VIETNAM NATIONAL UNIVERSITY, HANOI
UNIVERSITY OF ENGINEERING AND TECHNOLOGY


**HIEN VU NGOC**

# A HYBRID MODEL FOR VIETNAMESE MULTI-DOCUMENT SUMMARIZATION

**BACHELOR THESIS**

**Major**: Computer Science

**HA NOI - 2023**

**VIETNAM NATIONAL UNIVERSITY, HANOI**
**UNIVERSITY OF ENGINEERING AND TECHNOLOGY**

**Vu Ngoc Hien**

# A HYBRID MODEL FOR VIETNAMESE MULTI-DOCUMENT SUMMARIZATION

**BACHELOR THESIS**

**Major**: Computer Science

**Supervisor: Dr. Le Hoang Quynh**

**Msc. Can Duy Cat**

**HA NOI - 2023**

# Abstract

Automatic summarization system has been a rapidly developing research area in natural language processing. Various methods and approaches have been proposed with promising results for both extractive and abstractive output. Although there are many summarization models for English documents, researches that apply to Vietnamese document are in the early phases.

This thesis provides research about different techniques for automatic text summarization for multiple Vietnamese documents and how to combine these methods to achieve better results. Studies are made on both extractive and abstractive approaches; this thesis focuses on the data domain of Vietnamese articles, and news data. The applied techniques are graph-based, sentence scoring, and transformer-based pre-trained models to generate extractive and abstractive summaries. During the experiment phases, graph-based strategies prove to contribute the most to the proposed model.

The experimental results showed that the model was effective and the output is grammatical, coherent, and concise with a promising ROUGE score in the evaluation phase.

# Acknowledgements

I would like to express my sincere gratitude and appreciation to Master Can Duy Cat and Doctor Le Hoang Quynh for their invaluable guidance and support throughout the writing of my bachelor report. Their expertise, encouragement, and dedication have been crucial in helping me complete this project. I am truly grateful for their time, patience, and feedback, which have significantly improved the quality of my work. Their mentorship and guidance have not only helped me to achieve my academic goals but also inspired me to continue learning and growing in my field.

I would also like to thank them for creating a positive and engaging learning environment that has made my academic journey a fulfilling and enjoyable one. Their contributions have been instrumental in my academic success and I will always be grateful for their mentorship.

# Declaration

I declare that the thesis has been composed by myself and that the work has not been submitted for any other degree or professional qualification. I confirm that the work submitted is my own, except where work that has formed part of jointly-authored publications has been included. My contribution and those of the other authors to this work have been explicitly indicated below. I confirm that appropriate credit has been given within this thesis where reference has been made to the work of others.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material, I certify that I have obtained written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances in my appendix.

I declare that this thesis has not been submitted for a higher degree to any other University or Institution.

Student

**Vu Ngoc Hien**

# Table of Contents

# Acronyms

| | |
|---|---|
| ATS | Automatic Summarization System |
| GNNs | Graph Neural Networks |
| MMR | Maximal Marginal Relevance |
| NLP | Natural Language Processing |
| SVM | Support Vector Machine |
| TF-IDF | Term Frequency-Inverse Document Frequency |

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Motivation

Automatic text summarization is a problem being studied for many years. The research community has achieved great results and many Automatic Summarization Systems (ATS) have been developed with high applicability. Solving the text summarization problem aims to help humans minimize reading time but still understand the main idea of large and massive documents. Manual summarization is expensive and consumes time and effort; automatic summarization is the solution. The significance of automatic summarization as a problem has been growing rapidly in recent years, mainly due to the explosion in the amount of textual content generated across various domains. With the advent of digital technologies and the internet, the volume of text data has been increasing at an unprecedented rate, and this trend is showing no signs of slowing down. From news articles, scientific publications, and business reports to social media updates, emails, and instant messages, the amount of textual content being produced every day is overwhelming. As a result, the task of manually processing and analyzing this massive amount of data is becoming increasingly challenging and time-consuming for humans. Moreover, the sheer volume of text data can make it difficult for individuals to find the most relevant information that meets their specific needs. Therefore, automatic summarization systems are being developed to help alleviate this problem by providing concise and informative summaries of the key points contained within large volumes of text. These systems can significantly reduce the time and effort required to process and extract useful information from text data, thereby enhancing productivity and improving decision-making in various domains.

The research community has a keen interest in the automatic summarization problem for several reasons. Firstly, as mentioned earlier, the exponential growth in the amount of textual content being generated across various domains. Therefore, there is a pressing need for automated methods that can help to distill the most relevant and useful information from large volumes of text.

Secondly, automatic summarization is a complex problem that requires the integration of several natural language processing techniques, such as text classification, information retrieval, and text generation. Therefore, the development of automatic summarization systems requires significant research and innovation in these areas, as well as in machine learning and artificial intelligence more broadly. Researchers are constantly exploring new approaches and techniques for improving the accuracy and effectiveness of automatic summarization systems.

Thirdly, automatic summarization has a wide range of practical applications across various domains, including news media, business, healthcare, and education. For example, in the news media, automatic summarization systems can be used to quickly and accurately summarize breaking news stories, allowing readers to quickly grasp the key points without having to read lengthy articles. In the business domain, automatic summarization can be used to analyze customer feedback, financial reports, and market research data, enabling companies to make better-informed decisions. In healthcare, automatic summarization can be used to summarize patient records, allowing doctors to quickly identify relevant information and make more accurate diagnoses. These are just a few examples of the many practical applications of automatic summarization that are driving research in this area.

Automatic summarization is also closely related to other NLP tasks such as text summarization, text simplification, and text compression. These tasks involve various techniques for reducing the size of text data while retaining the most important information, which can be useful for a wide range of applications.

A summary of a document is a compressed document that is short with a clear definition and gives the main facts or ideas about the original content. Text summarization can be classified into two types [9] are extractive and abstractive. Extractive summarization is a compressed document from original documents by selecting the most important sentences and combining them to form a complete paragraph. Abstractive summarization is formed by creating new sentences that do not appear in the source text by rephrasing and/or merging original sentences or by rewriting based on original ideas; these new

sentences are usually short, and concise and capture the salient ideas of the source text.

Although various methods and approaches have been proposed, there are few studies on Vietnamese documents (apply for both extractive and abstractive approaches). The reason for this might be the deficiency of Vietnamese textual data. Given these circumstances, new methods, techniques, and improvements are needed.

## 1.2 Problem statement

**Text summarization** is the process of reducing the length of a piece of text while retaining its most important and relevant information. The goal of text summarization is to produce a condensed version of the original text that captures the key points, ideas, and arguments presented in the original document

**Automatic text summarization** can be understood as the process of shortening a set of data computationally to form a short enough summary containing the most important source data information. According to Radev et al. [19], the main objective of an automatic text summarization system is to produce a summary that includes the main ideas in the input document in less space. Extractive summarization and abstractive summarization are two different approaches used in automatic text summarization, which involves reducing a long document or text to a shorter version while preserving its most important information.

**The compression rate** in summarization refers to the degree of reduction in the length or size of the original text that is achieved by the summarization process. In other words, it represents the ratio of the original text's length or size to the summary text's length or size.

Measuring the quality of a summary can be challenging, as it can depend on various factors such as the purpose of the summary, the complexity of the original material, and the intended audience. Generally, a good summary can be considered a brief and concise overview of a text or topic's main ideas and key points. It should accurately and objectively capture the essence of the original material, without including unnecessary details or personal opinions. A well-written summary should effectively communicate the most important information in a clear, readable format and be accessible to a wide audience. A good summary should provide a useful and informative overview of the material, helping readers quickly understand the main ideas and concepts.

Table 1.1 and table 1.2 provides an example of Vietnamese extractive and abstrac-

tive summary generation for single-document and multi-document.

Table 1.1: Example of extractive and abstractive single-document summary

| | |
|---|---|
| **Document** | Thế giới hiện nay đang đối mặt với nhiều thách thức và khó khăn, từ những vấn đề kinh tế và chính trị đến những vấn đề xã hội và môi trường. Tuy nhiên, chúng ta không thể đứng nhìn và chấp nhận điều đó mà cần phải tìm cách đối phó và vượt qua những thử thách này. Một trong những cách để làm điều này là đầu tư vào giáo dục và phát triển con người, vì chúng ta tin rằng con người là nguồn lực quan trọng nhất của thế giới. Ngoài ra, chúng ta cũng cần tìm kiếm những giải pháp đổi mới và sáng tạo để giải quyết các vấn đề đang diễn ra, như sử dụng các công nghệ tiên tiến để tăng cường năng suất và hiệu quả kinh tế. Tuy nhiên, việc tìm kiếm giải pháp đòi hỏi chúng ta phải hợp tác và đồng tâm hiệp lực với nhau, bao gồm cả việc hợp tác quốc tế và hợp tác giữa các tổ chức và cá nhân. Chúng ta cũng cần đảm bảo rằng các quyết định được đưa ra được đánh giá và đưa ra theo cách khoa học và công bằng, để đảm bảo rằng chúng có thể đáp ứng được các nhu cầu và lợi ích của cả nhân loại và hành tinh chúng ta. |
| **Extractive summary** | Thế giới hiện nay đang đối mặt với nhiều thách thức và khó khăn, từ những vấn đề kinh tế và chính trị đến những vấn đề xã hội và môi trường. Tuy nhiên, việc tìm kiếm giải pháp đòi hỏi chúng ta phải hợp tác và đồng tâm hiệp lực với nhau, bao gồm cả việc hợp tác quốc tế và hợp tác giữa các tổ chức và cá nhân. |
| **Abstractive summary** | Thế giới hiện nay đang đối mặt với nhiều thách thức và khó khăn. Để vượt qua những thử thách này, chúng ta cần tìm kiếm giải pháp đổi mới và sáng tạo, sử dụng các công nghệ tiên tiến để tăng cường năng suất và hiệu quả kinh tế. Chúng ta cũng cần đảm bảo rằng các quyết định được đưa ra được đánh giá và đưa ra theo cách khoa học và công bằng, để đáp ứng được các nhu cầu và lợi ích của cả nhân loại và hành tinh chúng ta. |

According to El-Kassas et al. [9], there are multiple ways to classify text summarization systems based on its characteristic. In this thesis, I only mention four ways of classification as shown in Figure 1.1.

**Based on input size:** Single-document or multiple-document. Single-document

Table 1.2: Example of extractive and abstractive multi-document summary

| | |
|---|---|
| **Document 1** | Trí tuệ nhân tạo là lĩnh vực khoa học kỹ thuật đang được phát triển mạnh mẽ trong thời gian gần đây. Từ việc tạo ra các hệ thống thông minh giống như con người, cho đến việc phát triển các máy học có khả năng học tập và cải thiện chức năng của chúng theo thời gian. Trí tuệ nhân tạo đang được áp dụng rộng rãi trong nhiều lĩnh vực, bao gồm cả kinh doanh, y tế và công nghệ thông tin. |
| **Document 2** | Trí tuệ nhân tạo đang trở thành một phần quan trọng trong cuộc sống hiện đại, từ các công nghệ thông minh đến các ứng dụng trong lĩnh vực y tế và kinh doanh. Tuy nhiên, nhiều người vẫn còn lo lắng về tương lai của con người với sự phát triển của trí tuệ nhân tạo. Liệu chúng ta có thể kiểm soát được trí tuệ nhân tạo và tránh những tác động tiêu cực của nó? Đây là câu hỏi cần được đặt ra và đưa ra giải pháp trong tương lai. |
| **Extractive summary** | Trí tuệ nhân tạo là lĩnh vực khoa học kỹ thuật đang được phát triển mạnh mẽ trong thời gian gần đây. Tuy nhiên, nhiều người vẫn còn lo lắng về tương lai của con người với sự phát triển của trí tuệ nhân tạo. Đây là câu hỏi cần được đặt ra và đưa ra giải pháp trong tương lai. |
| **Abstractive summary** | Trí tuệ nhân tạo là một lĩnh vực khoa học kỹ thuật đang phát triển rất nhanh chóng và được sử dụng rộng rãi trong nhiều lĩnh vực khác nhau, bao gồm cả kinh doanh, y tế và công nghệ thông tin. Mặc dù trí tuệ nhân tạo có thể mang lại nhiều lợi ích cho con người, nhưng cũng có nhiều người lo ngại về tác động tiêu cực của nó đến cuộc sống và xã hội trong tương lai. Ta cần xem sét khả năng kiểm soát và tránh những tác động tiêu cực của trí tuệ nhân tạo, cũng như cần phải xem xét đến đạo đức và trách nhiệm trong việc phát triển và sử dụng trí tuệ nhân tạo. |

summarization system processes input from only one source text, aiming to shorten the paragraph while remaining important data. Summary in Multi-document summarization is generated by extracting information from a set of data and removing duplicate contents. One of the main differences between the two approaches is that multi-document summarization requires the removal of duplicate content across the input documents, whereas single-document summarization does not. Multi-document summa-

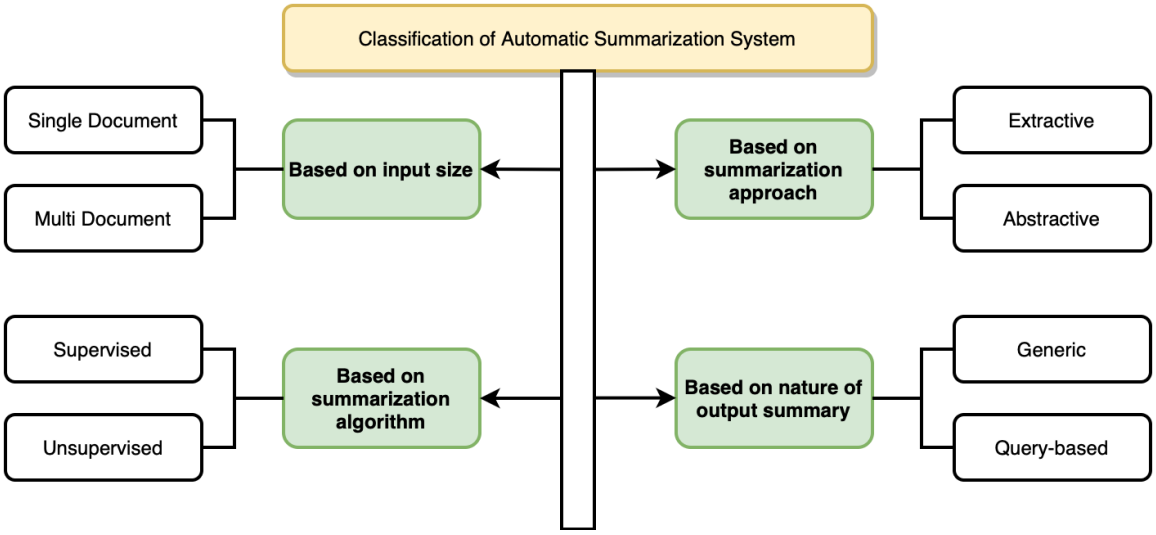


Figure 1.1: Classification of Automatic Text Summarization system

rization techniques need to ensure that redundant information is not included in the final summary to avoid repetition and maintain the coherence of the summary. Additionally, multi-document summarization requires more sophisticated algorithms to identify and extract relevant information from a set of documents.

**Based on summarization algorithm:** Unsupervised approaches do not require a training phase or labeled training data, while supervised approaches rely heavily on human efforts to label and annotate training data. As a result, supervised methods are often more expensive and time-consuming than unsupervised methods.

In unsupervised learning, the algorithm analyzes the input data and identifies patterns and structures on its own, without any guidance or supervision. This type of approach is useful in situations where there is no prior knowledge about the data or the patterns that may exist within it. On the other hand, supervised learning requires a significant amount of labeled data, which is used to train the algorithm to make predictions or classifications on new, unseen data. This process involves human efforts to label and annotate the training data, which can be time-consuming and expensive. However, once the algorithm has been trained, it can make high-quality summaries.

**Based on summarization approaches:** *Extractive summarization* involves selecting a subset of the most important sentences or phrases from the original document and presenting them as a summary. The selected sentences are usually taken verbatim from the original text, and no new text is generated in the summarization process. Extractive summarization is often used in situations where the goal is to provide a condensed version of the original text that still conveys the most important information, such as

news articles or scientific papers. The process of extractive summarization involves several steps, such as identifying important sentences, calculating the relevance of each sentence, and selecting the most relevant sentences to include in the summary. Several methods and algorithms are used to accomplish these steps, such as frequency-based, graph-based, and machine learning-based methods.

Despite its usefulness, extractive summarization has some limitations. One of the main drawbacks is that it only selects sentences from the original text, which means that it cannot generate new information or insights. Additionally, extractive summarization can sometimes produce a summary that is disjointed or lacks coherence, especially if the original text contains complex or technical language. Nonetheless, extractive summarization remains a popular and effective technique for generating summaries in many applications. Building models to form extractive summaries is usually simpler and costs fewer resources than abstractive models, but extractive systems' outputs are not always seamless, coherent, and concise. Extracted sentences may contain many salient points of the original paragraphs but these sentences can be very lengthy and have superfluous expressions. Consecutive sentences in extractive summaries can be very not relevant to each other and it causes the paragraph hard to follow for readers. Because of those defects, abstractive approaches are considered to create more human-readable summaries.

*Abstractive summarization* is a technique that goes beyond simply extracting sentences from the input document and instead aims to generate a summary that conveys the most important information concisely and coherently. Unlike extractive summarization, which only selects sentences from the original text, abstractive summarization involves generating new sentences that are not present in the original text. The goal of abstractive summarization is to create a summary that is more readable and natural-sounding than an extractive summary, while still preserving the meaning and information conveyed by the original text.

To generate abstractive summaries, summarization systems must use natural language generation techniques to create new sentences that convey the key information from the input document. This involves understanding the meaning of the input document and identifying the most salient information to include in the summary. Abstractive summarization is particularly useful when the original text contains information that cannot be easily summarized by simply selecting sentences, or when a more natural and readable summary is desired.

While abstractive summarization offers many benefits, it also presents significant

challenges. For example, generating new sentences that accurately capture the meaning and nuance of the original text can be difficult. Additionally, ensuring that the summary is both concise and coherent requires careful consideration of sentence structure and language use. Despite these challenges, researchers continue to explore ways to improve the accuracy and efficiency of abstractive summarization techniques.

**Based on nature of output summary:** Generic or query-based. Generic summaries are a type of summarization that provides an overview of the entire input document. These summaries cover the main points and do not focus on a specific topic. They are useful when a reader wants to get a general understanding of the content without having to read the entire document.

On the other hand, query-based summarization is a type of summarization that aims to generate summaries that are focused on answering users' specific questions. This type of summarization only includes the most relevant information that is needed to answer the user's query. This approach is particularly useful when a user is looking for specific information and does not want to sift through irrelevant content.

**Thesis scope** In this thesis, studies are made on Vietnamese multi-document news, and news data vary in many categories (education, sport, research, accident, calamity, etc). The implemented model contains a pre-processing module, an extractive module, and an abstractive module. The pre-processing module takes raw data and constructs it into input for the extractive module. The extractive model brings a low-cost, quick summarization time system and produces summaries that include important information. The abstractive module applies a pre-trained transformer-based model, generating a concise summary. By combining these two modules, the final model produces concise, coherent, and informative summaries.

## 1.3 Difficulties and challenges

In general, text summarization is a difficult problem, and multiple aspects are taken into consideration to solve the problem.

- **Evaluation** Determining the quality of a summary, even with manual evaluation, can be a challenging task. While the ROUGE score is a commonly used metric in summarization tasks, its reliability is not fully guaranteed due to its inability to capture the paragraph topic. Similarly, evaluation methods for computer-generated summaries that do not involve comparisons with human-produced summaries can

also be unreliable. Therefore, accurately evaluating the quality of a summary remains a complex and ongoing challenge in the field of natural language processing.

- **Inconsistency** There exist numerous approaches to writing a summary manually, as even experts may have their unique writing styles. For instance, academic abstracts are typically lengthy and provide comprehensive details, whereas executive summaries are typically brief and provide only the most essential information. It is also worth noting that articles and news stories covering a particular event can differ widely depending on the source, with duplicated content often present due to the use of varying writing styles.

  Identifying duplicate content can be challenging due to these differences in writing styles. This highlights the importance of developing effective techniques for automatically generating summaries that can capture the essence of the original text while avoiding duplicated content.

- **Lack of data** The majority of labeled Vietnamese textual datasets available for research purposes are derived from articles and news sources. While these datasets can be useful for developing natural language processing models, they may present a challenge in terms of evaluating the consistency of methods across different domains. Since these models are trained on a specific type of text, applying them to datasets from different domains may yield results that cannot be directly compared. This can make it difficult to assess the effectiveness of natural language processing methods in a broader context.

- **Appearance of non-textual data** In some news and article posts, images and videos are included alongside the text. While these materials may be directly referenced in the article, including sentences about them in the final summary can result in a summary that is not logically coherent or easily understandable. Furthermore, incorporating salient points from images and videos into the final abstractive summary requires additional research to determine the most effective representation of these materials. As such, further investigation is necessary to develop techniques for integrating these multimedia elements into the summarization process.

- **Time order problem** The challenge of summarizing multiple documents can have a significant impact on the accuracy of the resulting summary. While it may be relatively straightforward to arrange extracted sentences in the correct order, the sheer volume of documents within a cluster can make the summarization task quite

challenging. This can lead to a lack of coherence in the summary, as it can be difficult to create a logically understandable paragraph from the extracted information. One potential approach to addressing this challenge is to extract any timelines that may be present in the sentences, but even this can be problematic. Overall, summarizing multiple documents presents a complex and multi-faceted problem that requires careful consideration and strategic approaches to achieve accurate and coherent summaries.

- **Vietnamese natural language processing** There is still a lack of studies on abstractive text summarization for Vietnamese textual data. Some other works are language dependent so it is hard to tune the model to fit Vietnamese data. Finding a method to rewrite the whole sentence that ensures both grammatical and semantic can also be impossible. Also, the Vietnamese language is a tonal language with complex syntax and morphology, which makes it difficult for machines to understand and process. Ambiguity in the language can lead to difficulties in determining the main idea of a text, which can affect the quality of the summary.

- **Abstractive sequence-to-sequence language models** The application of deep learning, transformer-based language models can be a challenging and resource-intensive endeavor. The inherent complexity of these models can pose difficulties in fine-tuning the hyper-parameters, which can greatly impact their performance. Additionally, the extensive training required for these models can take many hours or even days to complete. Furthermore, due to the inherent limitations on input length, their ability to effectively summarize large documents may be constrained. These factors collectively make the use of transformer-based language models a complicated and costly undertaking.

## 1.4 Contribution of the thesis

**Previous work:** Our team achieved third place on the public test set of the VLSP 2022 shared task and fourth place on the private test set of the VLSP 2022 shared task. The previous work was only able to generate extractive summaries.

**Contribution of the thesis** This thesis proposed a combined model to produce extractive and abstractive summaries for Vietnamese textual data. Some main contributions are:

- Apply combined extractive methods [3] to produce extractive models for Vietnamese summarization. This approach focuses on applying ways of ranking sentences and then combining their scores to produce the final scores for each sentence.

- Using outputs of the fine-tuned extractive model as input for the abstractive pre-trained model to produce abstractive summaries. The extractive module is lightweight with fast execution time and serves as an improvement for the final abstractive summary. The output is short, concise, and coherent with correct grammar and semantically.

**Structure of the thesis:** The thesis has four remaining sections, organized as follows:

**Chapter 2 Related work:** This chapter introduces and covers some related studies being shared and researched in this report. This chapter introduces some graph-based, statistical-based, and machine learning-based extractive methods, and some modern abstractive methods and models.

**Chapter 3 Proposed model:** The proposed model will be discussed in detail in this chapter. The first main section of this chapter digs into the components of extractive models and how to calculate the final sentence scoring. The other section focus on abstractive methods and pre-trained models. The pre-processing phases are also described in detail in this chapter.

**Chapter 4 Experiments and results:** This chapter reports the implementation, settings, and achieved results on extractive and abstractive models. This chapter also compares the performance between implemented methods and provides reports about the errors analysis phase to show mistakes in the model.

**Conclusion** The final chapter summarizes contributions and results and briefly discusses future work.

# Chapter 2

# Related work

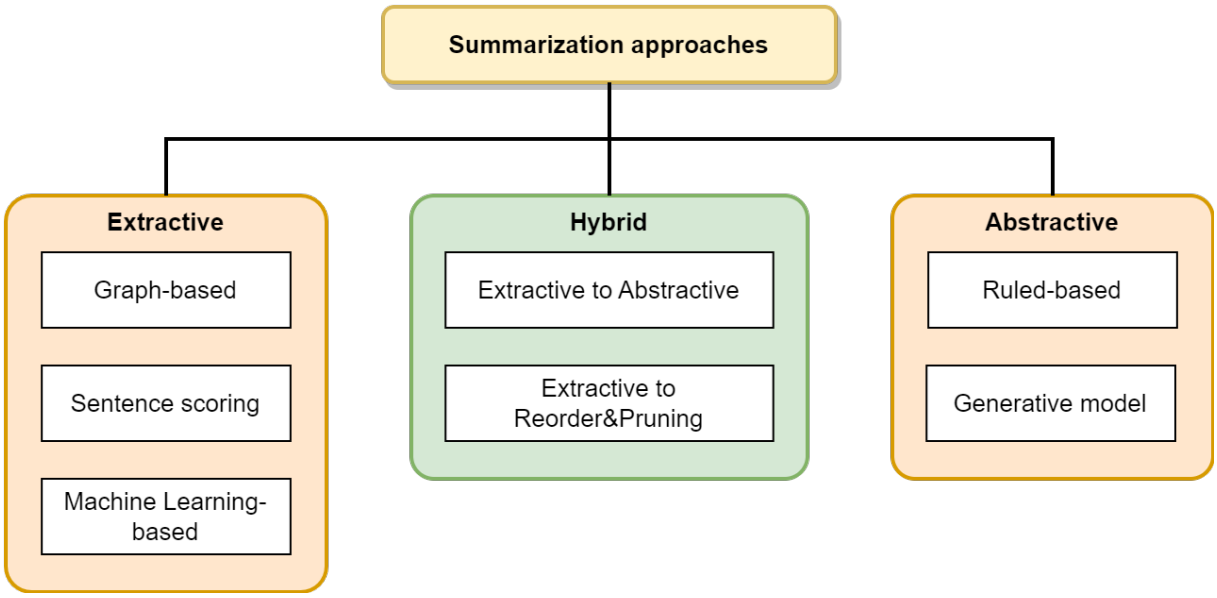


Figure 2.1: Automatic Text Summarization approaches

## 2.1 Extractive summarization approaches

Various techniques and methods have been developed to perform extractive summarization, each with its strengths and weaknesses. This section will explore some of the most common extractive summarization techniques and methods.

### 2.1.1 Graph-based

These methods score sentences by representing a document using graph data structures. PageRank [1] algorithm uses a directed graph to represent data and score each node. De-

rived from that idea, Lexrank [10] and Textrank [13] represent the document sentences using an undirected graph where each sentence is a node and semantic similarity scores between sentence embedding vectors are edges. Each sentence then being scored by a ranking algorithm to denote its importance.

The edges can be constructed in a variety of ways, such as by using word co-occurrence, sentence similarity, or semantic relationships between words. Once the graph is constructed, a variety of algorithms can be applied to identify the most important nodes. These algorithms usually involve calculating the importance or centrality of each node in the graph, based on factors such as its degree of connectivity, its position in the graph, or its importance to the overall meaning of the text. Figure 2.2 illustrates the general architecture of Lexrank and Textrank summarization techniques.

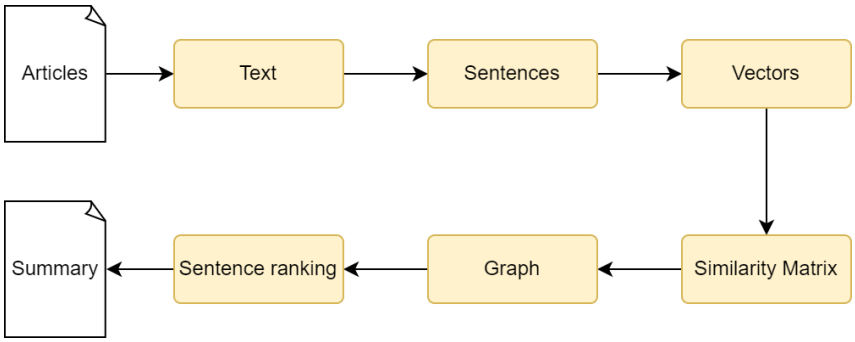


Figure 2.2: Lexrank and Textrank summarization architecture

This approach detects redundant information but does not consider the weights of words. Recent work on graph-based extractive summarization has focused on improving the accuracy and efficiency of the summarization process. For example, SGSum model [5] proposes a sub-graph selection approach for multi-document summarization and uses a machine learning approach based on graph neural networks (GNNs) to select the most informative sub-graphs for summarization. SGSum model resolves the drawback of Lexrank and Textrank methods as these two techniques can not check if two sentences in two different documents are similar in terms of content and information they convey.

### 2.1.2 Sentence scoring

In this approach, a set of statistical methods is employed to identify and extract the most relevant information from the document. These methods include the use of various attributes such as Term Frequency-Inverse Document Frequency (TF-IDF), Part-of-speech tagging, length of documents, number of words, entities, and many more.

One example of a sentence-scoring method for extractive summarization is the TF-IDF (Term Frequency-Inverse Document Frequency) approach. This method assigns a score to each sentence in the document based on the frequency of its words and their relevance to the document as a whole. Sentences that contain words that are rare or unique to the document are assigned a higher score, indicating that they are more important for summarization.

MEAD [18] is a software platform for extractive summarization that uses a feature-based approach. The method involves extracting features from the input text, such as sentence length and position, and using a clustering algorithm to group similar sentences. The most representative sentences from each cluster are then selected to form the summary. MEAD has been evaluated on various datasets and has been shown to achieve promising results.

The steps of this approach are usually feature extraction, which investigates statistical characteristics, assigns their weight, and finally finds an equation applying those weights to calculate scores of sentences. The disadvantages of this technique are some sentences are usually skipped because similar sentences have higher scores.

Sentence-scoring methods are effective in generating accurate summaries, particularly when applied to documents with well-defined structures, such as news articles or scientific papers. However, they can struggle with more complex texts, such as literary works or informal language, where context and nuance play a greater role in determining the importance of a sentence.

Similar to graph-based approaches, these techniques do not have the mechanism to classify sentences containing similar information but express it in different ways. To resolve this issue, Maximal Marginal Relevance [4] is a technique used in information retrieval and extractive summarization to select the most informative and diverse set of documents or sentences from a larger set of candidates. The idea behind MMR is to balance the relevance and diversity of the selected documents or sentences. Specifically, MMR seeks to maximize the relevance of each selected document or sentence to the query or topic of interest, while also ensuring that the selected documents or sentences are not too similar to each other.

To achieve this balance, MMR uses a scoring function that takes into account both the similarity of each candidate document or sentence to the query or topic, as well as the similarity of each candidate document or sentence to the already-selected documents or sentences. The scoring function is then used to rank the candidates, and the top-scoring

candidates are selected.

MMR has been shown to be effective in a variety of applications, including document retrieval, sentence extraction, and multi-document summarization. By balancing relevance and diversity, MMR can help to ensure that the selected documents or sentences provide a comprehensive and informative summary of the original text or dataset.

### 2.1.3 Machine Learning-based

Machine learning-based extractive summarization is a technique that aims to teach the model to identify the most important sentences in the input document that should be included in the final summary. This is accomplished by using a feature extraction phase, where a feature vector is calculated for each sentence. The feature vector captures various properties of the sentence, such as its length, position, and the presence of certain keywords.

Once the feature extraction phase is completed, a machine-learning model is trained on the extracted features to classify each sentence as either important or not important for inclusion in the final summary. The model is typically trained on a large corpus of text documents, which allows it to learn patterns and relationships between the features and the target variable (i.e., whether a sentence should be included in the summary or not).

The model SummaRuNNer [14] applies a recurrent neural network-based model to extractive summarization. The method involves encoding the input document as a sequence of sentence embeddings and using a sequence-to-sequence model to predict the importance of each sentence. The most important sentences are then selected to form the summary.

One of the advantages of the machine learning-based extractive approach is that it is relatively easy to train and implement. Additionally, it can be more transparent than other summarization techniques since the model is based on identifiable features and rules. However, there are also some limitations to this approach. For instance, it may not capture the overall meaning and context of the document as effectively as other techniques such as abstractive summarization. Moreover, it may struggle with identifying the most relevant information from the document if the features used for classification are not well-suited to the specific task at hand.

Collins et al. [7] proposes a supervised approach to extractive summarization us-

ing a Support Vector Machine (SVM) classifier. The method involves extracting features from the input document, such as sentence length, position, and similarity to the document title, and training the SVM classifier to predict the importance of each sentence. The most important sentences are then selected to form the summary. This approach was evaluated on a dataset of scientific articles and was shown to achieve competitive results.

### 2.1.4 Drawbacks of extractive summarization approach

While extractive-based summarization methods have shown promise in summarization tasks, they also have some drawbacks.

- **Inability to generate concise summaries:** Extractive summarization methods may struggle to generate concise summaries, especially in cases where the source text is long and contains many relevant sentences.
- **Over-Reliance on Sentence Features:** Extractive summarization methods rely heavily on features such as sentence length, position, and similarity to the document title. However, these features may not always be accurate indicators of the importance of a sentence and can lead to sub-optimal summarization.
- **Issues with handling duplicate information:** Extractive summarization can sometimes include multiple sentences that convey the same information, leading to redundancy in the summary.
- **Limited Cohesion and Coherence:** Extractive summarization methods select individual sentences without considering the overall cohesion and coherence of the summary. As a result, the generated summaries may lack coherence, which can make them difficult to read and understand.

## 2.2 Abstractive summarization approaches

Abstractive summarization requires the ability to make new sentences based on original paragraphs. The architecture of an abstractive summarization system usually consists of 4 modules: pre-processing, text representation, summary generation, and post-processing. Some advanced representation and sentence-generation techniques have appeared, applying very complicated deep-learning methods but to produce better output, it requires a lot of resources and processing time. Hence, methods that consume less memory and processing time like ruled-based are also taken into consideration.

### 2.2.1 Ruled-based approach

Rule-based abstractive summarization is a type of text summarization technique that aims to extract and rephrase the most relevant information from a document using a set of predefined rules. This approach requires a human to create and refine the rules used in the summarization process manually.

For instance, Cohan and Goharian [6] proposed a model that uses a set of domain-specific rules to summarize scientific papers. Recently, there have been efforts to automate the rule-creation process using machine learning and natural language processing techniques [11]. However, rule-based abstractive summarization still faces challenges in accurately capturing the meaning and nuances of the source text.

Furthermore, the creation and refinement of these rules can be time-consuming and resource-intensive. As a result, many researchers are now exploring the use of neural network-based approaches to improve the accuracy and efficiency of abstractive summarization. These approaches aim to overcome the limitations of rule-based methods by learning to generate summaries from large amounts of data without explicitly defining rules. The hope is that these methods will be more effective in capturing the meaning and nuances of the source text while requiring less human effort.

### 2.2.2 Generative model

Deep learning abstractive summarization is a powerful technique that can generate concise and accurate summaries of long-form documents. Different from mentioned approaches, this approach aims to learn the relationships between words and generate summaries that capture the most important information in the original text. One notable approach is the use of pre-trained language models such as GPT-3 [2] and T5 [20]. These models have shown impressive results in various natural language processing tasks, including abstractive summarization. ViT5 [17] is a state-of-the-art Vietnamese abstractive summarization model that is based on the Transformer architecture. It is designed to generate concise and coherent summaries from long Vietnamese texts. Both models use a combination of self-attention and cross-attention mechanisms to capture important information from the input text and generate a summary that captures the main ideas and key details of the text. For more details:

- **BERT** (Bidirectional Encoder Representations from Transformers) [8] is a transformer-based model that is trained on a large amount of text data using a self-supervised

learning approach. BERT is designed to understand the context and meaning of words in a sentence by considering the words that come before and after them. This allows BERT to generate highly accurate representations of words and sentences that can be used for a variety of NLP tasks, including summarization. The Transformer model is a type of neural network that uses self-attention mechanisms to process sequences of input data, such as text. The GPT model extends the Transformer architecture by training the model on a large corpus of text data using a language modeling objective. This training allows the model to learn the patterns and structures of language in a general way, making it a powerful tool for a wide range of natural language processing tasks, including abstractive summarization.

- **PhoBERT** [15] is a BERT-based model that is specifically designed for the Vietnamese language. PhoBERT is pre-trained on a large corpus of Vietnamese text data. While PhoBERT was primarily designed for extractive summarization, it has also been used for abstractive summarization with promising results.

- **GPT** Generative Pre-trained Transformer [2], is a language model developed by OpenAI. It is a type of neural network that is pre-trained on a large corpus of text data and can generate high-quality text by predicting the next word in a sequence of words. In the context of abstractive summarization, the GPT model takes a text document as input and generates a summary of the document that captures the main points and ideas. The model achieves this by generating new text that is a concise and coherent representation of the original text. The model uses a combination of techniques, such as language modeling, attention mechanisms, and sequence generation, to generate the summary.

- **T5:** (Text-to-Text Transfer Transformer) [20] is also based on the transformer architecture and uses a fine-tuned variant of the transformer architecture to generate summaries of input text. The advantage of this model is the ability to process long input efficiently. T5 is also trained on a large corpus of text data using a self-supervised learning approach that involves predicting the next sequence of text given the previous sequence. This approach allows T5 to learn the structure and coherence of natural language text, as well as the ability to generate new text that captures the essential meaning of the original text.

- **ViT5** [17] is an encoder-decoder model that is pre-trained on the vietnews [16] dataset using the Transformer architecture. The model is trained on a vast and diverse corpus of high-quality Vietnamese texts using a self-supervised pre-training

approach similar to T5. The encoder-decoder architecture of ViT5 is composed of multiple transformer layers. The encoder processes the input Vietnamese text, and the decoder generates a summary by decoding the encoded text.

- **PEGASUS** [25] use gap sentence strategy and transformer architecture to generate abstractive summaries. The gap sentence generation strategy involves masking a sentence in the input document and training the language model to predict the masked sentence based on the context of the surrounding sentences. The masked sentence is used as the target output for the model, and the context of the surrounding sentences is used as the input. The model is then trained to predict the missing sentence based on the context of the surrounding sentences. By repeatedly masking different sentences in the input document and training the model to predict the missing sentence, the model can learn to generate summaries that capture the essential meaning of the input document. This approach allows the model to learn the structure and coherence of natural language text, as well as the ability to generate multiple candidate summaries and select the best one. The gap sentence generation strategy is a form of self-supervised learning, as the model is trained on unlabeled text data without the need for human annotations. This allows the model to be trained on large amounts of text data and to learn general patterns and structures in natural language text that can be applied to a wide range of downstream tasks, including abstractive summarization.

## 2.3 Hybrid summarization approaches

Extractive methods involve selecting and combining the most relevant sentences or phrases from the source text to form a summary. Abstractive methods, on the other hand, involve generating a summary by paraphrasing and rephrasing the source text more concisely and coherently. Hybrid approaches (extractive to abstractive) can combine extractive and abstractive methods to take advantage of each approach. Figure 2.3 shows the general architecture of a hybrid summarization system, focusing on the processing phase with the extractive module and the sentence generation module.

In 2020, Tretyak and Stepanov [23] proposed a hybrid method that combines extractive and abstractive approaches for long text summarization. The method uses a two-stage approach, where the first stage extracts important sentences from the input text, and the second stage generates an abstractive summary using a transformer-based model. The method was evaluated on the WikiHow dataset and achieved state-of-the-art

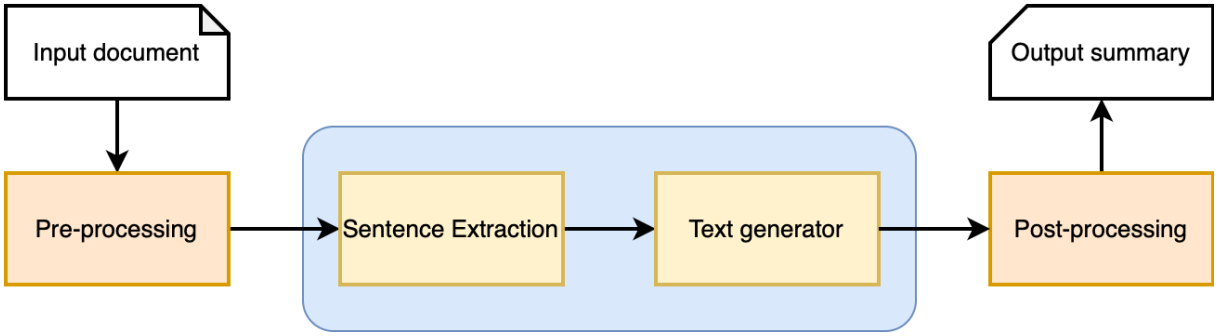


Figure 2.3: General architecture of the automatic hybrid (extractive to abstractive) summarization system

results. Shinde et al. [21] also combined an extractive module using k mean clustering with a pre-trained abstractive model to form a concise and coherent summary.

By combining these two methods, the hybrid approach can leverage the strengths of each method while mitigating their weaknesses. For example, extractive methods are good at preserving the factual accuracy of the source text, while abstractive methods are better at generating summaries that are more concise and readable. Extractive summarization methods can be limited in their ability to cover all the important information from the source text, especially in cases where the relevant information is spread across multiple sentences or paragraphs. Hybrid approaches can overcome this limitation by using abstractive methods to generate new sentences that fill in the gaps.

Other hybrid approaches may combine different types of models or techniques, such as rule-based and machine learning-based methods, or neural and non-neural methods. The goal of these hybrid approaches is to produce summaries that are more accurate, informative, and readable than those generated by individual methods.

# Chapter 3

# Proposed model

In this chapter, a detailed explanation is provided of the various components and methods used in the proposed Vietnamese multi-document summarization model. The architecture of the extractive to abstractive summarization model is depicted in Figure 3.1 and comprises three key components: pre-processing, extractive module, and abstractive module.

The first step in the pre-processing component involves parsing the raw input through a loader and then tokenizing and embedding it. The output of this step is then used as the input for the sentence-scoring algorithms. The extractive module is composed of multiple sentence-scoring sub-modules and a scoring function that is responsible for determining the extractive document.

Finally, in the abstractive module, a new summary is generated based on the selected sentences. In this last stage, the selected sentences can be segmented by indexing that map with the input multi-document before being fed into the fine-tuned Vit5 abstractive model.

## 3.1 Pre-processing

In the pre-processing phases, the input is a set of raw documents $D = \{D_1, D_2, D_3, ...D_n\}$. Each document $D$ contains the main information and may include additional data such as title, date created, or author. The steps in this phase are loader, tokenize, and sentence embedding. The output of this module is processed data saved as attributes of instances representing each document.

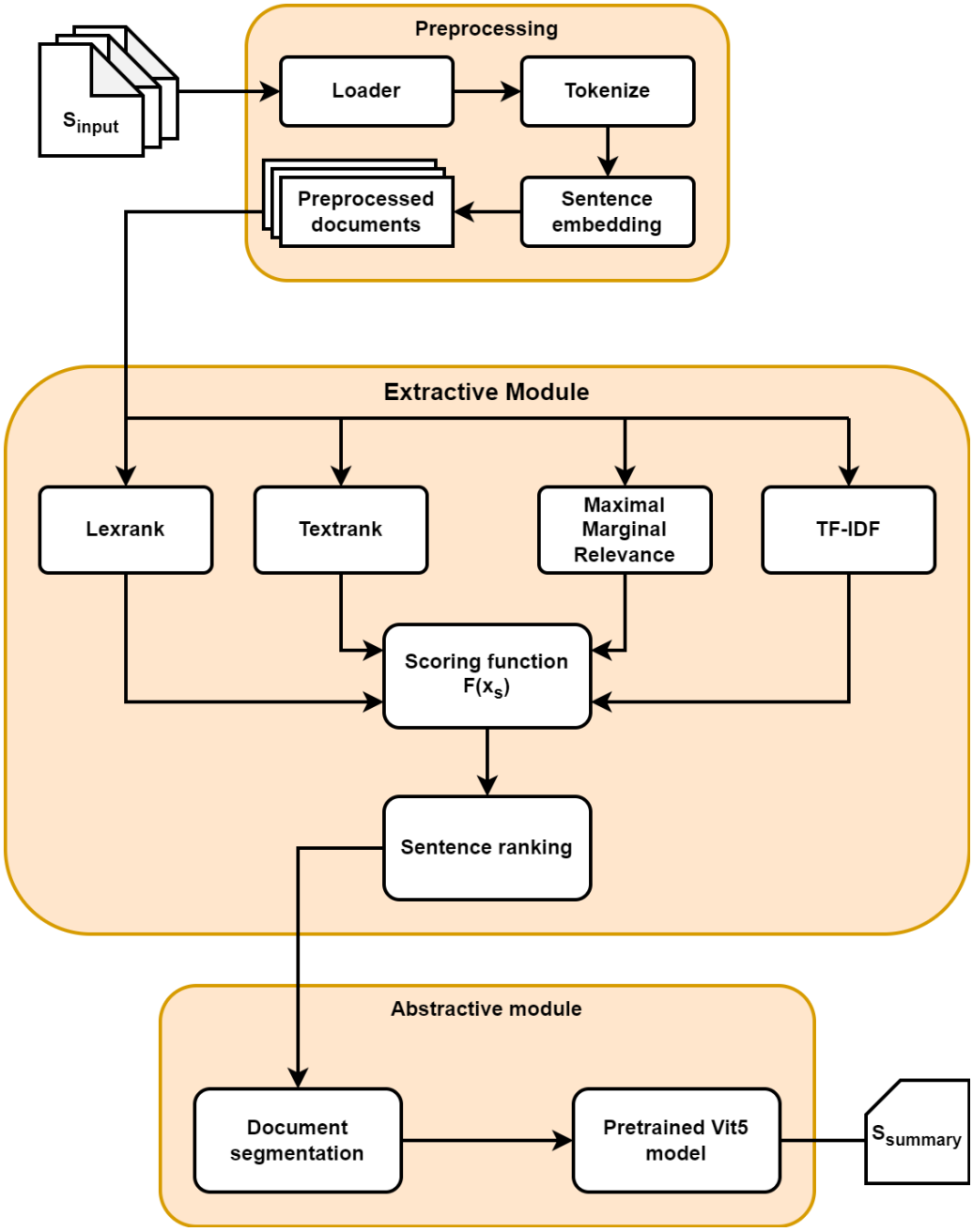


Figure 3.1: Architecture of proposed hybrid summarization model

### 3.1.1 Loader

The first step in pre-processing is to load the text data into memory. This step simply maps the attributes of the input data into objects of the class that represents the input document $D$, serving as input for tokenizing and embedding. Figure 3.2 visualizes an example of a document instance. The main attribute of each document is the content; depending on the dataset, other attributes might appear but currently, the proposed model has yet to take advantage of these properties. This additional information is normalized, it is either being skipped or treated as part of the main content.

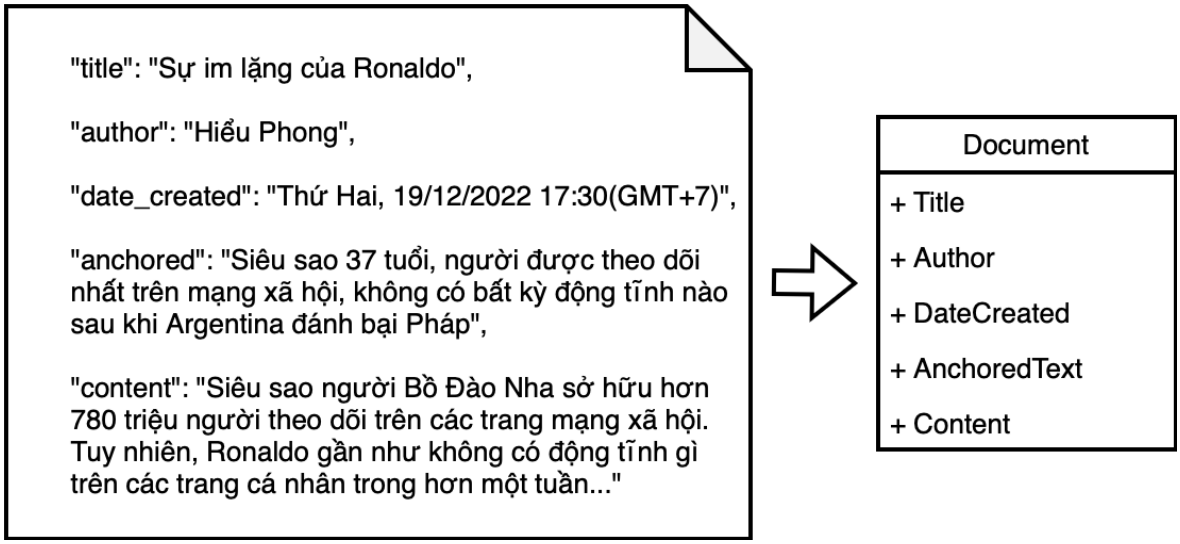


Figure 3.2: An example of loading raw data into a Document object

### 3.1.2 Tokenization

In the proposed approach, document tokenization is employed as the initial step in the extractive summarization process. This involves breaking down the text document into individual units known as tokens, where in this particular work, words and punctuation are treated as tokens.

To carry out the tokenization process, the Underthesea[1] package is utilized, which is a Vietnamese NLP toolkit that provides reliable and accurate segmentation of sentences, words, and punctuation in the input text. This library has been extensively verified and proven to produce high-quality tokenized output for Vietnamese text data.

The output of the sentence segmentation step serves as the basis for sentence indexing in the extractive summarization module, where the most relevant and informative sentences are selected to be included in the summary. The word tokenization module, which is performed simultaneously with sentence segmentation, further breaks down the text into individual words, allowing for a more fine-grained analysis of the document's content.

In addition to sentence and word tokenization, the proposed approach also includes a stop-word filtering module to experiment with its impact on the summarization results. Stop words are common words that have little semantic meaning, and removing them can potentially improve the sentence-scoring phases.

[1] Underthesea - Vietnamese NLP Toolkit: https://underthesea.readthedocs.io/

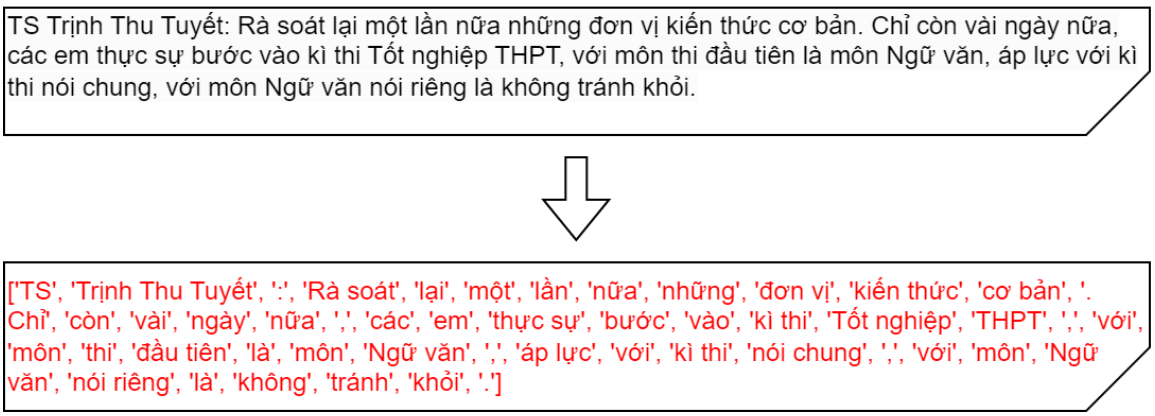


Figure 3.3: An example of document tokenization

### 3.1.3 Sentence embedding

In the proposed model, the tokenized documents are initially processed by being passed directly to the sentence embedding submodule. This submodule is responsible for generating vector representations for each sentence, which are subsequently used in the extractive summarization process.

Sentence embedding is a technique that converts textual data, such as sentences or phrases, into fixed-length vector representations. The objective is to capture the underlying meaning and semantic relationships between the words in a sentence, in a manner that can be easily processed by machine learning algorithms. The resulting vector representation can then be used in a variety of natural language processing tasks, such as classification, clustering, or summarization.

To perform sentence embedding in the proposed model, the authors employed the BERT model, which is a pre-trained model that is effective in a variety of natural language processing tasks. The BERT model generates 768-dimensional vector representations for sentences, which are used to calculate sentence similarity in the graph-based sentence extraction phases of the extractive summarization process.

Overall, the use of BERT-based sentence embedding in the proposed model enables the efficient calculation of sentence similarity, which is a critical component of the extractive summarization process. The resulting vector representations capture the semantic relationships between sentences, enabling the model to identify the most relevant and important sentences for inclusion in the summary.

After the three pre-processing steps, generated properties are saved and ready for

the extractive module.

## 3.2 Extractive module

This section provides a detailed description of the extractive components. The baseline extractive module is experimented with before the final extractive module is studied. The baseline model is constructed from a pre-processing module, 3 scoring modules (lexrank, textrank, TF-IDF), a scoring function, and an MMR filter to produce an extractive summary.

The proposed model contains four different ranking strategies, a scoring function, and a sentence extraction module. The four ranking methods being studied in this thesis are *lexrank*, *textrank*, *maximal marginal relevance*, and *TF-IDF scoring*. This extractive module acts as a sentence filterer, filtering less important sentences from the original documents.

### 3.2.1 Lexrank scoring

LexRank is an unsupervised algorithm that applies techniques from natural language processing to identify the most important sentences in a document or a set of related documents. The algorithm is based on the idea of using graph-based methods to identify sentences that are most similar to other sentences in the document, and that, therefore, represent the most important content.

The algorithm is composed of two main steps: first, it creates a graph where each node represents a sentence in the document, and the edges between nodes are determined by the similarity between the sentences. Second, it applies a ranking algorithm to assign a score to each sentence in the graph, based on its importance in the document.

- In the first step, the algorithm creates a graph where each node represents a sentence in the document. The algorithm then computes the cosine similarity between the vectors representing the sentences, which is calculated by applying TF-IDF vectorization, this process will be discussed further in the TF-IDF scoring step. The cosine similarity (3.1) between two vectors $\mathbf{A}$ and $\mathbf{B}$ is denoted as:

$$\text{simLexrank}(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A} \cdot \mathbf{B}}{|\mathbf{A}||\mathbf{B}|} \tag{3.1}$$

  The edges between nodes in the graph are determined by the similarity scores, such that each node is connected to its most similar nodes. To determine the number of

edges to include, the algorithm uses a threshold similarity score that is determined empirically, and which can be adjusted based on the desired level of summarization.

- In the second step, the algorithm applies a ranking algorithm to assign a score to each sentence in the graph, based on its importance in the document. The ranking algorithm used in LexRank is based on the PageRank[1] algorithm, which was originally developed by Google to rank web pages based on their importance.

  The PageRank algorithm assigns a score to each node in the graph based on the number and quality of the edges that connect to it. In the context of LexRank, the quality of the edges is determined by the similarity scores between the sentences. The algorithm iteratively updates the scores of each node based on the scores of its neighbors, until convergence is reached. The final output of the algorithm is a ranking of the sentences based on their scores, where the highest-ranked sentences represent the most important content in the document.

### 3.2.2 Textrank scoring

Overall, the textrank algorithm is very similar to lexrank, instead of calculating based on the TF-IDF vector between sentences like Lexrank, Textrank uses several common words normalized by the length of the sentence. The following equation 3.2 gives more detail about the sentence scoring step in textrank, $w$ is the token and $X$ and $Y$ are the two sentences.

$$\text{simTextrank}(X, Y) = \frac{|w \in X \cap w \in Y|}{\log(|X|) + \log(|Y|)} \tag{3.2}$$

Overall, LexRank and TextRank are unsupervised algorithms for automatic text summarization using graph-based approaches. They have several similarities, including:

- Graph-based approach: Both LexRank and TextRank represent sentences as nodes in a graph, and the edges between the nodes represent the similarity between the sentences.

- Ranking algorithm: Both algorithms use a ranking algorithm, based on the PageRank algorithm, to assign importance scores to each sentence in the graph. The sentences with the highest scores are then selected as the summary.

[1] https://en.wikipedia.org/wiki/PageRank

- Language-agnostic: Both algorithms can be applied to any language without modification.

Despite their similarities, both algorithms also have some drawbacks. One drawback of LexRank is that it may not handle very short or very long texts effectively. In very short texts, the algorithm may not have enough data to build a useful graph, while in very long texts, the algorithm may become computationally expensive. Another drawback of LexRank is that it does not take into account the importance of words or phrases within a sentence, which can lead to some important information being left out of the summary.

Similarly, TextRank also has some drawbacks. One of the main limitations is that it does not consider the coherence of the summary. This means that the selected sentences may not flow well together, which can make the summary difficult to read. Additionally, TextRank may not work well on very short or very long texts, and it may be affected by the quality of the similarity metric used.

In summary, while LexRank and TextRank share some similarities in their approaches to text summarization, they also have some differences and limitations that should be considered when choosing an algorithm for a particular task.

### 3.2.3 Maximal marginal relevance scoring

Maximal Marginal Relevance (MMR) [4] is a text summarization algorithm that is used to select the most relevant sentences or documents from a larger set of text. The main idea behind MMR is to balance the relevance and diversity of the selected text. MMR achieves this by selecting sentences that are both relevant to the query and different from each other. In other words, it tries to maximize the similarity of selected sentences to the query while minimizing the similarity between the selected sentences.

MMR works in two steps. The first step is to select the most relevant sentence or document using TF-IDF. The second step is to select the sentence or document that is most different from the ones already selected. This is achieved by computing a similarity score between the selected sentence and the remaining sentences or documents and then subtracting this score from the relevance score. The sentence or document with the highest combined score is selected next. Formally, the MMR score of a sentence $s_i$ is given by formula 3.3

$$\text{MMR}(s_i) = \lambda \cdot \text{sim}(s_i, q) - (1 - \lambda)\text{max}_{s_j \in S_{selected}}\text{sim}(s_i, s_j) \tag{3.3}$$

In this formula, $\lambda$ is a parameter that controls the trade-off between relevance and diversity, q is the query, $\text{sim}(s_i, q)$ is the similarity between sentence $s_i$ and the query, and $\max(\text{sim}(s_i, s_j))$ is the maximum similarity between sentence $s_i$ and any of the previously selected sentences $s_j$.

The value of $\lambda$ is usually set between 0.5 and 0.7 depending on the application. A higher value of $\lambda$ will result in more relevant but less diverse summaries, while a lower value of $\lambda$ will result in more diverse but less relevant summaries.

### 3.2.4 Term Frequency-Inverse Document Frequency scoring

TF-IDF (Term Frequency-Inverse Document Frequency) is a statistical measure that is commonly used in information retrieval and text mining. TF-IDF is a combination of two metrics: TF (Term Frequency) and IDF (Inverse Document Frequency). TF measures how frequently a term appears in a document, while IDF measures how rare the term is across all documents in the corpus. The product of these two metrics is the TF-IDF score of a term in a document. Choosing appropriate values for these parameters can be challenging, and the performance of the algorithm may be sensitive to their values.
The term frequency (TF) of a term in a document is calculated as follows:

$$TF(t, d) = \frac{\text{(number of times the term } t \text{ appears in document } d)}{\text{(total number of terms in document } d)} \tag{3.4}$$

The inverse document frequency (IDF) of a term is calculated as follows:

$$IDF(t) = \log\left(\frac{\text{total number of documents}}{\text{number of documents that contain the term } t}\right) \tag{3.5}$$

The TF-IDF score of a term in a document is then calculated as follows:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t) \tag{3.6}$$

In the proposed model, other than calculating TF-IDF for each word to form embedded vectors to represent sentences, TF-IDF also applies to score sentences by calculating the average value of a token in a sentence. To get the optimal results, top-k tokens with the highest score being chosen are taken into consideration.

### 3.2.5 Scoring function and sentence extraction

In this module, the four above scoring methods are normalized to the range between 0 and 1 by using min-max normalization. After being normalized, optimized weights are applied to combine calculated scores, denoted as a weighted sum fashion formula 3.7

$$\text{score}(s_i) = \sum w_t \times score_t \tag{3.7}$$

where $t$ is one of four scores: Lexrank, Texrank, MMR, and TF-IDF scoring.

**Ranking and selecting sentences:** after achieving the combined score, important sentences are selected from the original data. There are several ways to determine the number of selected sentences, including selecting the top-k sentences with the highest scores or selecting the top-p% highest score sentences. However, both approaches have their drawbacks.

The threshold-based approach involves selecting the top-k sentences with the highest scores. This can be effective in clusters with a large number of sentences, as it ensures that only the most relevant sentences are included in the summary. However, in clusters with a small number of sentences, this approach may result in redundancy, as the same sentences may be selected multiple times.

On the other hand, the percentage-based approach involves selecting the top-p% highest score sentences. This approach can be useful in clusters with a small number of sentences, as it ensures that a sufficient number of sentences are included in the summary. However, in clusters with a high number of sentences, this approach may result in a summary that is too long and includes less relevant sentences.

Based on empirical results, a hybrid approach has been developed to overcome these limitations. This approach involves choosing the number of selected sentences as the minimum of the percentage-based and threshold-based approach values. This ensures that a sufficient number of relevant sentences are included in the summary while avoiding redundancy or excessive length.

## 3.3 Abstractive module

This section describes how the proposed model applies the pre-trained model to generate an abstract summary. This module contains two main steps:

**Document segmentation:** this sub-module takes selected sentences as input and produces a list of documents. The output can contain only a single document that com-

bines from all selected sentences or N documents $\{D_i'\}$ corresponding to N original documents $\{D_i\}$

### 3.3.1 ViT5 pre-trained model

This subsection explains the applied pre-trained model. The deep architecture of the ViT5 abstractive Vietnamese summarization model is based on the Transformer model, which is a type of neural network architecture that was first introduced by Vaswani et al. [24] in 2017.

The ViT5 model consists of two main components: an encoder and a decoder. The encoder takes in the input text and generates a series of representations for each token in the text. These representations are then passed on to the decoder, which generates a summary of the input text.

The encoder in ViT5 consists of multiple layers of self-attention and feed-forward neural networks. Self-attention is a technique that allows the model to focus on different parts of the input text when generating the summary. In each self-attention layer, the model computes a weighted sum of the input representations, where the weights are based on the similarity between the representations. The resulting weighted sum is then passed through a feed-forward neural network to generate a new set of representations.

The decoder in ViT5 is also based on the Transformer architecture and consists of multiple layers of self-attention and feed-forward neural networks. In each self-attention layer, the model computes a weighted sum of the representations generated by the encoder, where the weights are based on the similarity between the representations. The resulting weighted sum is then passed through a feed-forward neural network to generate a new set of representations, which are used to generate the summary.

ViT5 also uses a technique called positional encoding, which helps the model learn the position of each token in the input text. This is important because the order of the words in the input text can significantly affect the meaning of the text.

One of the key features of ViT5 is its ability to handle long input texts. It achieves this by breaking the input text into smaller segments and processing each segment separately. In the proposed model, different configurations are practiced to achieve the desired output, further reports will be discussed in chapter 4.

**Transformer** The Transformer architecture is a type of deep neural network architecture that was introduced by Vaswani et al. [24] in 2017. It has since become one

of the most widely used and powerful architectures in natural language processing and related fields. Figure 3.4 visualizes the main components of the transformer model.

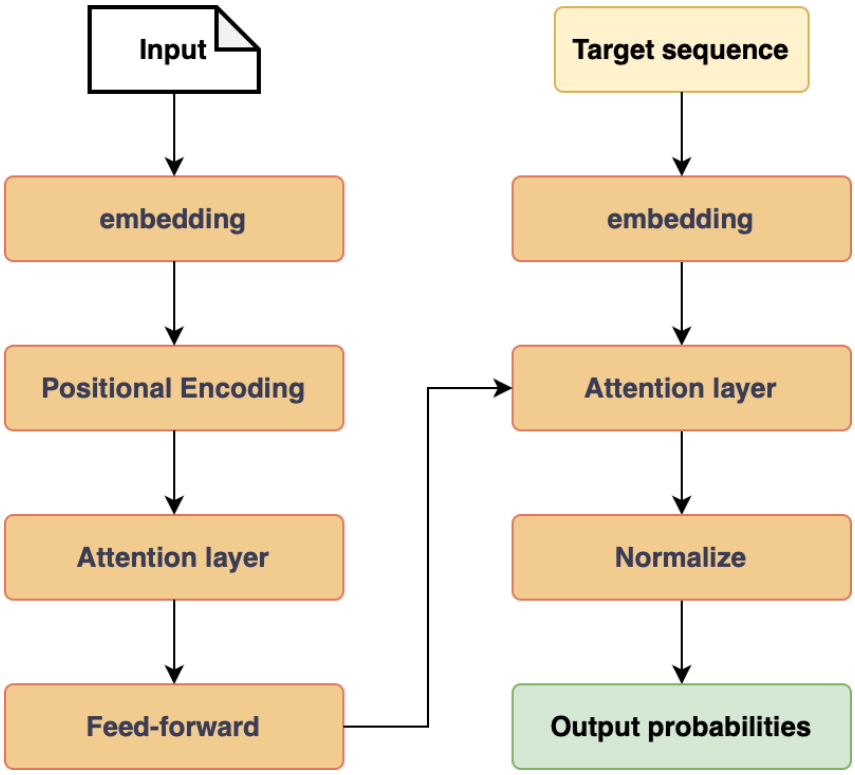


Figure 3.4: Main components of Transformer

The Transformer architecture is designed to process sequences of data, such as sentences or time-series data. It consists of two main components: an encoder and a decoder. The encoder takes a sequence of input data and produces a set of hidden representations, which the decoder uses to generate an output sequence. The encoder and decoder are connected by a set of attention mechanisms, which allow the decoder to selectively attend to different parts of the input sequence during the decoding process.

One of the key innovations of the Transformer architecture is the self-attention mechanism, which allows the network to attend to different parts of the input sequence with varying degrees of emphasis. This is achieved by computing attention scores between each pair of positions in the input sequence and using these scores to weight the contributions of each position to the final hidden representation.

The Transformer architecture also includes several other important features, including layer normalization, residual connections, and multi-head attention. Layer normalization helps to stabilize the training process by normalizing the inputs to each layer, while residual connections allow the network to propagate gradients more effectively through deep architectures. Multi-head attention allows the network to attend to multi-

ple “heads” of the input sequence simultaneously, which can improve the quality of the learned representations.

# Chapter 4

# Experiments and Results

This chapter provides reports about experiment phases and achieved results, divided into four sections. The first section, 4.1, describes the environmental settings and implementation of the model. Section 4.2 provides details about the datasets used and the evaluation metrics employed in the study. The following section, 4.3, presents the performance of the proposed model. Lastly, section 4.4 focuses on the analysis of errors encountered during the study.

## 4.1 Configuration and implementation

### 4.1.1 Package Dependencies and Environments Installation

The model was implemented using Python 3.9. The following packages are required during the phase:

- `underthesea` and `nltk` for pre-processing
- `pandas` and `numpy` for data loader and manipulation
- `lexrank` for lexrank interface
- `textrank` for textrank interface
- `sklearn` for TF-IDF vectorizer and TF-IDF scoring module
- `pytorch` for sentence transformer
- `tensorflow` and `transformer` for pre-trained ViT5 abstractive summarization model

- `rouge-score` for ROUGE sentence scoring module and metrics

**Environment setup**: My personal computers are used to implement and execute the model. At first, the model was implemented on an Intel chipset-based computer, running under a Linux-based operating system but due to slow execution time, a computer with an Apple M2 chipset is put into operation. The configurations are listed in table 4.1

Table 4.1: Installed configuration

| **Machine** | **Configuration** |
|---|---|
| **Personal Computer 1** | Intel(R) Core(TM) i5-10400 CPU @2.90GHz<br>2 x 8GB DDR4 RAM<br>Ubuntu LTS 20.04 |
| **Personal Computer 2** | Apple silicon M2 x 8 CPUs x 8 GPUs<br>1 x 16GB LPDDR5 RAM<br>MacOS Ventura version 13.2.1 |

### 4.1.2 Model settings

To optimize the performance of the proposed extractive summarization model, a thorough analysis of the hyper-parameters of each component is conducted. Table 4.2 provides a comprehensive overview of the fine-calibrated parameters used in the model, including the number of hyperparameters and their corresponding values. It is important to note that the specific hyper-parameters and values utilized may vary depending on the dataset being used and any additional techniques that have been implemented.

By carefully tuning these hyper-parameters, the model can achieve the best possible performance in terms of summarization quality and computational efficiency. The process of hyper-parameter tuning involves selecting optimal values using grid search.

## 4.2 Dataset and evaluation metrics

### 4.2.1 Dataset

The VLSP-AbMuSu dataset (Tran et al. [22]), contains not only a training dataset but also a validation and test dataset. The training and validation datasets both have a similar structure, consisting of multiple clusters.

Each cluster contains a variety of single documents, a golden summary, and a

Table 4.2: Configuration of model's components

| Component | | Component |
|---|---|---|
| **Preprocess** | Underthesea | `stop_word` = false |
| | SBERT embedding | `model` = 'sentence_transformer'<br>`distance` = 'cosine'<br>`max_df` = 0.5 |
| **Extractive module** | Lexrank | `w` = 0.35 |
| | Textrank | `w` = 0.35 |
| | MMR | `sigma` = 0.7<br>`w` = 0.1 |
| | TF-IDF | `w` = 0.2 |
| | Threshold | `thresh_hold` = 0.8<br>`top_k` = 8 |
| **Abstractive module** | Segmentation | `segmentation` = true |
| | ViT5 | `length_penalty` = 2.0<br>`no_repeat_ngram_size` = 3<br>`num_beams` = 4<br>`top_p` = 0.9 |

category tag. A single document within the VLSP-AbMuSu dataset is constructed using three components: a title, anchor text, and raw text. These components are combined to form a press article that provides contextual information for the summarization task. The golden summary included in each cluster is a multi-document summary that has been verified by experts, providing a high-quality reference for evaluating the performance of summarization models.

- `title` is the title of the article
- `anchor_text` is the brief introduction content of the article
- `raw_text` is the whole content of the article

A golden summary is a summary of multiple documents in a cluster that has been generated and verified by experts. In contrast, the test dataset follows the same structure but does not include a golden summary.

Table 4.3: Statistic of VLSP-ABMuSu dataset

| **Aspect** | **Training** | **Validation** | **Test** |
|---|---|---|---|
| Clusters | 200 | 100 | 300 |
| **Average** | | | |
| Doc | 3.105 | 3.04 | 3.05 |
| Sent per doc | 38.76 | 37.72 | 34.98 |
| Word per doc | 162.84 | 161.97 | 162.12 |
| Sent per sum | 4.94 | 4.82 | _ |
| Word per sum | 33.756 | 34.38 | _ |
| **Compression rate** | 0.207 | 0.212 | _ |

### 4.2.2 Evaluation metrics

In the task of automatic text summarization, it is necessary to evaluate the similarity between the predicted summary (hypothesis) and the true summary (reference). ROUGE [12] is an official metric for multiple summarization tasks and contests and is most widely used.

ROUGE is based on the concept of n-grams, which are contiguous sequences of words in a text. The ROUGE metrics compare the n-gram overlap between the generated and reference summaries. The most commonly used ROUGE metrics are ROUGE-N and ROUGE-L.

ROUGE-L measures the longest common subsequence (LCS) between the generated and reference summaries. A subsequence is a sequence of words appearing in the same order in both the generated and the reference summaries, but not necessarily consecutively. The LCS is the longest subsequence that appears in both the generated and reference summaries. The ROUGE scores range from 0 to 1, with a score of 1 indicating perfect similarity between the generated and reference summaries.

ROUGE scores can be expressed mathematically using the equation 4.1 and 4.3:

$$\text{ROUGE-n Precision} = \frac{|\text{Matched N-grams}|}{|\text{Predict summary N-grams}|} \tag{4.1}$$

$$\text{ROUGE-n Recall} = \frac{|\text{Matched N-grams}|}{|\text{Reference summary N-grams}|} \tag{4.2}$$

The ROUGE-L score can be calculated using the longest common subsequence

(LCS) between the candidate and reference sentences, as follows:

$$\text{ROUGE-L Precision} = \frac{LCS(S_g, S_r)}{|\text{Predict summary tokens}|} \tag{4.3}$$

$$\text{ROUGE-L Recall} = \frac{LCS(S_g, S_r)}{|\text{Predict summary tokens}|)} \tag{4.4}$$

The evaluation metric for the F1 score, as expressed in Formula 4.5, is created by taking the harmonic mean of the precision and recall scores.

$$F1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{4.5}$$

## 4.3 Experimental results

The extractive baseline model generates output with the highest ROUGE score when the weights of lexrank, textrank, and TF-IDF modules are 0.4, 0.4, and 0.2 correspondingly, sigma is set to 0.7 for the MMR component. The extractive baseline model achieved the ROUGE-2 F1 score of 0.293, better than all of the baseline models used in the competition, as shown in table 4.4. Grid search is applied to find these settings and configurations. The output of the model contains sentences that convey important information, but due to the essence of the extractive summary, the generated paragraphs are not coherent. A similar strategy is applied in the extractive module of the hybrid model.

Table 4.4: Previous work results and baseline model results on VLSP 2022

| **Model name** | **ROUGE-1** | | | **ROUGE-2** | | | **ROUGE-L** | | |
|---|---|---|---|---|---|---|---|---|---|
| | P | R | F1 | P | R | F1 | P | R | F1 |
| **ExtVLSP** | 0.437 | 0.643 | 0.508 | 0.242 | 0.414 | **0.293** | 0.408 | 0.598 | 0.473 |
| extractive_baseline | 0.461 | 0.551 | 0.484 | 0.246 | 0.327 | 0.265 | 0.422 | 0.504 | 0.442 |
| ruled_baseline | 0.455 | 0.507 | 0.464 | 0.252 | 0.295 | 0.258 | 0.420 | 0.468 | 0.428 |
| anchor_baseline | 0.521 | 0.390 | 0.432 | 0.230 | 0.133 | 0.184 | 0.466 | 0.350 | 0.389 |
| ViT5 base | **0.572** | 0.221 | 0.312 | 0.304 | 0.096 | 0.146 | 0.472 | 0.198 | 0.279 |

### 4.3.1 Proposed model results

The proposed model is designed to produce concise and informative summaries by leveraging multiple source documents. Additionally, the model benefits from a pre-

trained Vietnamese transformer-based language model to generate grammatically correct sentences. As shown in Table 4.5, the evaluation results demonstrate that the proposed model outperforms the base ViT5 model applied to the same dataset, achieving a ROUGE-2 F1 score of 0.184 and a ROUGE-L F1 score of 0.339. This score serves as a testament to the effectiveness of the proposed model in summarization tasks, highlighting its ability to effectively capture the most relevant information from the source documents and generate coherent and accurate summaries.

Table 4.5: Proposed model results and pre-trained ViT5 results on AbMusu dataset

| **Model name** | **ROUGE-1** | | | **ROUGE-2** | | | **ROUGE-L** | | |
|---|---|---|---|---|---|---|---|---|---|
| | P | R | F1 | P | R | F1 | P | R | F1 |
| **Proposed model** | 0.491 | 0.340 | 0.402 | 0.383 | 0.121 | **0.184** | 0.383 | 0.304 | **0.339** |
| No segmented | 0.476 | 0.256 | 0.333 | 0.356 | 0.109 | 0.167 | 0.423 | 0.230 | 0.298 |
| ViT5 base | **0.572** | 0.221 | 0.312 | 0.304 | 0.096 | 0.146 | 0.472 | 0.198 | 0.279 |

In addition to the previous experiments, the proposed model is also tested without remapping the extracted sentences back to their corresponding documents in the abstractive module. The result of this experiment shows a surprising decrease in the ROUGE-2 F1 score to 0.167, indicating that the remapping process is a crucial step for the model's performance. However, the precision on the ROUGE-L score increases significantly compared to the other two models as presented in the table. Moreover, it is interesting to note that the base ViT5 model exhibits the highest precision score on ROUGE-1. This observation implies that the proposed model may still have room for improvement and further optimization. Overall, filtering out less important sentences and remapping extracted ones is more effective, as it yields better results and output.

### 4.3.2 Model components contribution

Different configurations are applied for both extractive and abstractive modules. Table 4.6 shows the performances of comparative models. Based on the experiments on the validation set, Lexrank and Texrank are the two most important components in the sentence scoring step. Downgrading the weights of these two components receives a significant ROUGE score reduction. There also appear to be big differences in the document segmentation step if the output of the extractive module is segmented by the input document instead of combining all of them.

Figure 4.1 presents data on the impact of excluding different scoring methods on

Table 4.6: Proposed model results and other versions of model results

| Model name | ROUGE-1 | | | ROUGE-2 | | | ROUGE-L | | |
|---|---|---|---|---|---|---|---|---|---|
| | P | R | F1 | P | R | F1 | P | R | F1 |
| **Proposed model** | 0.491 | 0.340 | 0.402 | 0.164 | 0.208 | **0.184** | 0.383 | 0.304 | **0.339** |
| No TF-IDF | 0.413 | 0.305 | 0.351 | 0.293 | 0.131 | 0.182 | 0.601 | 0.136 | 0.223 |
| No Lexrank | 0.474 | 0.314 | 0.378 | 0.165 | 0.193 | 0.178 | 0.391 | 0.246 | 0.302 |
| No Texrank | 0.514 | 0.342 | 0.420 | 0.403 | 0.115 | 0.179 | 0.337 | 0.278 | 0.305 |
| No MMR | 0.511 | 0.236 | 0.323 | 0.213 | 0.157 | 0.181 | 0.471 | 0.249 | 0.326 |

the ROGUE-2 F1 score of the proposed model. The bar chart reveals that the exclusion of certain scoring methods results in a decline in the ROGUE-2 F1 score, indicating the importance of these methods in the proposed model.

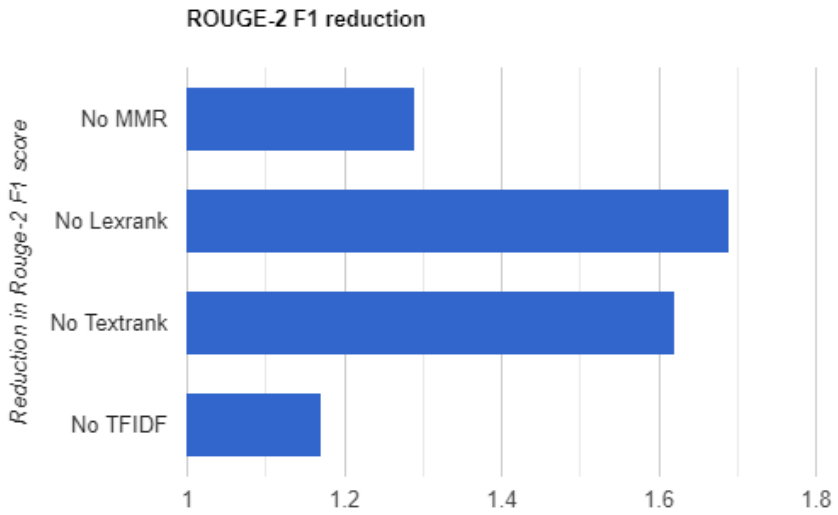


Figure 4.1: The reduction of ROUGE-2 F1 score for each scoring method after being excluded in the hybrid model

Specifically, the data shows that the Lexrank and Texrank scoring methods contribute significantly to the proposed model, as their exclusion leads to a substantial reduction in the ROGUE-2 F1 score. This suggests that these methods play a critical role in improving the summarization results of the proposed model.

The MMR scoring method also has a significant impact on the proposed model, as its exclusion results in a 1.3% reduction in the ROGUE-2 F1 score. This underscores the importance of MMR in the proposed model and its ability to further enhance the

summarization results.

On the other hand, the TF-IDF scoring method contributes relatively less to the proposed model, as its exclusion results in a huge reduction in the ROGUE-2 F1 score. This suggests that TF-IDF is still an essential method in the proposed model, although its impact is relatively minor compared to other scoring methods.

## 4.4 Error Analysis

To further evaluate the proposed model, the output summary of the model on the validation is analyzed.

Firstly, because the output from the extractive module is segmented into documents corresponding with the input documents then they are fed into the ViT5 model, duplicated data could appear. The input multi-document can contain duplicated data and the pre-process and extractive modules have no duplicate detection mechanism. For example, the proposed model produces a summary for cluster #65 in the validated set that has two sentences containing the same information. These sentences both describe the growth of the GDP in the first 6 months in Vietnam, which is 6.42%, which led to a less coherent summary. However, the non-segmented hybrid model produces output without duplicated information but because of short output summaries

Table 4.7: Example of error output of the model

| | |
|---|---|
| **Valid #65** | 7,72% là tốc độ tăng trưởng ấn tượng của kinh tế Việt Nam trong quý II năm nay, mức tăng cao nhất trong hơn một thập kỷ qua, góp phần **thúc đẩy GDP trong 6 tháng đầu năm tăng 6,42%**, vượt mục tiêu tăng trưởng 6 tháng đầu năm 2022 theo Nghị quyết 01/NQ-CP. Bức tranh kinh tế nửa đầu năm nay phần nhiều là những gam màu sáng. Đây là nhận định chung của nhiều chuyên gia, một số tổ chức trong và ngoài nước. Dựa trên kết quả tăng trưởng mạnh trong quý II/2022 và dữ liệu lịch sử, ngày 30/6, Ngân hàng UOB (Singapore) công bố nâng mức dự báo tăng trưởng GDP năm 2022 của Việt Nam lên 7%, từ mức 6,5% trước đó. GDP quý II tăng cao nhất thập kỷ, **GDP 6 tháng đầu năm tăng 6,42%**. |

Another problem appears because of extractive output segmentation, which is the connection between sentences. The different documents in the input might share stories about a specific topic but their main contents can be very much different. For example,

the output for cluster #126 in the training set includes a sentence that is supposed to end the whole summary but another sentence appears right after that. This is because it is the last sentence of the corresponding input document, the next sentence is from the generated summary for the next document.

Some important sentences can also be ignored after the extractive module is executed. This is because of the scoring methods or the threshold of the sentence extractor. Missing these important sentences cause the output summary to be less accurate, this problem usually appears in clusters containing short articles or fewer documents.

The proposed model encountered a challenge in processing non-textual data, such as images and videos, present in articles and news documents. As a consequence, advanced techniques related to vision and video processing are required to address this issue. The inability of the model to process non-textual data impedes its ability to generate comprehensive and accurate summaries of documents that contain such data. This limitation highlights the need for further research and development in this area to enable the proposed model to process and incorporate non-textual data into its summarization process.

# Conclusions

This thesis is centered around finding solutions to the multi-document summarization problem in Vietnamese language. To achieve this goal, a hybrid model is proposed that combines multiple extractive methods with a pre-trained abstractive model, ultimately resulting in short and meaningful summaries.

Several related studies were explored, and based on the findings, various extractive methods were utilized to filter out less significant sentences and improve the overall output of the abstractive summary. The primary objectives were to implement the extractive techniques and fine-tune the pre-trained model. The extractive module involved three essential steps: sentence scoring, sentence ranking, and sentence selection. Four different strategies were used for sentence scoring, namely Lexrank, Textrank, MMR, and TF-IDF. The abstractive module included a document segmentation and a pre-trained model.

The proposed model demonstrated a higher ROUGE-2 F1 score in comparison to the base ViT5 model on the AbMuSu dataset, with the graph-based strategies contributing the most to the extractive module. However, the model can be further improved by utilizing better extractive strategies, more advanced abstractive summarization methods, or more effective post-processing techniques.

# References


[1] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer Networks and ISDN Systems*, vol. 30, no. 1, pp. 107–117, 1998, proceedings of the Seventh International World Wide Web Conference. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016975529800110X

[2] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901.

[3] D.-C. Can, Q.-A. Nguyen, Q.-H. Duong, M.-Q. Nguyen, H.-S. Nguyen, L. N. T. Ngoc, Q.-T. Ha, and M.-V. Tran, "UETrice at MEDIQA 2021: A prosper-thy-neighbour extractive multi-document summarization model," in *Proceedings of the 20th Workshop on Biomedical Language Processing*. Online: Association for Computational Linguistics, Jun. 2021, pp. 311–319. [Online]. Available: https://aclanthology.org/2021.bionlp-1.36

[4] J. Carbonell and J. Stewart, "The use of mmr, diversity-based reranking for reordering documents and producing summaries," *SIGIR Forum (ACM Special Interest Group on Information Retrieval)*, 06 1999.

[5] M. Chen, W. Li, J. Liu, X. Xiao, H. Wu, and H. Wang, "SgSum:transforming multi-document summarization into sub-graph selection," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online

and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 4063–4074. [Online]. Available: https://aclanthology.org/2021.emnlp-main.333

[6] A. Cohan and N. Goharian, "Scientific document summarization via citation contextualization and scientific discourse," *International Journal on Digital Libraries*, vol. 19, pp. 1–17, 09 2018.

[7] E. Collins, I. Augenstein, and S. Riedel, "A supervised approach to extractive summarisation of scientific papers," in *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*. Vancouver, Canada: Association for Computational Linguistics, Aug. 2017, pp. 195–205. [Online]. Available: https://aclanthology.org/K17-1021

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019.

[9] W. S. El-Kassas, C. R. Salama, A. A. Rafea, and H. K. Mohamed, "Automatic text summarization: A comprehensive survey," *Expert Syst. Appl.*, vol. 165, p. 113679, 2021.

[10] G. Erkan and D. Radev, "Lexrank: Graph-based lexical centrality as salience in text summarization," *Journal of Artificial Intelligence Research - JAIR*, vol. 22, 09 2011.

[11] S. Gupta and C. D. Manning, "Improved pattern learning for bootstrapped entity extraction," in *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, 2014, pp. 98–108.

[12] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 74–81. [Online]. Available: https://aclanthology.org/W04-1013

[13] R. Mihalcea and P. Tarau, "TextRank: Bringing order into text," in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*. Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 404–411. [Online]. Available: https://aclanthology.org/W04-3252

[14] R. Nallapati, F. Zhai, and B. Zhou, "Summarunner: A recurrent neural network based sequence model for extractive summarization of documents," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, 11 2016.

[15] D. Q. Nguyen and A. Tuan Nguyen, "PhoBERT: Pre-trained language models for Vietnamese," in *Findings of the Association for Computational Linguistics: EMNLP 2020*. Online: Association for Computational Linguistics, Nov. 2020, pp. 1037–1042. [Online]. Available: https://aclanthology.org/2020.findings-emnlp.92

[16] V.-H. Nguyen, T.-C. Nguyen, M.-T. Nguyen, and N. X. Hoai, "Vnds: A vietnamese dataset for summarization," in *2019 6th NAFOSTED Conference on Information and Computer Science (NICS)*, 2019, pp. 375–380.

[17] L. Phan, H. Tran, H. Nguyen, and T. H. Trinh, "Vit5: Pretrained text-to-text transformer for vietnamese language generation," 2022.

[18] D. Radev, T. Allison, S. Blair-Goldensohn, J. Blitzer, A. Çelebi, S. Dimitrov, E. Drabek, A. Hakim, W. Lam, D. Liu, J. Otterbacher, H. Qi, H. Saggion, S. Teufel, M. Topper, A. Winkel, and Z. Zhang, "MEAD - a platform for multidocument multilingual text summarization," in *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*. Lisbon, Portugal: European Language Resources Association (ELRA), May 2004. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2004/pdf/757.pdf

[19] D. R. Radev, E. Hovy, and K. McKeown, "Introduction to the Special Issue on Summarization," *Computational Linguistics*, vol. 28, no. 4, pp. 399–408, 12 2002. [Online]. Available: https://doi.org/10.1162/089120102762671927

[20] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," 2020.

[21] K. Shinde, T. Roy, and T. Ghosal, "An extractive-abstractive approach for multi-document summarization of scientific articles for literature review," in *Proceedings of the Third Workshop on Scholarly Document Processing*. Gyeongju, Republic of Korea: Association for Computational Linguistics, Oct. 2022, pp. 204–209. [Online]. Available: https://aclanthology.org/2022.sdp-1.25

[22] M.-V. Tran, H.-Q. Le, D.-C. Can, and Q.-A. Nguyen, "Vlsp 2022 - ABMUSU Challenge: Vietnamese Abstractive multi-document summarization," in *Proceedings of*

*the 9th International Workshop on Vietnamese Language and Speech Processing (VLSP 2022)*, 2022.

[23] V. Tretyak and D. Stepanov, "Combination of abstractive and extractive approaches for summarization of long scientific texts," 2020.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[25] J. Zhang, Y. Zhao, M. Saleh, and P. J. Liu, "Pegasus: Pre-training with extracted gap-sentences for abstractive summarization," in *Proceedings of the 37th International Conference on Machine Learning*, ser. ICML'20. JMLR.org, 2020.