

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN**

**KHOA TOÁN CƠ TIN**

**🙟🕮🙝**

**ĐỀ TÀI: VIETNAMESE TEXT SUMMARIZER**



Sinh viên thực hiện : Nguyễn Trung Lâm

MSSV : **20001938**

... **tháng** ... **năm** ...

Contents

[Mở đầu 3](#_Toc167002859)

[Bối Cảnh Và Mục Tiêu 3](#_Toc167002860)

[Ý Nghĩa Của Nghiên Cứu 4](#_Toc167002861)

[Cấu Trúc Báo Cáo 4](#_Toc167002862)

[Xác Định Bài Toán 6](#_Toc167002863)

[Môi Trường Nghiên Cứu 6](#_Toc167002864)

[Phương Pháp Dự Kiến 6](#_Toc167002865)

[CHƯƠNG II. PHƯƠNG PHÁP 7](#_Toc167002866)

[Xử Lý Dữ Liệu Văn Bản 7](#_Toc167002867)

[Sử Dụng Mô Hình 7](#_Toc167002868)

[So sánh giữa 2 mô hình 8](#_Toc167002869)

[Xây Dựng Giao Diện FastAPI 9](#_Toc167002870)

[CHƯƠNG III. Quá trình huấn luyện 11](#_Toc167002871)

[Bộ dữ liệu 11](#_Toc167002872)

[Quy trình tiền xử lý 12](#_Toc167002873)

[Mô hình đào tạo 13](#_Toc167002874)

[CHƯƠNG IV. KẾT QUẢ THỬ NGHIÊM 15](#_Toc167002875)

[Công cụ đánh giá 15](#_Toc167002876)

[Kết quả sau khi đánh giá 2 mô hình trên tập kiểm thử 15](#_Toc167002877)

[Xây Dựng Giao Diện Bằng FastAPI 17](#_Toc167002878)

[CHƯƠNG V: HƯỚNG PHÁT TRIỂN 19](#_Toc167002879)

[Tài liệu tham khảo 20](#_Toc167002880)

# Mở đầu

Báo cáo này đưa ra một cái nhìn chi tiết và đầy đủ về quá trình phát triển mô hình tóm tắt văn bản tiếng Việt sử dụng mô hình ngôn ngữ lớn (Large Language Model - LLM) và xây dựng giao diện FastAPI. Bài toán này không chỉ đặt ra thách thức trong việc xử lý ngôn ngữ tự nhiên (NLP) tiếng Việt mà còn đòi hỏi sự tích hợp chặt chẽ giữa mô hình LLM và giao diện lập trình ứng dụng nhanh chóng, hiệu quả.

## Bối Cảnh Và Mục Tiêu

Việc tóm tắt văn bản tiếng Việt đang trở thành một vấn đề quan trọng trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP). Với sự bùng nổ của dữ liệu trên Internet và nhu cầu ngày càng tăng về việc tiếp cận thông tin một cách nhanh chóng và hiệu quả, việc phát triển các công cụ tóm tắt văn bản tiếng Việt là cần thiết.

Nhu cầu về Vietnamese Text Summarizer đặc biệt nổi lên trong các lĩnh vực như trích xuất tin tức, tổng hợp thông tin trong các bài báo khoa học, tóm tắt nội dung cho các ứng dụng di động, và hỗ trợ đọc hiệu quả cho người dùng không có thời gian hoặc khả năng đọc văn bản dày đặc

Cơ sở lý thuyết của Vietnamese Text Summarizer:

* Mô hình Transformer: Các mô hình Transformer đã chứng minh khả năng mạnh mẽ trong xử lý ngôn ngữ tự nhiên, bao gồm cả tác vụ tóm tắt văn bản. Bằng cách sử dụng các mô hình Transformer như BERT, GPT và PEGASUS, chúng ta có thể xây dựng các hệ thống tóm tắt văn bản tiếng Việt hiệu quả.
* Phương pháp Extractive và Abstractive: Có hai phương pháp chính để tạo ra tóm tắt văn bản: extractive (trích xuất) và abstractive (sáng tạo). Trong extractive summarization, các đoạn văn bản quan trọng được chọn và ghép lại để tạo ra tóm tắt. Trong khi đó, trong abstractive summarization, mô hình sẽ tự tạo ra các câu tóm tắt mới dựa trên nội dung của văn bản gốc.
* Xử lý Ngôn ngữ tự nhiên tiếng Việt: Để xây dựng Vietnamese Text Summarizer, cần phải có kiến thức về xử lý ngôn ngữ tự nhiên tiếng Việt, bao gồm việc phân tích cú pháp, phân tích ngữ cảnh và xử lý từ vựng đặc thù của tiếng Việt.

Bằng cách kết hợp các phương pháp và kiến thức trên, Vietnamese Text Summarizer có thể trở thành một công cụ quan trọng trong việc giúp người dùng tiếp cận thông tin một cách nhanh chóng và hiệu quả trong môi trường ngôn ngữ tiếng Việt.

## Ý Nghĩa Của Nghiên Cứu

Việc tóm tắt văn bản tiếng Việt hiệu quả không chỉ là một công cụ hữu ích để tối ưu hóa trải nghiệm người dùng trong việc tiếp cận thông tin một cách nhanh chóng và thuận tiện, mà còn đóng góp vào sự phát triển của lĩnh vực trí tuệ nhân tạo (AI) và xử lý ngôn ngữ tự nhiên (NLP). Trong thời đại số hóa ngày nay, dữ liệu văn bản ngày càng tăng lên với tốc độ chóng mặt trên Internet, từ các bài báo, blog, tài liệu học thuật cho đến các nội dung trên mạng xã hội. Việc tiếp cận và xử lý lượng thông tin lớn này một cách hiệu quả trở thành một thách thức lớn đối với người dùng cuối cũng như các nhà nghiên cứu và doanh nghiệp.

Bằng cách kết hợp sức mạnh của các mô hình ngôn ngữ lớn như BERT, GPT và PEGASUS với công nghệ giao diện nhanh và hiệu quả của FastAPI, nghiên cứu này mong muốn cung cấp những giải pháp tiên tiến và hữu ích trong việc quản lý và truy xuất thông tin từ các nguồn dữ liệu phong phú. Vietnamese Text Summarizer không chỉ giúp người dùng tiết kiệm thời gian và công sức trong việc tiếp cận thông tin, mà còn mở ra nhiều tiềm năng và cơ hội mới trong lĩnh vực AI và NLP.

## Cấu Trúc Báo Cáo

Báo cáo này sẽ bao gồm các phần chính như Mô Tả Về Bài Toán, Phương Pháp, Quá Trình Huấn Luyện, Kết Quả Thử Nghiệm và Hướng Phát Triển Tiếp Theo. Mỗi phần sẽ được trình bày chi tiết về quá trình phát triển, các quyết định quan trọng đã được đưa ra và những bài học quý báu từ quá trình nghiên cứu.

Trong phần Mô Tả Về Bài Toán, chúng tôi sẽ giải thích tại sao việc tóm tắt văn bản là một bài toán quan trọng và cần thiết trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP), cùng với các ứng dụng thực tiễn của nó. Chúng tôi cũng sẽ mô tả chi tiết về bài toán cụ thể mà chúng tôi đã chọn và những thách thức đặt ra.

Phần Phương Pháp sẽ đưa ra các chiến lược và phương pháp chúng tôi đã sử dụng để giải quyết bài toán tóm tắt văn bản, bao gồm cả việc lựa chọn mô hình ngôn ngữ lớn (LLM) phù hợp và phương pháp tiếp cận để huấn luyện và sử dụng mô hình đó.

Trong phần Quá Trình Huấn Luyện, chúng tôi sẽ trình bày chi tiết về quá trình chuẩn bị dữ liệu, lựa chọn các siêu tham số, và các kỹ thuật tinh chỉnh mô hình đã thực hiện trong quá trình huấn luyện.

Phần Kết Quả Thử Nghiệm sẽ trình bày kết quả và đánh giá của hệ thống dựa trên các thử nghiệm thực tế và tiêu chí đánh giá đã đề ra.

Cuối cùng, phần Hướng Phát Triển Tiếp Theo sẽ đề xuất các hướng phát triển và nghiên cứu tiếp theo dựa trên những kết quả và bài học từ dự án này.

Trong quá trình mô tả và trình bày, chúng tôi sẽ linh hoạt sử dụng thuật ngữ "mô hình ngôn ngữ lớn" và "LLM" để mô tả các công nghệ và kỹ thuật chúng tôi đã sử dụng. Điều này giúp chúng tôi truyền đạt thông tin một cách chính xác và linh hoạt tùy thuộc vào ngữ cảnh cụ thể của mô hình.

CHƯƠNG I. MÔ TẢ BÀI TOÁN

## Xác Định Bài Toán

Bài toán chúng tôi nghiên cứu nhằm phát triển một mô hình tóm tắt văn bản tiếng Việt sử dụng mô hình ngôn ngữ lớn (Large Language Model - LLM) và xây dựng giao diện FastAPI để triển khai mô hình này. Thách thức đặt ra không chỉ là xử lý ngôn ngữ tự nhiên tiếng Việt một cách hiệu quả mà còn phải tích hợp mô hình này vào một hệ thống có thể hoạt động trơn tru trong các ứng dụng thực tế.

## Môi Trường Nghiên Cứu

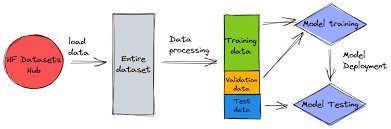
Chúng tôi đã sử dụng một bộ dữ liệu đa dạng chứa các văn bản tiếng Việt thuộc nhiều thể loại và chủ đề khác nhau để đảm bảo sự đa dạng và phong phú của dữ liệu huấn luyện. Môi trường nghiên cứu của chúng tôi được xây dựng trên nền tảng của các công cụ và thư viện phổ biến như PyTorch, Transformers của Hugging Face, và FastAPI. Sự linh hoạt và tiện ích của các công cụ này giúp chúng tôi dễ dàng thực hiện các thử nghiệm và tối ưu hóa mô hình tóm tắt văn bản tiếng Việt.

## Phương Pháp Dự Kiến

Chúng tôi sử dụng các mô hình ngôn ngữ lớn (LLM) tiên tiến như BERT, GPT hoặc các biến thể của chúng được huấn luyện đặc biệt cho ngôn ngữ tiếng Việt. Dữ liệu văn bản được xử lý và chuẩn bị để huấn luyện mô hình với mục tiêu tóm tắt văn bản một cách chính xác và hiệu quả. Giao diện FastAPI được xây dựng để cung cấp các dịch vụ tóm tắt văn bản, giúp dễ dàng tích hợp mô hình vào các ứng dụng thực tế. Chúng tôi cũng sẽ tiến hành các thử nghiệm và đánh giá hiệu suất của mô hình để đảm bảo chất lượng tóm tắt.

# CHƯƠNG II. PHƯƠNG PHÁP

Phần này trình bày phương pháp sử dụng mô hình ngôn ngữ lớn (LLM) ViT5-large-vietnews-summarization của VietAI để tóm tắt văn bản tiếng Việt, đồng thời xây dựng giao diện FastAPI cho mô hình này. Kiến trúc tổng thể của phương pháp được minh họa trong hình dưới đây. Đầu tiên, chúng tôi xử lý dữ liệu văn bản và chuẩn bị dữ liệu huấn luyện cho mô hình (Phần II.1). Tiếp theo, chúng tôi sử dụng mô hình ViT5 để thực hiện quá trình tóm tắt văn bản (Phần II.2). Cuối cùng, chúng tôi xây dựng giao diện FastAPI để triển khai và cung cấp dịch vụ tóm tắt văn bản (Phần II.3).



Hình 1: Tổng quan về phương pháp đề xuất

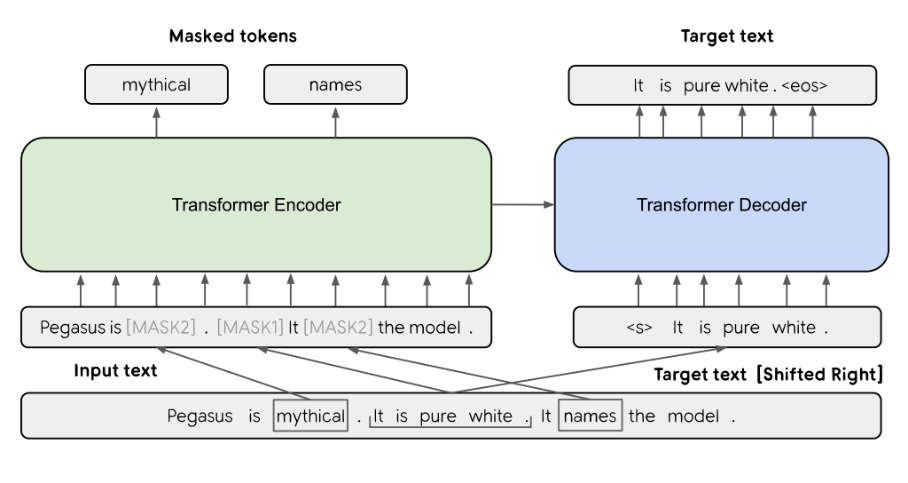
## Xử Lý Dữ Liệu Văn Bản

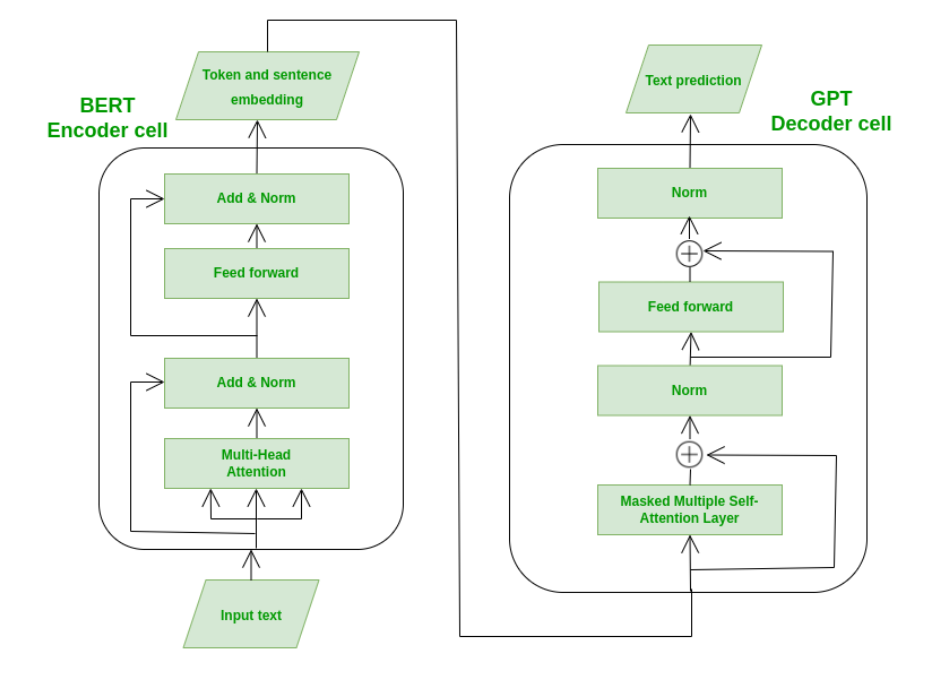
Chúng tôi khởi đầu bằng việc xử lý và chuẩn bị dữ liệu văn bản tiếng Việt. Bước này bao gồm việc thu thập, làm sạch và tiền xử lý dữ liệu. Chúng tôi sử dụng các phương pháp tiền xử lý như loại bỏ ký tự đặc biệt, tách từ, và chuẩn hóa văn bản để đảm bảo dữ liệu đầu vào cho mô hình ViT5 là sạch và có cấu trúc phù hợp. Dữ liệu sau khi được xử lý sẽ được chia thành tập huấn luyện và tập kiểm tra để đánh giá hiệu quả của mô hình.

## Sử Dụng Mô Hình

Chúng tôi so sánh 2 mô hình ViT5-large-vietnews-summarization của VietAI - một biến thể của mô hình Pegasus đã được fine-tuning để tóm tắt văn bản tiếng Việt, và bart-large-cnn của Facebook - là một phiên bản của mô hình BART (Bidirectional and Auto-Regressive Transformers) được đào tạo trên tập dữ liệu của Facebook

Cả 2 mô hình này đã được đào tạo trên 1 lượng lớn ngôn ngữ tự nhiên tiếng Việt, và có khả năng tạo ra các bản tóm tắt ngắn gọn, chính xác từ các đoạn văn bản dài. ViT5 và BARTtận dụng kiến trúc Transformer để học các mối quan hệ phức tạp giữa các từ và câu trong văn bản, từ đó tạo ra các bản tóm tắt có ý nghĩa.





Hình 3: Kiến trúc mô hình ViT5 (trên) và BART (dưới)

## So sánh giữa 2 mô hình

Giống nhau :

1. Kiến Trúc Encoder-Decoder

* Cả hai mô hình ViT5-large-vietnews-summarization và bart-large-cnn đều sử dụng kiến trúc encoder-decoder:
* Encoder: Cả hai mô hình đều có một encoder để mã hóa văn bản đầu vào thành một biểu diễn ngữ nghĩa sâu.
* Decoder: Cả hai mô hình sử dụng một decoder để sinh văn bản đầu ra dựa trên biểu diễn ngữ nghĩa từ encoder.

1. Cơ Chế Attention

* Cả hai mô hình đều sử dụng cơ chế attention để cải thiện khả năng mô hình tập trung vào các phần quan trọng của văn bản đầu vào khi tạo ra văn bản đầu ra:
* Self-Attention: Cả encoder và decoder của hai mô hình đều sử dụng self-attention để tạo ra mối quan hệ giữa các từ trong văn bản.
* Multi-Head Attention: Cả hai mô hình đều áp dụng multi-head attention để nắm bắt được nhiều khía cạnh khác nhau của ngữ cảnh.

1. Ứng Dụng

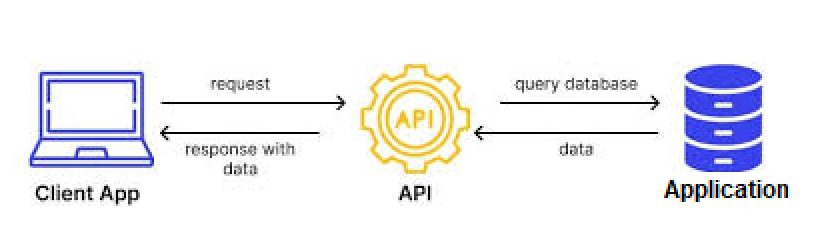
* Cả hai mô hình đều được thiết kế cho các tác vụ xử lý ngôn ngữ tự nhiên như tóm tắt văn bản, dịch thuật, và sinh văn bản tự nhiên.

Khác nhau

|  |  |  |
| --- | --- | --- |
|  | ViT5 | BART |
| Thiết kế | ViT5 giữ nguyên kiến trúc T5, sử dụng một bộ mã hóa (encoder) và một bộ giải mã (decoder) đơn giản nhưng hiệu quả, phù hợp cho nhiều tác vụ khác nhau. | BART sử dụng kiến trúc encoder của BERT, giúp hiểu ngữ cảnh hai chiều, và decoder của GPT, giúp sinh văn bản một cách tuần tự. |
| Kỹ Thuật Huấn Luyện | T5 được huấn luyện để thực hiện nhiều tác vụ xử lý ngôn ngữ khác nhau dưới dạng bài toán chuyển đổi văn bản sang văn bản. ViT5 được huấn luyện đặc biệt trên dữ liệu tiếng Việt, tối ưu hóa cho ngôn ngữ này. | BART sử dụng các hàm noise như token masking, token deletion, và sentence permutation trong quá trình huấn luyện để làm méo văn bản đầu vào, cải thiện khả năng phục hồi và độ chính xác của mô hình. BART được huấn luyện trên bộ dữ liệu lớn gồm nhiều ngôn ngữ khác nhau trong đó có cả tiếng Việt |

## Xây Dựng Giao Diện FastAPI

Để triển khai mô hình tóm tắt văn bản, chúng tôi xây dựng một giao diện FastAPI. FastAPI là một framework mạnh mẽ và linh hoạt, cho phép chúng tôi dễ dàng tạo ra các API RESTful để cung cấp dịch vụ tóm tắt văn bản. Giao diện này bao gồm các endpoint để nhận yêu cầu tóm tắt văn bản, xử lý yêu cầu bằng mô hình ViT5, và trả về kết quả tóm tắt cho người dùng. Quá trình này được thiết kế để đảm bảo hiệu suất cao và đáp ứng nhanh chóng các yêu cầu từ người dùng.



Hình 4: Kiến trúc giao diện FastAPI

Giao diện FastAPI được xây dựng nhằm mục đích triển khai mô hình tóm tắt văn bản một cách hiệu quả, đồng thời dễ dàng tích hợp vào các ứng dụng thực tế.

# CHƯƠNG III. Quá trình huấn luyện

Phần này sẽ tập trung trình bày về quá trình chọn lọc bộ dữ liệu, tiền xử lý dữ liệu và cấu trúc cũng như quá trình huấn luyện mô hình của chúng tôi.

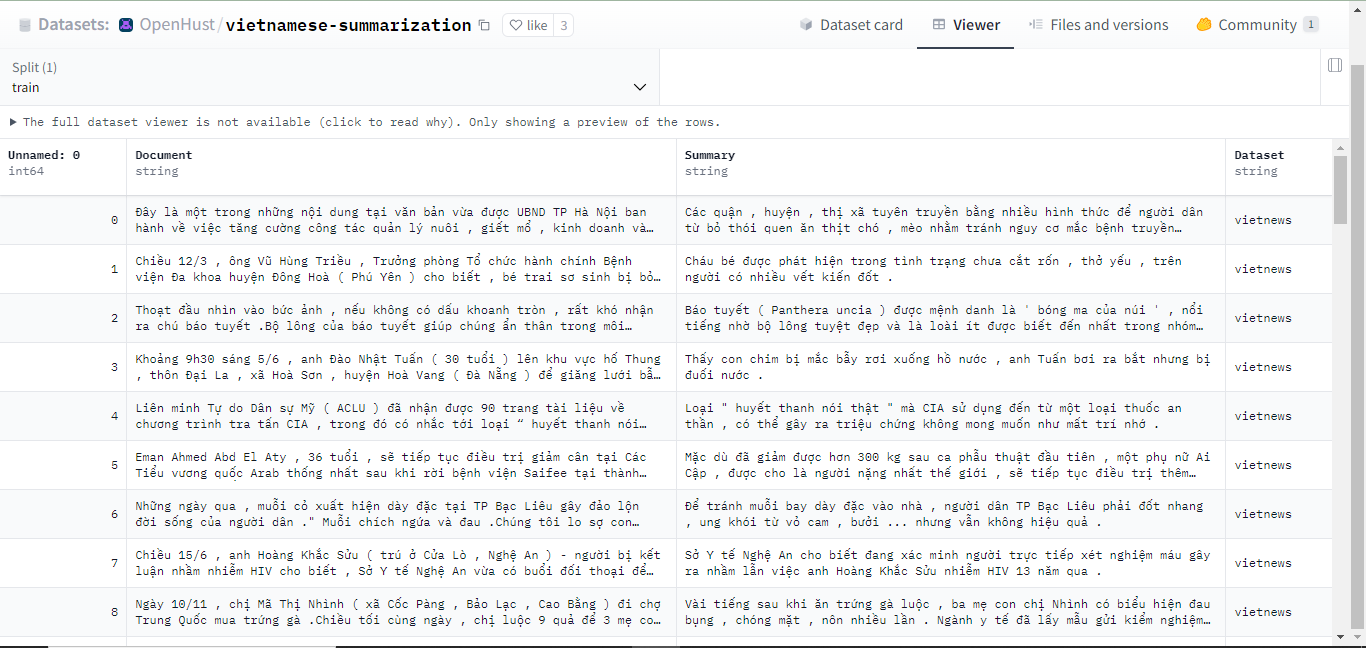
## Bộ dữ liệu

Bộ dữ liệu "Vietnamese Summarization" trên Hugging Face, do OpenHust phát triển, là một bộ dữ liệu chứa các văn bản và bản tóm tắt tương ứng bằng tiếng Việt. Bộ dữ liệu này nhằm phục vụ cho việc huấn luyện và đánh giá các mô hình tóm tắt văn bản tự động.

OpenHust, được thành lập bởi Đại học Bách Khoa Hà Nội (HUST) với sự hỗ trợ từ Tập đoàn Naver của Hàn Quốc, là một tổ chức nghiên cứu và phát triển AI hàng đầu tại Việt Nam. Trung tâm này tập trung vào việc nghiên cứu, đào tạo và ứng dụng trí tuệ nhân tạo, góp phần nâng cao chất lượng nguồn nhân lực AI và thúc đẩy sự phát triển của công nghệ này trong khu vực và trên thế giới.

Bộ dữ liệu "Vietnamese Summarization" bao gồm nhiều loại tài liệu khác nhau nhằm mục đích cải thiện hiệu suất của các mô hình tóm tắt văn bản tiếng Việt. Dữ liệu được chia thành hai phần chính: tập huấn luyện và tập kiểm tra, với số lượng mục từ 10,000 đến 100,000. Việc sử dụng bộ dữ liệu này giúp các nhà nghiên cứu và phát triển kiểm tra và cải thiện các thuật toán tóm tắt văn bản một cách hiệu quả.

Dữ liệu được chia theo tỷ lệ 8:2:2 tương ứng với tập train, validation và test.



Hình 5: Bộ dữ liệu Vietnamese Summarization

## Quy trình tiền xử lý

Quy trình tiền xử lý dữ liệu là một bước quan trọng trong việc chuẩn bị dữ liệu cho các mô hình học máy và các ứng dụng xử lý ngôn ngữ tự nhiên (NLP). Đối với tiếng Việt, quy trình này bao gồm loại bỏ dấu câu, chuyển đổi văn bản thành chữ thường, và loại bỏ các từ dừng (stopword) để làm sạch dữ liệu và tối ưu hóa hiệu suất của các mô hình sau này. Dưới đây là một phác thảo về quy trình này:

Loại bỏ dấu câu:

* Dấu câu không mang lại nhiều ý nghĩa trong việc phân tích văn bản hoặc huấn luyện mô hình.
* Các dấu câu như chấm, phẩy, dấu chấm hỏi, dấu chấm than, dấu hai chấm, ngoặc đơn, ngoặc kép, và dấu gạch ngang thường được loại bỏ.

Chuyển đổi thành chữ thường:

* Việc chuyển đổi tất cả các từ trong văn bản thành chữ thường giúp đồng nhất dữ liệu.
* Ví dụ: "Chào" và "chào" sẽ được xem xét như là cùng một từ khi chúng được chuyển đổi thành chữ thường.

Loại bỏ stopword:

* Các stopword là các từ phổ biến như "và", "của", "là", "một", "ở", không mang lại nhiều ý nghĩa trong việc phân tích văn bản và có thể được loại bỏ.
* Một danh sách các stopword tiếng Việt có thể được sử dụng để loại bỏ các từ này từ văn bản.

Tokenization (Phân đoạn):

* Sau khi loại bỏ dấu câu và stopword, văn bản cần được phân đoạn thành các từ riêng biệt hoặc các "token".
* Tokenization có thể được thực hiện bằng cách chia văn bản thành các từ dựa trên khoảng trắng hoặc các quy tắc ngữ cảnh.

vnTokenizer:

* vnTokenizer là một công cụ mã nguồn mở được phát triển bởi nhóm VNU-LF và được thiết kế đặc biệt cho tiếng Việt.
* Thư viện này cung cấp các chức năng tokenization và phân đoạn từ.

Xử lý đặc biệt (nếu cần):

* Trong một số trường hợp, có thể cần xử lý các trường hợp đặc biệt như chuỗi ký tự liên kết (URLs), số điện thoại, địa chỉ email, v.v.
* Điều này giúp đồng nhất dữ liệu và loại bỏ thông tin không cần thiết.

Sau khi hoàn thành quy trình tiền xử lý, dữ liệu sẽ được làm sạch và sẵn sàng cho việc sử dụng trong các mô hình học máy hoặc các ứng dụng NLP khác. Quy trình này không chỉ giúp cải thiện hiệu suất của mô hình mà còn giảm thiểu nhiễu và tăng tính đáng tin cậy của kết quả.

## Mô hình đào tạo

Mục tiêu của quá trình huấn luyện là phát triển hai mô hình tổng hợp văn bản, bao gồm mô hình bart-large-cnn và ViT5-large-vietnews-summarization, để thực hiện tác vụ tóm tắt văn bản. Cả hai mô hình này được huấn luyện trên cùng một bộ dữ liệu tiếng Việt và cùng 1 bộ siêu tham số để so sánh hiệu suất và khả năng tổng hợp của chúng.

Trong quá trình huấn luyện, chúng tôi đã sử dụng các siêu tham số sau:

* Number of Epochs: 5
* Warmup Steps: 500
* Per Device Train Batch Size: 4
* Per Device Evaluation Batch Size: 4
* Weight Decay: 0.01
* Logging Steps: 10
* Evaluation Strategy: Steps
* Evaluation Steps: 500
* Save Steps: 1,000,000
* Gradient Accumulation Steps: 16

Để huấn luyện mô hình **bart-large-cnn**, mỗi epoch mất khoảng 1.088 giờ để hoàn thành. Mô hình **ViT5-large-vietnews-summarization** mất 0.703 giờ cho mỗi epoch.

# CHƯƠNG IV. KẾT QUẢ THỬ NGHIÊM

## Công cụ đánh giá

Ở nghiên cứu này, chúng tôi sử dụng công cụ đánh giá ROUGE (Recall-Oriented Understudy for Gisting Evaluation). ROUGE là một tập hợp các số liệu phổ biến được sử dụng để đánh giá chất lượng của các văn bản được tạo ra bởi các mô hình ngôn ngữ, đặc biệt là trong các tác vụ như tóm tắt văn bản.

Các Số Liệu ROUGE

* ROUGE-1 (rouge1): Đo lường sự trùng khớp giữa các từ đơn lẻ trong văn bản tóm tắt tự động với văn bản tóm tắt tham chiếu. Chỉ số này đánh giá mức độ chính xác của từng từ trong tóm tắt.
* ROUGE-2 (rouge2): Đo lường sự trùng khớp giữa các cặp từ liên tiếp (bi-gram) trong văn bản tóm tắt tự động với văn bản tóm tắt tham chiếu. Chỉ số này đánh giá mức độ chính xác của các cặp từ liên tiếp.
* ROUGE-L (rougeL): Đo lường độ dài của chuỗi con chung dài nhất (Longest Common Subsequence - LCS) giữa văn bản tóm tắt tự động và văn bản tóm tắt tham chiếu. Chỉ số này đánh giá mức độ trùng khớp về mặt cấu trúc giữa hai văn bản.
* ROUGE-LSum (rougeLsum): Tương tự như ROUGE-L, nhưng được tính toán dựa trên việc tổng hợp các chuỗi con chung dài nhất trên nhiều câu trong văn bản. Chỉ số này đặc biệt hữu ích cho các bài toán tóm tắt đa văn bản.

## Kết quả sau khi đánh giá 2 mô hình trên tập kiểm thử

| **Mô hình** | **ROUGE-1** | **ROUGE-2** | **ROUGE-L** | **ROUGE-LSum** |
| --- | --- | --- | --- | --- |
| ViT5 | 0.627278 | 0.402359 | 0.251823 | 0.250732 |
| Pegasus | 0.502396 | 0.390249 | 0.269933 | 0.269575 |

Hình 6: Kết quả sau khi đánh giá 2 mô hình trên tập kiểm thử

ViT5 có hiệu suất tốt hơn về các chỉ số ROUGE-1 và ROUGE-2, cho thấy nó có khả năng tái hiện lại các từ đơn lẻ và cặp từ liên tiếp trong tóm tắt tham chiếu tốt hơn. Điều này có thể cho thấy ViT5 mạnh hơn trong việc nắm bắt các chi tiết nhỏ và các mối quan hệ ngữ cảnh gần.

Pegasus có hiệu suất tốt hơn về các chỉ số ROUGE-L và ROUGE-LSum, cho thấy nó có khả năng duy trì cấu trúc và tính liền mạch của tóm tắt tốt hơn. Điều này có thể cho thấy Pegasus mạnh hơn trong việc tạo ra các tóm tắt có cấu trúc và mạch lạc hơn.

# Xây Dựng Giao Diện Bằng FastAPI

Chúng tôi đã sử dụng FastAPI để phát triển giao diện cho hệ thống tóm tắt văn bản của mình, nhằm mang lại một trải nghiệm tối ưu và dễ dàng cho người dùng trong quá trình tạo ra các tóm tắt văn bản tự động từ các đoạn văn bản đầu vào. Giao diện này không chỉ là một phần của hệ thống mà còn là cầu nối quan trọng giữa người dùng và tính năng chính của ứng dụng.

* Form Đầu Vào: Giao diện cung cấp một form đầu vào linh hoạt, cho phép người dùng nhập đoạn văn bản cần tóm tắt dễ dàng. Điều này có thể bao gồm cả việc cho phép người dùng nhập văn bản trực tiếp vào trường văn bản hoặc tải lên tệp tin văn bản từ máy tính của họ. Điều này tạo điều kiện thuận lợi cho việc tóm tắt văn bản từ nhiều nguồn khác nhau và đa dạng.
* Nút Tóm Tắt: Sau khi nhập đoạn văn bản, người dùng có thể nhấn vào nút "Tóm Tắt" để gửi yêu cầu tạo tóm tắt văn bản đến hệ thống. Nút này là điểm kết nối giữa người dùng và tính năng tóm tắt, giúp họ dễ dàng truy cập và sử dụng tính năng này.
* Hiển Thị Kết Quả: Khi kết quả được tính toán và trả về từ hệ thống, giao diện sẽ hiển thị tóm tắt văn bản tự động được tạo ra. Kết quả này được hiển thị một cách rõ ràng và dễ đọc, giúp người dùng nhanh chóng hiểu được nội dung tóm tắt một cách chính xác và đầy đủ.
* Tùy Chọn Tinh Chỉnh: Giao diện cung cấp các tùy chọn tinh chỉnh cho người dùng, cho phép họ điều chỉnh các tham số như độ dài tóm tắt, mô hình sử dụng, hoặc các yếu tố khác để tùy chỉnh quá trình tạo tóm tắt. Điều này tạo điều kiện linh hoạt và cá nhân hóa cho người dùng, giúp họ tạo ra các tóm tắt văn bản chính xác và phù hợp với nhu cầu của họ.

Ngoài ra, chúng tôi sẽ tiếp tục phát triển và cải thiện giao diện bằng cách thêm các tính năng mới và tối ưu hóa hiệu suất của nó. Các tính năng mới có thể bao gồm việc hỗ trợ nhiều ngôn ngữ, tích hợp công nghệ AI mới nhằm cải thiện chất lượng tóm tắt, hoặc thêm các tùy chọn tùy chỉnh và cải tiến giao diện người dùng để tăng cường trải nghiệm người dùng.

Giao diện của chúng tôi không chỉ là một phần của hệ thống mà còn là cầu nối quan trọng giữa người dùng và tính năng chính của ứng dụng, và chúng tôi cam kết tiếp tục cải thiện và phát triển nó để đáp ứng nhu cầu ngày càng đa dạng và phức tạp của người dùng

# CHƯƠNG V: HƯỚNG PHÁT TRIỂN

Hướng phát triển tiếp theo của Vietnamese Text Summarizer có thể tập trung vào các mặt sau:

* Ứng dụng vào các lĩnh vực cụ thể: Tiếp tục nghiên cứu và phát triển Vietnamese Text Summarizer để ứng dụng vào các lĩnh vực cụ thể như y tế, giáo dục, pháp luật và doanh nghiệp. Điều này đòi hỏi sự tùy chỉnh và tối ưu hóa của hệ thống để đáp ứng các nhu cầu đặc thù của từng lĩnh vực, từ đó cung cấp các giải pháp hiệu quả và phù hợp.
* Tăng cường tính đa dạng và linh hoạt: Nâng cao khả năng của Vietnamese Text Summarizer để xử lý nhiều loại văn bản và ngữ cảnh khác nhau. Điều này có thể bao gồm việc cải thiện khả năng tổng hợp tin tức, trích xuất thông tin từ các văn bản học thuật, và xử lý các nội dung đa dạng trên mạng xã hội.
* Cải tiến hiệu suất và chất lượng: Tiếp tục tối ưu hóa Vietnamese Text Summarizer để cải thiện hiệu suất và chất lượng của hệ thống. Mặc dù có hạn chế về tài nguyên, nhưng việc tinh chỉnh các siêu tham số, sử dụng kỹ thuật tối ưu hóa hiệu suất và phát triển các phương pháp mới có thể giúp đạt được kết quả tốt nhất

Bằng cách tập trung vào các hướng phát triển này, Vietnamese Text Summarizer có thể trở thành một công cụ quan trọng và hiệu quả trong việc quản lý và truy xuất thông tin, đồng thời đáp ứng được nhu cầu đa dạng của người dùng trong các lĩnh vực cụ thể.

# Tài liệu tham khảo

1. Google Research. (2019). PEGASUS: A Pre-trained Contextual Neural Machine Translation Model with Multi-Task Learning. ArXiv. Link: https://arxiv.org/abs/1912.08777
2. Google Research. (2019). Leveraging Pre-trained Checkpoints for Sequence Generation Tasks. ArXiv. Link: https://arxiv.org/abs/1907.12461
3. Zhou, A., Cui, X., Zhang, J., Yang, C., Liu, Z., Wang, L., & Wang, W. (2021). Pre-training Transformers as Energy-Based Cloze Models. ArXiv. Link: https://arxiv.org/abs/2103.03404
4. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... & Stoyanov, V. (2019). BertSum: Text Summarization with Pretrained Encoders. ArXiv. Link: https://arxiv.org/abs/1903.10318
5. Yang, Z., Hu, Z., Dyer, C., Xing, E., & Berg-Kirkpatrick, T. (2019). Text Summarization with Pretrained Encoders. ArXiv. Link: https://arxiv.org/abs/1908.08345
6. See, A., Liu, P. J., & Manning, C. D. (2017). Get To The Point: Summarization with Pointer-Generator Networks. ArXiv. Link: https://arxiv.org/abs/1704.04368
7. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is All You Need. ArXiv. Link: https://arxiv.org/abs/1706.03762