findings, providing gradual updates to the model parameters without overwhelming the pre-trained knowledge.
- **Weight Decay:** Configured at 0.01, it helps prevent overfitting by penalizing large weight updates.
- **Optimizer Enhancements:** The AdamW optimizer, with its improved weight decay handling, is well-suited for fine-tuning large language models like mBART. It adjusts the learning rate dynamically for each parameter, ensuring stable and efficient training.

These hyperparameters are meticulously selected and fine-tuned to strike a balance between optimizing the model for the text summarization task and preserving the multilingual capabilities of the base mBART model.

- **Training Execution:** The script uses PyTorch Lightning's Trainer object to handle the training process, which simplifies hardware acceleration usage and improves reproducibility. Training is executed with a specified number of epochs, and progress is logged at regular intervals. This script is designed for robustness and flexibility, allowing researchers and developers to experiment with different model configurations and track their experiments efficiently. The use of PyTorch Lightning and wandb enhances the training process by reducing boilerplate code and providing a powerful platform for monitoring and analyzing model performance

By configuring the `facebook/mbart-large-50` and `VietAI/vit5-base` model and with task-specific inputs, carefully chosen hyperparameters, and efficient batch processing, this implementation effectively adapts the model for summarizing text. Logging and checkpointing provide transparency and flexibility throughout the training process, ensuring the model achieves high performance without overfitting.
The end result is a fine-tuned model capable of generating concise and relevant summaries, showcasing the power of adapting a multilingual transformer to a specialized NLP task.

Chapter 5

Experimental Results

5.1 Metrics

ROUGE, which stands for "Recall-Oriented Understudy for Gisting Evaluation", is a set of methods for evaluating the quality of automatically generated text. It is widely used in the NLP community to measure the similarity between automatically generated texts and reference texts, often human-generated summaries.

ROUGE encompasses several measurement methods, including ROUGE-N and ROUGE-L. ROUGE-N focuses on measuring similarity based on words or phrases appearing in both texts. ROUGE-L, on the other hand, emphasizes similarity based on the longest common subsequences of words in the texts, not necessarily in the same order.

Specifically, ROUGE-1 evaluates similarity based on unigrams (single words), ROUGE-2 on bigrams (pairs of adjacent words), and ROUGE-L on longest common subsequences. Formulas for 3 metrics above are:

$$ROUGE-1 = \frac{Count(\text{Overlapping unigrams})}{Count(\text{Total unigrams in reference summary})}$$

$$ROUGE-2 = \frac{Count(\text{Overlapping bigrams})}{Count(\text{Total bigrams in reference summary})}$$

$$ROUGE-L = \frac{Count(\text{Longest common subsequences})}{Count(\text{Total words in reference summary})}$$

Here, "Count" represents the number of occurrences of overlapping n-grams or longest common subsequences between the generated text and the reference text and "Total unigrams/bigrams/words in reference summary" is the total count of unigrams, bigrams or words in the reference summary, respectively.

5.2 Experimental Results

Analysis of Training Phase

In our project, we have trained our models with the hyperparameters:

Learning rate: 5e-5

Number of epochs: 8

Batch size: 8

Additionally, we also use these version for all the models of our project:

- **VietAI/vit5-base:** 226M parameters
- **facebook/mbart-large-50:** 611M parameters

Model Performance

- ViT5: Demonstrates a higher recall rate in ROUGE-1 and ROUGE-L than BART, indicating it might be capturing a broader range of content from the reference summaries. Although it has lower precision scores, the F1 scores are competitive, especially in ROUGE-L, suggesting its strength in understanding longer sequences.
- mBART: Shows varying performance across different metrics, with ROUGE-1 scores indicating moderate effectiveness in capturing unigrams. It has lower scores in ROUGE-2, suggesting difficulties in capturing bigrams effectively. However, it performs better on ROUGE-L, reflecting its capability in capturing longer subsequences.

	Rouge-1	Rouge-2	Rouge-L
ViT5	0.408	0.1860	0.2836
mBART	0.402	0.1804	0.2746

Chapter 6

Conclusion

In conclusion, our project leverages the powerful capabilities of pre-trained large language models (LLMs) like mBART and ViT5 for the downstream task of text summarization. By fine-tuning these models, which are already rich in linguistic information, we harness their advanced understanding of language nuances to generate concise and informative summaries. Our methodology involves using scrapped dataset from VnExpress and VietnamNet news to train and fine-tune these models specifically for summarization in Vietnamese. The performance of mBART and ViT5 is evaluated using the ROUGE score, which quantifies the quality of the generated summaries by comparing them to reference summaries. This approach not only allows us to capitalize on the inherent strengths of each model but also provides a direct comparison of their effectiveness in handling complex summarization tasks. Through this project, we aim to demonstrate the practical benefits of employing sophisticated LLMs in real-world applications, ultimately contributing to the advancement of automated summarization technologies.