

**-----------------------------------**

**NGUYỄN ĐÌNH SƠN**

PHÁT TRIỂN MÔ HÌNH TEXT MINING DỰA TRÊN

KỸ THUẬT MACHINE LEARNING CHO TÓM TẮT

VĂN BẢN TIẾNG VIỆT

**ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT**

**(Theo định hướng ứng dụng)**

**TP. HỒ CHÍ MINH – NĂM 2024**

**--------------------------------------**



**NGUYỄN ĐÌNH SƠN**

PHÁT TRIỂN MÔ HÌNH TEXT MINING DỰA TRÊN

KỸ THUẬT MACHINE LEARNING CHO TÓM TẮT

VĂN BẢN TIẾNG VIỆT

Chuyên ngành: Hệ thống thông tin Mã số: 8.48.01.04

**ĐỀ ÁN TỐT NGHIỆP THẠC SĨ KỸ THUẬT**

**(Theo định hướng ứng dụng)**

**NGƯỜI HƯỚNG DẪN KHOA HỌC: TS. TÂN HẠNH**

**TP. HỒ CHÍ MINH – NĂM 2024**

# LỜI CAM ĐOAN

Tôi cam đoan rằng đề án tốt nghiệp thạc sĩ: ***“Phát triển mô hình Text Mining dựa trên kĩ thuật Machine Learning cho tóm tắt văn bản tiếng Việt”*** là công trình nghiên cứu của chính tôi.

Các số liệu, kết quả nêu trong đề án là trung thực và chưa từng được ai công bố trong bất kỳ công trình nào khác.

Không có sản phẩm/nghiên cứu nào của người khác được sử dụng trong đề án này mà không được trích dẫn theo đúng quy định.

TP. Hồ Chí Minh, ngày 18 tháng 08 năm 2024

|  |  |  |  |
| --- | --- | --- | --- |
| **Học viên thực hiện đề án** | | | |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| **Nguyễn Đình Sơn** | | | |

# LỜI CẢM ƠN

Trong suốt quá trình học tập và nghiên cứu thực hiện đề án tốt nghiệp thạc sĩ, ngoài nỗ lực của bản thân, tôi đã nhận được sự hướng dẫn nhiệt tình quý báu của quý Thầy Cô, cùng với sự động viên và ủng hộ của gia đình, bạn bè và đồng nghiệp. Với lòng kính trọng và biết ơn sâu sắc, tôi xin gửi lời cảm ơn chân thành tới:

**Thầy TS. Tân Hạnh**, người thầy kính yêu đã hết lòng giúp đỡ, hướng dẫn, động viên, tạo điều kiện cho tôi trong suốt quá trình thực hiện và hoàn thành đề án tốt nghiệp thạc sĩ.

Ban Giám Đốc, Phòng đào tạo sau đại học và quý Thầy Cô đã tạo mọi điều kiện thuận lợi giúp tôi hoàn thành đề án.

Tôi xin chân thành cảm ơn gia đình, bạn bè, đồng nghiệp trong cơ quan đã động viên, hỗ trợ tôi trong lúc khó khăn để tôi có thể học tập và hoàn thành đề án. Mặc dù đã có nhiều cố gắng, nỗ lực, nhưng do thời gian và kinh nghiệm nghiên cứu khoa học còn hạn chế nên không thể tránh khỏi những thiếu sót. Tôi rất mong nhận được sự góp ý của quý Thầy Cô cùng bạn bè đồng nghiệp để kiến thức của tôi ngày một hoàn thiện hơn.

Xin chân thành cảm ơn!

TP. Hồ Chí Minh, ngày 15 tháng 08 năm 2024

|  |  |  |  |
| --- | --- | --- | --- |
| **Học viên thực hiện đề án** | | | |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| **Nguyễn Đình Sơn** | | | |

# MỤC LỤC

[LỜI CAM ĐOAN 1](#_Toc173690341)

[LỜI CẢM ƠN 2](#_Toc173690342)

[MỤC LỤC 3](#_Toc173690343)

[DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT 4](#_Toc173690344)

[DANH SÁCH BẢNG 5](#_Toc173690345)

[DANH SÁCH HÌNH VẼ 6](#_Toc173690346)

[MỞ ĐẦU 7](#_Toc173690347)

[1. Tính cấp thiết của đề tài 7](#_Toc173690348)

[2. Tổng quan về vấn đề nghiên cứu 7](#_Toc173690349)

[2.1. Khái quát ngắn gọn tổng quan về vấn đề nghiên cứu 7](#_Toc173690350)

[2.2. Khảo sát các công trình liên quan 8](#_Toc173690351)

[2.3. Mục đích nghiên cứu 9](#_Toc173690352)

[3. Đối tượng và phạm vi nghiên cứu 10](#_Toc173690353)

[3.1 Đối tượng nghiên cứu 10](#_Toc173690354)

[3.2 Phạm vi nghiên cứu 10](#_Toc173690355)

[4. Phương pháp nghiên cứu 10](#_Toc173690356)

[5. Bố cục đề án 11](#_Toc173690357)

[CHƯƠNG 1: NGHIÊN CỨU TỔNG QUAN 12](#_Toc173690358)

[1. Tổng quan về Text Mining 12](#_Toc173690359)

[2. Sự khác nhau giữa Text Mining và Data Mining 13](#_Toc173690360)

[3. Tại sao sự phổ biến của việc sử dụng Text Mining như một công cụ phân tích ngày càng tăng? 14](#_Toc173690361)

[4. Một số lĩnh vực ứng dụng phổ biến nhất. 14](#_Toc173690362)

[CHƯƠNG 2: THUẬT TOÁN NLP 16](#_Toc173690363)

[1. Tổng quát về NLP 16](#_Toc173690364)

[2. Những lợi ích và thách thức liên quan đến việc triển khai NLP 17](#_Toc173690365)

[3. Những công việc phổ biến áp dụng NLP 20](#_Toc173690366)

[CHƯƠNG 3: CÁC THUẬT TOÁN LIÊN QUAN 22](#_Toc173690367)

[1. FastText thay vì Word2Vec 22](#_Toc173690368)

[2. Term Frequency-Inverse Document Frequency. 23](#_Toc173690369)

[3. Kmean Clustering. 24](#_Toc173690370)

[4. ROUGE Score. 25](#_Toc173690371)

[CHƯƠNG 4: TỔNG KẾT VÀ THỰC NGHIỆM CHƯƠNG TRÌNH 27](#_Toc173690372)

[1. Tiền xử lý văn bản. 27](#_Toc173690373)

[2. Tách các câu trong văn bản. 27](#_Toc173690374)

[3. Chuyển các câu sang vector. 28](#_Toc173690375)

[4. Xếp hạng câu để tìm ra các câu đại diện cho văn bản. 31](#_Toc173690376)

[5. Xây dựng đoạn văn bản tóm tắt. 32](#_Toc173690377)

[6. Đánh giá mô hình. 33](#_Toc173690378)

[7. Kết luận và hướng phát triển. 34](#_Toc173690379)

[KẾT LUẬN 36](#_Toc173690380)

[1. Kết quả nghiên cứu của đề tài 36](#_Toc173690381)

[2. Hạn chế đề tài 37](#_Toc173690382)

[3. Vấn đề kiến nghị và hướng đi tiếp theo của đề tài 37](#_Toc173690383)

[DANH MỤC TÀI LIỆU THAM KHẢO 39](#_Toc173690384)

# DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **VIẾT TẮT** | **TIẾNG ANH** | **TIẾNG VIỆT** |
| NLP | Natural Language Processing | Xử lý ngôn ngữ tự nhiên |
| TF-IDF | Term Frequency-Inverse Document Frequency | Tần suất xuất hiện - Tần suất nghịch đảo tài liệu |
| TM | Text Minig | Khai phá văn bản |
| TDM | Text Data Mining | Khai phá dữ liệu văn bản |
| KDT | Discovery in Textual Databases | Khai phá tri thức trong cơ sở dữ liệu văn bản |
| LSA | Latent Semantic Analysis | Phân tích ngữ nghĩa ẩn |
| LSI | Latent Semantic Indexing | Lập chỉ mục ngữ nghĩa ẩn |
| SVD | Singular Value Decomposition | Phân tích giá trị kỳ dị |
| IR | Information Retrieval | Truy hồi thông tin |
| CLIR | Cross-Language Information Retrieval | Truy hồi thông tin đa ngôn ngữ |

# DANH SÁCH BẢNG

[Bảng 1. Kết quả Rouge-Score sau khi train model 35](#_Toc173691401)

# DANH SÁCH HÌNH VẼ

[Hình 1. Mô hình học máy tóm tắt văn bản tiếng Việt 10](#_Toc173691448)

[Hình 2. Nghiên cứu của các tác giả trên tập dữ liệu VNDS 12](#_Toc173691449)

[Hình 3. Quá trình khai phá văn bản 13](#_Toc173691450)

[Hình 4. Lợi ích của NLP 18](#_Toc173691451)

[Hình 5. Những thách thức với NLP 20](#_Toc173691452)

[Hình 6. Những công việc phổ biến áp dụng NLP 21](#_Toc173691453)

[Hình 7. FastText & Word2Vec 23](#_Toc173691454)

[Hình 8. Công thức tính Term Frequency 24](#_Toc173691455)

[Hình 9. Công thức tính Inverse Document Frequency 24](#_Toc173691456)

[Hình 10. Công thức tính TF-IDF 25](#_Toc173691457)

[Hình 11. Công thức tính Recall - Precision 26](#_Toc173691458)

[Hình 12. Công thức tính Fmeasure Score 26](#_Toc173691459)

[Hình 13. Tiền xử lí dữ liệu - Đọc files 28](#_Toc173691460)

[Hình 14. Tiền xử lí dữ liệu – Kiểm tra files 28](#_Toc173691461)

[Hình 15. Tách các câu trong văn bản 29](#_Toc173691462)

[Hình 16. Chuyển các câu sang vector - Tokenize\_text 29](#_Toc173691463)

[Hình 17. Chuyển các câu sang vector - Tfidf\_weighted\_vector 30](#_Toc173691464)

[Hình 18. Chuyển các câu sang vector - process\_paragraph 31](#_Toc173691465)

[Hình 19. Chuyển các câu sang vector - convert\_sentence\_to\_vector 32](#_Toc173691466)

[Hình 20. Training model 33](#_Toc173691467)

[Hình 21. Xếp hạng câu để tìm ra các câu đại diện cho văn bản 33](#_Toc173691468)

[Hình 22. Xây dựng đoạn văn bản tóm tắt 34](#_Toc173691469)

[Hình 23. Đánh giá mô hình 34](#_Toc173691470)

[Hình 24. So sánh kết quả với các phương pháp của các tác giả khác trên cùng tập dữ liệu 37](#_Toc173691471)

# MỞ ĐẦU

## Tính cấp thiết của đề tài

Thời đại số hóa thông tin mà chúng ta đang sống được đặc trưng bởi sự tăng trưởng nhanh chóng của lượng dữ liệu và thông tin được thu thập, lưu trữ và cung cấp dưới định dạng điện tử. Đa số dữ liệu doanh nghiệp được lưu trữ trong các tài liệu văn bản mà hầu hết là không có cấu trúc. Theo một nghiên cứu của Merrill Lynch và Gartner, 85% tổng số dữ liệu doanh nghiệp được thu thập và lưu trữ dưới dạng không có cấu trúc (McKnight, 2005). Cùng một nghiên cứu cũng cho biết rằng dữ liệu không có cấu trúc này đang tăng gấp đôi kích thước của nó mỗi 18 tháng. Bởi vì tri thức là quyền lực trong thế giới kinh doanh ngày nay, và tri thức được tạo ra từ dữ liệu và thông tin, các doanh nghiệp có khả năng tận dụng hiệu quả nguồn dữ liệu văn bản của họ sẽ có tri thức cần thiết để đưa ra quyết định tốt hơn, dẫn đến lợi thế cạnh tranh so với những doanh nghiệp kém phát triển. Đây là nơi mà nhu cầu về khai thác văn bản (Text Mining) phù hợp với bức tranh tổng thể của doanh nghiệp ngày nay.

Việc tóm tắt văn bản tiếng Việt nhằm giải quyết vấn đề tràn ngập thông tin trong thời đại hiện nay, sử dụng các công nghệ máy học và trí tuệ nhân tạo để tiết kiệm thời gian và tăng hiệu suất trong việc nắm bắt thông tin, đồng thời áp dụng ứng dụng rộng rãi trong các lĩnh vực khác nhau.

## Tổng quan về vấn đề nghiên cứu

### Khái quát ngắn gọn tổng quan về vấn đề nghiên cứu

Việc tóm tắt một đoạn văn là trình bày các điểm chính của nó một cách ngắn gọn. Công việc tự động tóm tắt văn bản bắt đầu hơn 40 năm trước. Sự phát triển của Internet đã thúc đẩy công việc này trong những năm gần đây, và các hệ thống tóm tắt bắt đầu được áp dụng trong các lĩnh vực như chăm sóc sức khỏe và thư viện số. Hiện nay, có nhiều chương trình tóm tắt văn bản thương mại trên thị trường. Các ví dụ bao gồm: ViT5 large, ViT5 base, BARTpho, mBART, mT5 và Transformer, ...

Đề tài **"Phát triển mô hình text mining dựa trên kỹ thuật Machine Learning cho tóm tắt văn bản tiếng Việt"** nghiên cứu về việc tóm tắt văn bản một cách tự động. Tóm tắt giúp chuyên gia phân tích thông tin tiết kiệm thời gian và tập trung vào những nguồn dữ liệu quan trọng bằng cách xác định những tài liệu cần đọc hoặc xem xét trong các nguồn dữ liệu lớn. Các tóm tắt cũng giúp phát hiện các điểm tương đồng trong nội dung của các tài liệu, tạo cơ sở cho việc nhóm hoặc phân loại chúng thành các nhóm cụ thể. Quá trình này không chỉ giúp tối ưu hóa quản lý thông tin mà còn khám phá các mối quan hệ không ngờ đến giữa các tài liệu. Đề tài nghiên cứu này hứa hẹn đem lại những tiện ích đáng giá trong việc xử lý văn bản tiếng Việt.

### Khảo sát các công trình liên quan

*BARTpho: Pre-trained Sequence-to-Sequence Models for Vietnamese (Nguyen Luong Tran, Duong Minh Le, Dat Quoc Nguyen)*, bài báo giới thiệu BARTpho với hai phiên bản, BARTphosyllable và BARTphoword, đây là những mô hình Sequence-to-Sequence đơn ngôn ngữ quy mô lớn công khai đầu tiên được tiền huấn luyện cho tiếng Việt. BARTpho sử dụng kiến trúc "large" và phương pháp tiền huấn luyện của mô hình tự giám sát Sequence-to-Sequence BART, do đó nó đặc biệt thích hợp cho các nhiệm vụ sinh văn bản trong lĩnh vực xử lý ngôn ngữ tự nhiên. Bài báo đã tiến hành các thí nghiệm để so sánh BARTpho của các tác giả với đối thủ mBART trên nhiệm vụ tóm tắt văn bản tiếng Việt và cho thấy rằng: trong cả đánh giá tự động và đánh giá từ con người, BARTpho vượt trội so với mBART, một mô hình cơ sở mạnh mẽ, và cải thiện trạng thái nghệ thuật hiện đại.

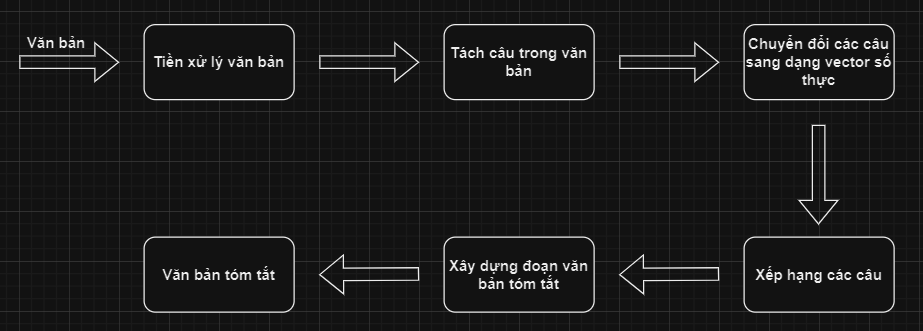
*ViT5: Pretrained Text-to-Text Transformer for Vietnamese Language Generation (Long Phan, Hieu Tran, Hieu Nguyen, Trieu H. Trinh),* các tác giả giới thiệu ViT5, một mô hình Transformer tiền huấn luyện Text-to-Text cho ngôn ngữ tiếng Việt. Sử dụng công thức tiền huấn luyện tự giám sát T5 trên các nguồn văn bản tiếng Việt lớn và chất lượng cao, nghiên cứu đã chỉ ra rằng ViT5 đạt được kết quả hàng đầu trên nhiệm vụ tóm tắt văn bản trên cả tập dữ liệu Wikilingua và Vietnews.

*Vietnamese doc summarization basic (Hoang Anh Pham),* tác giả đã giới thiệu một trong số những cách đơn giản nhất. Với việc áp dụng những phương pháp cơ bản nhất của học máy (Machine Learning) hay xử lý ngôn ngữ tự nhiên (Natural Language Processing). Tác giả thực hiện các bước: Preprocessing input document, Sentence tokenizer, Encode Sentences to vectors, Clustering, Build the summarization trên bộ dữ liệu neg.pkl – (5000 news and some papers in Vietnam) để xây dựng và hoàn thành chương trình tóm tắt văn bản (tiếng Việt).

**Nhận xét***:* Thông qua khảo sát các công trình liên quan, theo sự hiểu biết tốt nhất của học viên, chủ đề “Phát triển mô hình text mining dựa trên kỹ thuật Machine Learning cho tóm tắt văn bản tiếng Việt” đang là một chủ đề còn rất nhiều vấn đề mở chưa được giải quyết. Xây dựng một mô hình tóm tắt văn bản tiếng Việt tối ưu và chính xác hơn có thể mang lại nhiều lợi ích, bao gồm cải thiện hiểu biết về thông tin, tăng cường kỹ năng viết và đọc, và hỗ trợ nhanh chóng trong việc tiếp cận nội dung phức tạp trên internet và trong văn học.

### Mục đích nghiên cứu

Xây dựng mô hình học máy để tóm tắt văn bản tiếng Việt dựa trên kỹ thuật Text Mining.



**Hình 1. Mô hình học máy tóm tắt văn bản tiếng Việt**

## Đối tượng và phạm vi nghiên cứu

### Đối tượng nghiên cứu

Các đối tượng nghiên cứu để phát triển một mô hình sử dụng Text Mining và machine learning cho tóm tắt văn bản tiếng Việt:

- Text Mining.

- NLP.

- Word2Vec for Vietnamese.

- Term Frequency-Inverse Document Frequency.

- Kmean Clustering.

- ROUGE Score.

### Phạm vi nghiên cứu

Bộ dữ liệu**: VNDS (VNDS: A Vietnamese Dataset for Summarization)** (Van-Hau Nguyen, Thanh-Chinh Nguyen, Minh-Tien Nguyen, Nguyen Xuan Hoai)

Bộ dữ liệu này đã được công bố vào tháng 12 năm 2019 và có mã DOI là 10.1109/NICS48868.2019.9023886. Được giới thiệu tại Hội nghị Thứ sáu về Khoa học Thông tin và Máy tính của NAFOSTED năm 2019 (NICS). Bộ dữ liệu đã được chính thức công bố và chia sẻ trong cộng đồng nghiên cứu.

VNDS là một bộ dữ liệu tiếng Việt được thiết kế đặc biệt cho nhiệm vụ tóm tắt văn bản. Đây là một nguồn tài nguyên quan trọng giúp nghiên cứu và phát triển các hệ thống tóm tắt văn bản tự động trong tiếng Việt. Bộ dữ liệu này cung cấp các tài liệu nguồn đa dạng, bao gồm nhiều chủ đề khác nhau, để phản ánh sự đa dạng của ngôn ngữ và văn hóa tiếng Việt.

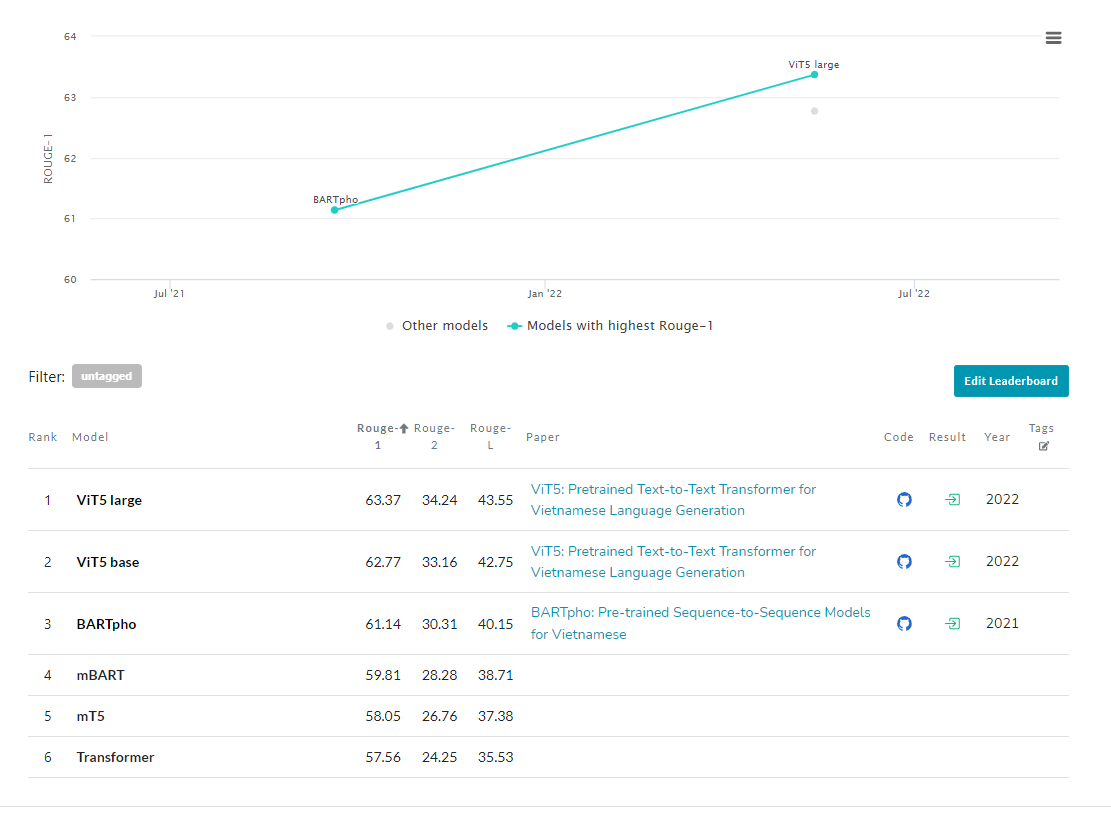
Các tài liệu trong VNDS được chú trọng để đảm bảo rằng chúng bao gồm thông tin chính và ý nghĩa tổng thể của văn bản nguồn. Bộ dữ liệu này cung cấp cơ hội cho các nhà nghiên cứu tiếng Việt nghiên cứu các phương pháp tóm tắt văn bản và đánh giá hiệu suất của chúng trên các tài liệu tiếng Việt thực tế.

Ngoài ra, VNDS không chỉ là một nguồn tài nguyên quan trọng cho việc phát triển và đánh giá các mô hình tóm tắt văn bản, mà còn giúp tạo ra một tiêu chuẩn trong lĩnh vực nghiên cứu tiếng Việt và thúc đẩy sự tiến bộ trong lĩnh vực này.

## Phương pháp nghiên cứu

Dựa vào các công trình nghiên cứu liên quan, học viên xây dựng mô hình tóm tắt văn bản tiếng Việt dự kiến như Hình 1 trên bộ dữ liệu **VNDS (VNDS: A Vietnamese Dataset for Summarization).** Học viên muốn tích hợp nhiều kỹ thuật và phương pháp khác nhau trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP) và học máy, để đạt được mục tiêu tóm tắt văn bản. Điều này cho thấy một hướng tiếp cận đa dạng và linh hoạt để giải quyết vấn đề tóm tắt văn bản tiếng Việt. Cách tiếp cận này cho phép học tận dụng các công cụ và kỹ thuật phù hợp nhất với đặc điểm cụ thể của dữ liệu tiếng Việt.

Sau khi thực hiện xây dựng thành công mô hình, học viên thực hiện đánh giá so sánh mô hình đã được xây dựng với những nghiên cứu của các tác giả khác như: ViT5 large, ViT5 base, BARTpho, mBART, mT5, Transformer… trên cùng bộ dữ liệu VNDS (*như hình …*). Từ đó, đưa ra nhận xét, hướng phát triển để cải thiện mô hình.



**Hình 2. Nghiên cứu của các tác giả trên tập dữ liệu VNDS**

## Bố cục đề án

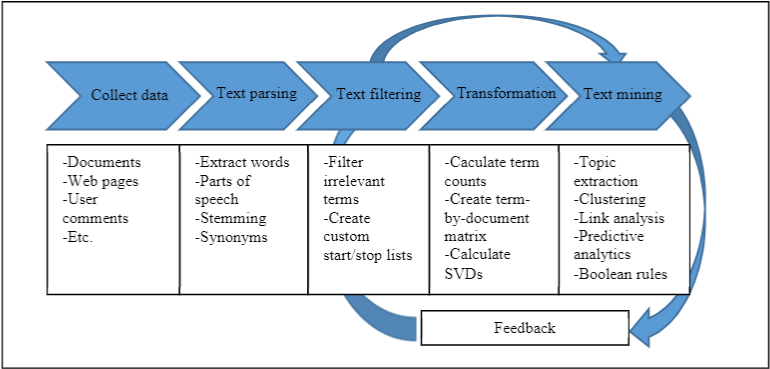
Bên cạnh phần mở đầu, phần kết luận và phần tài liệu tham khảo, phần nội dung chính của đề án được chia thành 5 chương chính như sau:

# CHƯƠNG 1: NGHIÊN CỨU TỔNG QUAN

## Tổng quan về Text Mining

Văn bản số ngày càng tăng và các kỹ thuật khai phá dữ liệu truyền thống không đủ để phân tích dữ liệu không cấu trúc. Xử lý văn bản (TM) trích xuất các chỉ số có ý nghĩa từ văn bản để sử dụng trong các thuật toán khai phá dữ liệu, cải thiện khả năng cạnh tranh của doanh nghiệp bằng cách phân tích dữ liệu khách hàng và đối thủ. Lợi ích của TM rõ ràng trong các lĩnh vực thu thập nhiều dữ liệu văn bản từ giao dịch kinh doanh, như theo dõi khiếu nại, yêu cầu bảo hành, và lỗi sản phẩm.

Text Mining là quá trình tìm kiếm hoặc trích xuất thông tin hữu ích từ dữ liệu văn bản. Đây là lĩnh vực nghiên cứu thú vị vì nó cố gắng khám phá kiến thức từ các văn bản không cấu trúc. Khai phá văn bản còn được gọi là Text Data Mining (TDM) và Discovery in Textual Databases (KDT). KDT đóng vai trò ngày càng quan trọng trong các ứng dụng mới nổi như Text Understanding.



**Hình 3. Quá trình khai phá văn bản**

Sau khi thu thập dữ liệu, bước tiếp theo là trích xuất, làm sạch và tạo từ điển từ các tài liệu bằng xử lý ngôn ngữ tự nhiên. Quá trình này bao gồm nhận diện câu, xác định thành phần câu, và gốc từ, loại bỏ từ dừng và kiểm tra chính tả. Văn bản được biến đổi thành biểu diễn số học bằng các phương pháp như LSA, LSI, và mô hình không gian vector, tạo ra ma trận thuật ngữ-tài liệu biểu diễn tần suất xuất hiện của thuật ngữ. Ma trận này có thể được giảm chiều bằng SVD và lọc các thuật ngữ hoặc tài liệu không liên quan. Sau đó, các thuật toán khai phá dữ liệu truyền thống như phân cụm, phân loại, phân tích kết hợp, và phân tích liên kết được áp dụng. Quá trình này lặp lại nhiều lần để cải thiện kết quả, dẫn đến các cụm tài liệu, danh sách chủ đề, hoặc quy tắc phân loại.

## Sự khác nhau giữa Text Mining và Data Mining

**Data Mining**

Khai thác dữ liệu là quá trình tìm kiếm các mẫu và trích xuất dữ liệu hữu ích từ các tập dữ liệu lớn. Nó được sử dụng để chuyển đổi dữ liệu thô thành dữ liệu hữu ích. Khai thác dữ liệu có thể cực kỳ hữu ích để cải thiện chiến lược tiếp thị của một công ty, vì với sự giúp đỡ của dữ liệu có cấu trúc, chúng ta có thể nghiên cứu dữ liệu từ các cơ sở dữ liệu khác nhau và sau đó có được những ý tưởng sáng tạo hơn để tăng năng suất của một tổ chức. Khai thác văn bản chỉ là một phần của khai thác dữ liệu.

Khai thác dữ liệu là quá trình khám phá các mẫu và kiến thức từ một lượng lớn dữ liệu. Nó liên quan đến việc sử dụng các kỹ thuật khác nhau như học máy, phân tích thống kê và quản lý cơ sở dữ liệu để trích xuất thông tin chi tiết và thông tin từ dữ liệu.

Khai thác dữ liệu có thể được áp dụng cho nhiều loại dữ liệu như dữ liệu số, dữ liệu phân loại và dữ liệu hình ảnh. Nó có thể được sử dụng cho các nhiệm vụ như dự đoán, phân loại, phân cụm và khai thác luật kết hợp. Khai thác dữ liệu có thể được áp dụng trong nhiều ngành công nghiệp như tài chính, chăm sóc sức khỏe, bán lẻ và sản xuất.

**Text Mining**

Khai thác văn bản về cơ bản là một công nghệ trí tuệ nhân tạo liên quan đến việc xử lý dữ liệu từ các tài liệu văn bản khác nhau. Nhiều thuật toán học sâu được sử dụng để đánh giá văn bản một cách hiệu quả. Trong khai thác văn bản, dữ liệu được lưu trữ ở dạng không có cấu trúc. Nó chủ yếu sử dụng các nguyên tắc ngôn ngữ học để đánh giá văn bản từ các tài liệu.

Khai thác văn bản, còn được gọi là khai thác dữ liệu văn bản, là một ứng dụng cụ thể của khai thác dữ liệu liên quan đến dữ liệu văn bản không có cấu trúc. Nó liên quan đến việc sử dụng các kỹ thuật xử lý ngôn ngữ tự nhiên (NLP) để trích xuất thông tin và thông tin chi tiết hữu ích từ một lượng lớn dữ liệu văn bản không có cấu trúc, chẳng hạn như tài liệu, email và bài đăng trên mạng xã hội. Khai thác văn bản có thể được sử dụng cho các nhiệm vụ như phân tích cảm xúc, nhận dạng thực thể có tên và mô hình hóa chủ đề.

Khai thác văn bản chủ yếu được sử dụng để trích xuất thông tin và thông tin chi tiết hữu ích từ dữ liệu văn bản không có cấu trúc. Điều này có thể bao gồm việc trích xuất các thực thể có tên, chẳng hạn như con người và tổ chức, từ một tài liệu hoặc xác định cảm xúc, chẳng hạn như tích cực hoặc tiêu cực, từ một bài đăng trên mạng xã hội. Khai thác văn bản có thể được sử dụng trong nhiều lĩnh vực như xử lý ngôn ngữ tự nhiên, truy xuất thông tin và phân tích mạng xã hội.

## Tại sao sự phổ biến của việc sử dụng Text Mining như một công cụ phân tích ngày càng tăng?

**Sự bùng nổ của dữ liệu phi cấu trúc**: Với sự phát triển của Internet và mạng xã hội, lượng dữ liệu phi cấu trúc (như email, bài viết trên mạng xã hội, bài báo, tài liệu văn bản) đã tăng lên nhanh chóng. Text Mining cung cấp các công cụ để khai thác và phân tích loại dữ liệu này, biến nó thành thông tin hữu ích.

**Cải thiện trải nghiệm khách hàng**: Bằng cách phân tích phản hồi của khách hàng từ các kênh khác nhau (như email, mạng xã hội, và khảo sát), các công ty có thể cải thiện dịch vụ và sản phẩm của mình để đáp ứng tốt hơn nhu cầu của khách hàng.

**Hỗ trợ ra quyết định**: Thông qua việc phân tích văn bản, các tổ chức có thể thu thập thông tin chi tiết và chính xác hơn để hỗ trợ quá trình ra quyết định. Điều này có thể áp dụng trong nhiều lĩnh vực như tài chính, y tế, marketing, và quản lý rủi ro.

**Tăng cường cạnh tranh**: Trong môi trường kinh doanh cạnh tranh khốc liệt, các công ty cần sử dụng mọi lợi thế có thể để dẫn đầu. Text Mining cung cấp cho họ công cụ để khai thác dữ liệu văn bản một cách hiệu quả, giúp họ cạnh tranh tốt hơn.

## Một số lĩnh vực ứng dụng phổ biến nhất.

Text Mining có thể được áp dụng trong nhiều lĩnh vực khác nhau, từ nghiên cứu học thuật, phát hiện gian lận, phân tích tình cảm, đến phân tích thị trường và dự báo xu hướng. Điều này làm cho nó trở thành một công cụ đa năng và có giá trị cao:

**Information Retrieval (IR)**: là việc liên kết và truy xuất thông tin từ một số lượng lớn tài liệu dựa trên văn bản. IR khác với hệ thống cơ sở dữ liệu ở chỗ nó không gặp phải một số vấn đề như kiểm soát đồng thời, khôi phục, quản lý giao dịch và cập nhật. Ngược lại, IR phải xử lý các tài liệu không có cấu trúc, tìm kiếm ước lượng dựa trên từ khóa và khái niệm liên quan. Do lượng thông tin văn bản khổng lồ, IR đã được áp dụng rộng rãi trong các hệ thống như thư viện trực tuyến, hệ thống quản lý tài liệu trực tuyến và các công cụ tìm kiếm trên web.

**Categorization**: Phân loại là quá trình xác định các chủ đề chính của một tài liệu bằng cách đưa tài liệu vào một tập hợp các chủ đề đã được xác định trước. Khi phân loại một tài liệu, chương trình máy tính thường xử lý tài liệu như một "túi từ" ("bag of words"). Nó không cố gắng xử lý thông tin thực sự như trích xuất thông tin. Thay vào đó, phân loại chỉ đếm số lần xuất hiện của các từ và từ đó, từ đó nhận diện các chủ đề chính mà tài liệu bao gồm. Phân loại thường phụ thuộc vào một từ điển mà các chủ đề đã được xác định trước, và các mối quan hệ được xác định bằng cách tìm kiếm các thuật ngữ lớn, thuật ngữ hẹp hơn, từ đồng nghĩa và các thuật ngữ liên quan.

**Natural Language Processing**: là một lĩnh vực nghiên cứu và ứng dụng nghiên cứu cách thức mà máy tính có thể được sử dụng để hiểu và xử lý văn bản ngôn ngữ tự nhiên. Các nhà nghiên cứu NLP nhằm thu thập kiến ​​thức về cách con người hiểu và sử dụng ngôn ngữ để phát triển các công cụ và kỹ thuật phù hợp, giúp hệ thống máy tính hiểu và xử lý các ngôn ngữ tự nhiên để thực hiện các nhiệm vụ mong muốn.

# CHƯƠNG 2: THUẬT TOÁN NLP

## Tổng quát về NLP

**Xử lý ngôn ngữ tự nhiên (NLP) là một lĩnh vực nghiên cứu và ứng dụng tìm hiểu cách máy tính có thể được sử dụng để hiểu và xử lý văn bản ngôn ngữ tự nhiên. Các nhà nghiên cứu NLP nhắm tới việc thu thập kiến thức về cách con người hiểu và sử dụng ngôn ngữ để phát triển các công cụ và kỹ thuật phù hợp, giúp các hệ thống máy tính có thể hiểu và xử lý các ngôn ngữ tự nhiên để thực hiện các nhiệm vụ mong muốn.**

**Các kiến thức cơ bản của NLP nằm trong nhiều lĩnh vực như khoa học máy tính và thông tin, ngôn ngữ học, toán học, kỹ thuật điện và điện tử, trí tuệ nhân tạo và robot, tâm lý học, v.v. Các ứng dụng của NLP bao gồm nhiều lĩnh vực nghiên cứu, chẳng hạn như dịch máy, xử lý và tóm tắt văn bản ngôn ngữ tự nhiên, giao diện người dùng, truy xuất thông tin đa ngôn ngữ và xuyên ngôn ngữ (CLIR), nhận dạng giọng nói, trí tuệ nhân tạo và hệ thống chuyên gia, v.v.**

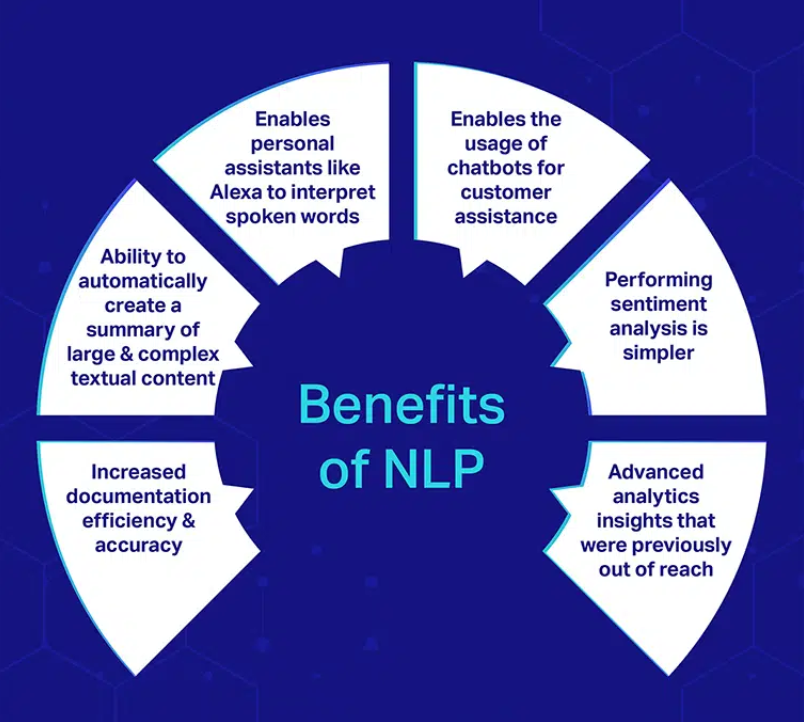
**Một số ứng dụng text mining ban đầu đã sử dụng mô hình đơn giản gọi là bag-of-words khi tạo cấu trúc cho bộ sưu tập các tài liệu văn bản để phân loại chúng vào hai hoặc nhiều lớp đã xác định trước hoặc để phân nhóm chúng vào các nhóm tự nhiên. Trong mô hình bag-of-words, văn bản, như câu, đoạn văn hoặc tài liệu hoàn chỉnh, được biểu diễn như một tập hợp các từ, không quan tâm đến ngữ pháp hoặc thứ tự xuất hiện của các từ. Mô hình bag-of-words vẫn được sử dụng trong một số công cụ phân loại tài liệu đơn giản. Ví dụ, trong lọc thư rác, một thư điện tử có thể được mô hình hóa như một tập hợp các từ không có thứ tự (bag-of-words) và được so sánh với hai túi từ đã xác định trước. Một túi chứa các từ tìm thấy trong tin nhắn rác và túi kia chứa các từ tìm thấy trong thư điện tử hợp pháp. Mức độ khớp giữa bag-of-words của một email cụ thể và hai túi từ chứa các từ mô tả xác định liệu email đó là thư rác hay hợp pháp.**

**Tuy nhiên, con người không sử dụng từ ngữ mà không có thứ tự hay cấu trúc. Chúng ta sử dụng từ trong các câu, có cấu trúc ngữ nghĩa cũng như cú pháp. Vì vậy, các kỹ thuật tự động (như text mining) cần tìm cách vượt qua cách diễn giải bag-of-words và tích hợp ngày càng nhiều cấu trúc ngữ nghĩa vào hoạt động của chúng. Xu hướng hiện nay trong text mining là hướng tới việc bao gồm nhiều tính năng nâng cao có thể đạt được bằng cách sử dụng xử lý ngôn ngữ tự nhiên (NLP)**

**Natural language processing (NLP) là một thành phần quan trọng của** text mining **và là một nhánh con của trí tuệ nhân tạo và ngôn ngữ học tính toán. Nó nghiên cứu vấn đề "hiểu" ngôn ngữ tự nhiên của con người, với mục tiêu chuyển đổi các mô tả của ngôn ngữ con người (như các tài liệu văn bản) thành các biểu diễn chính thức hơn (dưới dạng dữ liệu số và ký hiệu) dễ dàng cho các chương trình máy tính thao tác. Mục tiêu của NLP là vượt ra khỏi việc thao tác văn bản dựa trên cú pháp (thường được gọi là "word counting") để đạt đến một sự hiểu biết và xử lý thực sự của ngôn ngữ tự nhiên, xem xét cả các ràng buộc ngữ pháp và ngữ nghĩa cũng như ngữ cảnh.**

## Những lợi ích và thách thức liên quan đến việc triển khai NLP

**Lợi ích của NLP:**

****

**Hình 4. Lợi ích của NLP**

* **Tăng hiệu quả và độ chính xác của việc lập tài liệu: Một tài liệu được tạo bởi NLP có thể tóm tắt chính xác bất kỳ văn bản gốc nào mà con người không thể tự động tạo ra. Ngoài ra, nó có thể thực hiện các nhiệm vụ lặp đi lặp lại như phân tích các khối dữ liệu lớn để cải thiện hiệu quả của con người.**
* **Khả năng tự động tạo bản tóm tắt cho nội dung văn bản lớn và phức tạp:** Xử lý ngôn ngữ tự nhiên có thể được sử dụng cho các nhiệm vụ khai thác văn bản đơn giản như trích xuất thông tin từ tài liệu, phân tích cảm xúc, hoặc xác định các thực thể được đặt tên. Nó cũng có thể được sử dụng cho các nhiệm vụ phức tạp hơn, chẳng hạn như hiểu hành vi và cảm xúc của con người.
* **Cho phép các trợ lý cá nhân như Alexa hiểu được từ ngữ nói: NLP hữu ích cho các trợ lý cá nhân như Alexa, cho phép trợ lý ảo hiểu các lệnh bằng lời nói. Nó cũng giúp tìm kiếm nhanh chóng thông tin liên quan từ các cơ sở dữ liệu chứa hàng triệu tài liệu trong vài giây.**
* **Cho phép sử dụng chatbot để hỗ trợ khách hàng: NLP có thể được sử dụng trong các chatbot và các chương trình máy tính sử dụng trí tuệ nhân tạo để giao tiếp với con người qua văn bản hoặc giọng nói. Chatbot sử dụng NLP để hiểu những gì người dùng đang gõ và phản hồi một cách thích hợp. Chúng cũng cho phép một tổ chức cung cấp hỗ trợ khách hàng 24/7 trên nhiều kênh.**
* **Đơn giản hóa việc phân tích cảm xúc:** Phân tích cảm xúc là một quá trình phân tích một tập hợp các tài liệu (chẳng hạn như đánh giá hoặc tweet) liên quan đến thái độ hoặc trạng thái cảm xúc của chúng (ví dụ: vui mừng, tức giận). Phân tích cảm xúc có thể được sử dụng để phân loại và phân nhóm các bài đăng trên mạng xã hội hoặc các văn bản khác thành các danh mục như tích cực, tiêu cực hoặc trung lập.
* **Cung cấp thông tin phân tích nâng cao mà trước đây khó tiếp cận:** Sự phát triển gần đây của các cảm biến và thiết bị kết nối Internet đã dẫn đến sự bùng nổ về khối lượng và đa dạng của dữ liệu được tạo ra. Kết quả là nhiều tổ chức tận dụng NLP để hiểu dữ liệu của họ nhằm đưa ra các quyết định kinh doanh tốt hơn.

**Những thách thức với NLP**

****

**Hình 5. Những thách thức với NLP**

* **Lỗi chính tả: Ngôn ngữ tự nhiên đầy những lỗi chính tả, lỗi đánh máy, và không nhất quán trong phong cách. Ví dụ, từ "process" có thể được viết là "process" hoặc "processing." Vấn đề trở nên phức tạp hơn khi thêm các dấu hoặc các ký tự không có trong từ điển.**
* **Khác biệt ngôn ngữ: Một người nói tiếng Anh có thể nói, "I’m going to work tomorrow morning," trong khi một người nói tiếng Ý sẽ nói, "Domani Mattina vado al lavoro." Mặc dù hai câu này có nghĩa giống nhau, NLP sẽ không hiểu câu tiếng Ý trừ khi dịch nó sang tiếng Anh trước.**
* **Thiên vị bẩm sinh: Ngôn ngữ tự nhiên dựa trên logic và bộ dữ liệu của con người. Trong một số tình huống, hệ thống NLP có thể mang theo những thiên vị của người lập trình hoặc bộ dữ liệu mà chúng sử dụng. Nó cũng có thể đôi khi hiểu sai ngữ cảnh do những thiên vị bẩm sinh, dẫn đến kết quả không chính xác.**
* **Từ có nhiều nghĩa: NLP dựa trên giả định rằng ngôn ngữ là chính xác và không mơ hồ. Trong thực tế, ngôn ngữ không chính xác cũng không không mơ hồ. Nhiều từ có nhiều nghĩa và có thể được sử dụng theo nhiều cách khác nhau. Ví dụ, khi chúng ta nói "bark," nó có thể là tiếng sủa của chó hoặc vỏ cây.**
* **Sự không chắc chắn và kết quả dương tính giả: Kết quả dương tính giả xảy ra khi NLP phát hiện một thuật ngữ lẽ ra có thể hiểu được nhưng lại không thể phản hồi chính xác. Mục tiêu là tạo ra một hệ thống NLP có thể xác định được các giới hạn của nó và làm rõ sự nhầm lẫn bằng cách sử dụng câu hỏi hoặc gợi ý.**
* **Dữ liệu đào tạo: Một trong những thách thức lớn nhất với xử lý ngôn ngữ tự nhiên là dữ liệu đào tạo không chính xác. Càng có nhiều dữ liệu đào tạo, kết quả sẽ càng tốt. Nếu chúng ta cung cấp cho hệ thống dữ liệu không chính xác hoặc thiên vị, nó sẽ hoặc học những điều sai hoặc học không hiệu quả.**

## Những công việc phổ biến áp dụng NLP



**Hình 6. Những công việc phổ biến áp dụng NLP**

**Nhận diện giọng nói:** Điều này liên quan đến việc chuyển đổi dữ liệu giọng nói hoặc âm thanh thành văn bản. Quá trình này rất quan trọng đối với bất kỳ ứng dụng NLP nào có tính năng lệnh giọng nói. Nhận diện giọng nói giải quyết sự đa dạng trong phát âm, phương ngữ, tốc độ, âm lượng, giọng điệu và các yếu tố khác để giải mã thông điệp dự định.

**Gán nhãn từ loại:** Tương tự như cách chúng ta được dạy các nguyên tắc cơ bản về ngữ pháp ở trường, điều này dạy máy móc xác định các thành phần từ loại trong câu như danh từ, động từ, tính từ và nhiều hơn nữa. Nó cũng dạy hệ thống hiểu khi nào một từ được sử dụng như một động từ và khi từ đó được sử dụng như một danh từ.

**Phân biệt nghĩa của từ:** Đây là một quá trình quan trọng chịu trách nhiệm về việc hiểu đúng nghĩa của một câu. Sử dụng phân tích ngữ nghĩa trong nhiệm vụ này cho phép máy hiểu liệu một người đã nói "Mọi thứ đang diễn ra tốt đẹp" như một lời châm biếm khi đang trải qua khủng hoảng.

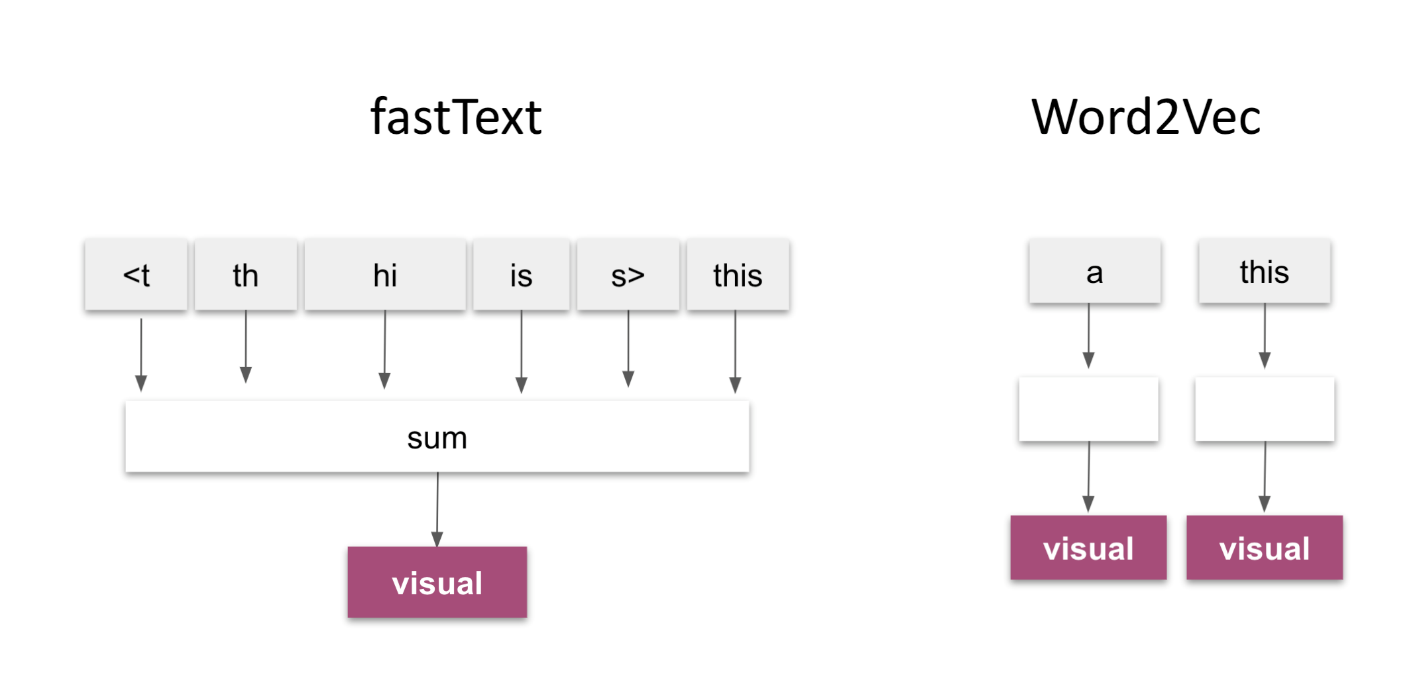
**Nhận dạng thực thể có tên:** Khi có nhiều trường hợp danh từ như tên, địa điểm, quốc gia, và nhiều hơn nữa, một quá trình gọi là Nhận dạng thực thể có tên được triển khai. Quá trình này xác định và phân loại các thực thể trong một thông điệp hoặc lệnh và gia tăng giá trị cho việc hiểu của máy.

**Giải quyết đồng tham chiếu:** Con người thường rất sáng tạo khi giao tiếp, vì vậy có rất nhiều ẩn dụ, so sánh, cụm động từ và thành ngữ. Tất cả các sự mơ hồ phát sinh từ những điều này được làm rõ bằng nhiệm vụ Giải quyết đồng tham chiếu, giúp máy học rằng nó không thực sự "mưa mèo và chó" mà là đề cập đến cường độ của mưa.

**Tạo ngôn ngữ tự nhiên:** Nhiệm vụ này liên quan đến việc tạo ra văn bản giống như con người từ dữ liệu. Văn bản này có thể được tùy chỉnh theo tiếng lóng, ngôn ngữ khu vực, và nhiều hơn nữa.

# CHƯƠNG 3: CÁC THUẬT TOÁN LIÊN QUAN

## FastText thay vì Word2Vec



**Hình 7. FastText & Word2Vec**

**Word2Vec**

* **Ưu điểm:**

Hiệu suất cao: Word2Vec có thể xử lý các tập dữ liệu lớn nhanh chóng và hiệu quả.

Biểu diễn ngữ nghĩa mạnh mẽ: Dễ dàng nắm bắt các mối quan hệ ngữ nghĩa giữa các từ thông dụng.

* **Nhược điểm:**

Không xử lý tốt các từ hiếm và từ mới: Do không xem xét các thành phần con của từ, Word2Vec gặp khó khăn với các từ ít gặp hoặc từ mới.

Hạn chế trong ngôn ngữ có cấu trúc phức tạp: Tiếng Việt có nhiều từ ghép và từ có dấu, khiến cho việc biểu diễn từ ngữ trở nên phức tạp.

**FastText**

* **Ưu điểm:**

Xử lý tốt các từ hiếm và từ mới: Nhờ vào việc sử dụng các subword (n-grams), fastText có thể nắm bắt được các từ ngữ mới và hiếm gặp.

Hiệu quả trong ngôn ngữ phức tạp: Có thể phân tích và biểu diễn tốt các từ ngữ có dấu và từ ghép trong tiếng Việt.

Biểu diễn ngữ nghĩa chính xác hơn: Các vector từ ngữ trong fastText thường có độ chính xác cao hơn trong nhiều tác vụ xử lý ngôn ngữ tự nhiên.

* **Nhược điểm:**

Tốc độ huấn luyện chậm hơn: Do phải xử lý các subword, fastText có tốc độ huấn luyện chậm hơn so với Word2Vec.

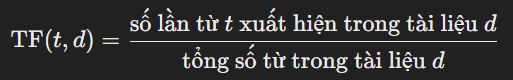
Độ phức tạp tính toán cao hơn: Việc tính toán các vector subword phức tạp hơn, yêu cầu tài nguyên tính toán nhiều hơn.

* Dựa trên các ưu điểm của fastText trong việc xử lý ngôn ngữ tiếng Việt, bao gồm khả năng nắm bắt các từ ngữ hiếm và mới, cũng như xử lý tốt các đặc điểm ngôn ngữ phức tạp như từ ghép và dấu, lựa chọn fastText là hợp lý và tối ưu cho việc biểu diễn từ ngữ trong các tác vụ xử lý ngôn ngữ tự nhiên tiếng Việt.

## Term Frequency-Inverse Document Frequency.

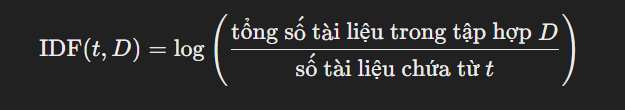
Term Frequency-Inverse Document Frequency (TF-IDF) là một phương pháp phổ biến để biểu diễn các từ ngữ dưới dạng vector trong tài liệu văn bản. Phương pháp này kết hợp hai yếu tố chính: tần suất xuất hiện của một từ trong một tài liệu (Term Frequency - TF) và tầm quan trọng của từ đó trên toàn bộ tập hợp các tài liệu (Inverse Document Frequency - IDF). Khi sử dụng TF-IDF để tạo vector cho câu, ta có thể xác định tầm quan trọng của từng từ trong câu đó dựa trên tần suất xuất hiện và mức độ hiếm gặp của từ trong toàn bộ tập dữ liệu.

* **Term Frequency (TF):** Đo lường tần suất xuất hiện của một từ trong một tài liệu.



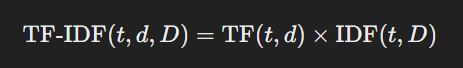
**Hình 8. Công thức tính Term Frequency**

* **Inverse Document Frequency (IDF):** Đo lường tầm quan trọng của một từ trên toàn bộ tập hợp các tài liệu.

****

**Hình 9. Công thức tính Inverse Document Frequency**

* **TF-IDF:** Kết hợp TF và IDF để đo lường tầm quan trọng của một từ trong một tài liệu cụ thể.

****

**Hình 10. Công thức tính TF-IDF**

* TF-IDF là một phương pháp mạnh mẽ để biểu diễn từ ngữ trong văn bản, cho phép xác định tầm quan trọng của từ ngữ dựa trên tần suất xuất hiện và mức độ hiếm gặp trong toàn bộ tập dữ liệu. Sử dụng TF-IDF để tạo vector cho câu giúp nắm bắt được thông tin ngữ nghĩa và tầm quan trọng của từ ngữ, hữu ích trong nhiều tác vụ xử lý ngôn ngữ tự nhiên như phân loại văn bản, tìm kiếm thông tin và phân cụm văn bản.

## Kmean Clustering.

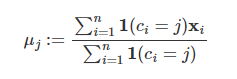
Trong thuật toán k-Means mỗi cụm dữ liệu được đặc trưng bởi một tâm (centroid). tâm là điểm đại diện nhất cho một cụm và có giá trị bằng trung bình của toàn bộ các quan sát nằm trong cụm. Chúng ta sẽ dựa vào khoảng cách từ mỗi quan sát tới các tâm để xác định nhãn cho chúng trùng thuộc về tâm gần nhất. Ban đầu thuật toán sẽ khởi tạo ngẫu nhiên một số lượng xác định trước tâm cụm. Sau đó tiến hành xác định nhãn cho từng điểm dữ liệu và tiếp tục cập nhật lại tâm cụm. Thuật toán sẽ dừng cho tới khi toàn bộ các điểm dữ liệu được phân về đúng cụm hoặc số lượt cập nhật tâm chạm ngưỡng.

Các bước của thuật toán k-Means được tóm tắt như sau:

1. Khởi tạo ngẫu nhiên 𝑘 tâm cụm 𝜇1,𝜇2,…,𝜇𝑘.
2. Lặp lại quá trình cập nhật tâm cụm cho tới khi dừng:
3. Xác định nhãn cho từng điểm dữ liệu 𝑐𝑖 dựa vào khoảng cách tới từng tâm cụm:



1. Tính toán lại tâm cho từng cụm theo trung bình của toàn bộ các điểm dữ liệu trong một cụm:



**So sánh với các phương pháp khác:**

*Latent Semantic Analysis (LSA):* Phương pháp này phân tích các mối quan hệ giữa các từ trong văn bản và có thể tạo ra tóm tắt chất lượng cao. Tuy nhiên, LSA có thể phức tạp hơn và đòi hỏi nhiều tài nguyên tính toán hơn so với K-means.

*LexRank và TextRank:* Đây là các phương pháp dựa trên đồ thị để tóm tắt văn bản, rất hiệu quả và không cần phải xác định số cụm trước. Tuy nhiên, các phương pháp này có thể trở nên phức tạp và khó triển khai hơn so với K-means.

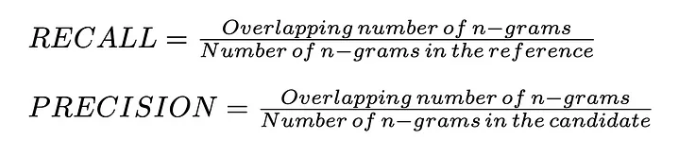
*Deep Learning:* Các mô hình tóm tắt văn bản dựa trên deep learning như BERTSUM hoặc T5 có thể tạo ra tóm tắt chất lượng rất cao nhưng đòi hỏi nhiều tài nguyên tính toán và dữ liệu đào tạo lớn. K-means là một giải pháp đơn giản và nhẹ nhàng hơn trong nhiều trường hợp.

* Sử dụng K-means trong tóm tắt văn bản tiếng Việt là một lựa chọn hợp lý vì một phương pháp phân cụm đơn giản, hiệu quả và có thể mở rộng. Nó đảm bảo bản tóm tắt bao quát được nhiều khía cạnh khác nhau của văn bản gốc và có thể được điều chỉnh linh hoạt cho nhiều loại văn bản khác nhau.

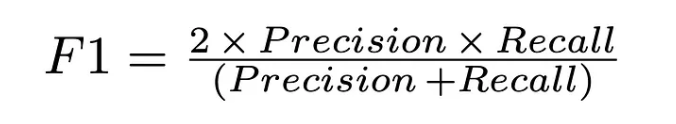
## ROUGE Score.

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) là một bộ các thước đo phổ biến để đánh giá chất lượng của hệ thống tóm tắt văn bản tự động. ROUGE đo lường sự trùng khớp giữa bản tóm tắt do máy tạo ra và bản tóm tắt tham chiếu do con người viết. ROUGE bao gồm các biến thể như ROUGE-N, ROUGE-L, và ROUGE-S. Dưới đây là cách tính toán và sử dụng các biến thể này:

**ROUGE-N:**



**Hình 11. Công thức tính Recall - Precision**



**Hình 12. Công thức tính Fmeasure Score**

**ROUGE-1:** Đo lường sự trùng khớp của các unigram (từ đơn lẻ) giữa bản tóm tắt máy và bản tóm tắt tham chiếu.

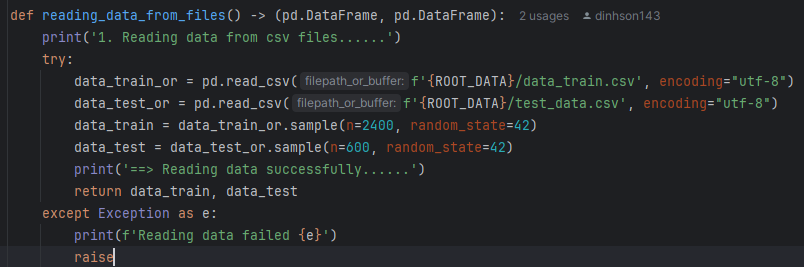
**ROUGE-2**: Đo lường sự trùng khớp của các bigram (cặp từ liên tiếp) giữa bản tóm tắt máy và bản tóm tắt tham chiếu.

* Việc sử dụng các thước đo ROUGE trong đánh giá model tóm tắt văn bản có những ưu điểm sau:
* Khả năng đánh giá tự động: ROUGE cho phép đánh giá chất lượng của các hệ thống tóm tắt một cách tự động và nhanh chóng, giảm bớt sự phụ thuộc vào đánh giá thủ công.
* Tính khách quan: Các thước đo ROUGE dựa trên sự trùng khớp của các n-gram hoặc chuỗi con chung, mang lại sự khách quan trong đánh giá.
* Khả năng so sánh: ROUGE cho phép so sánh hiệu suất giữa các hệ thống tóm tắt khác nhau một cách dễ dàng.
* Độc lập ngôn ngữ: ROUGE có thể được áp dụng cho nhiều ngôn ngữ khác nhau, bao gồm cả tiếng Việt.

# CHƯƠNG 4: TỔNG KẾT VÀ THỰC NGHIỆM CHƯƠNG TRÌNH

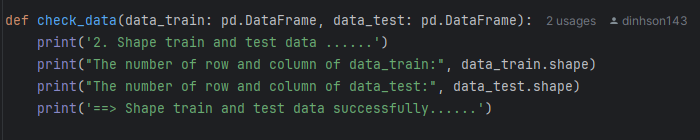
## Tiền xử lý văn bản.

* Đọc dữ liệu từ file CSV:



**Hình 13. Tiền xử lí dữ liệu - Đọc files**

* Hàm này có nhiệm vụ đọc dữ liệu từ các file CSV và trả về hai DataFrame: một chứa dữ liệu huấn luyện (data\_train) và một chứa dữ liệu kiểm tra (data\_test).
* Kiểm tra kích thước dữ liệu



**Hình 14. Tiền xử lí dữ liệu – Kiểm tra files**

* Hàm này kiểm tra và in ra kích thước (số dòng và cột) của dữ liệu huấn luyện và dữ liệu kiểm tra.

## Tách các câu trong văn bản.

* Tạo token từ các câu

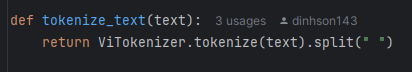


**Hình 15. Tách các câu trong văn bản**

* nltk.download('punkt'): Tải về gói tokenizer 'punkt' của NLTK.
* for i in data.index: Lặp qua từng dòng của DataFrame.
* data.loc[i, 'original'].lower().replace('\n', ' ').strip(): Chuẩn hóa văn bản bằng cách chuyển thành chữ thường, thay thế dấu xuống dòng bằng khoảng trắng và loại bỏ khoảng trắng thừa.
* nltk.sent\_tokenize: Tách văn bản thành các câu.
* re.sub(bieu\_thuc\_chinh\_quy, ' ', cau).strip(): Làm sạch câu bằng cách thay thế các ký tự đặc biệt bằng khoảng trắng và loại bỏ khoảng trắng thừa.
* paras.append(paras\_): Thêm các câu đã được làm sạch vào danh sách.
* Hàm này chuẩn hóa văn bản, tách các câu và tạo token từ các câu đã tách.

## Chuyển các câu sang vector.

* Hàm **tokenize\_text:**

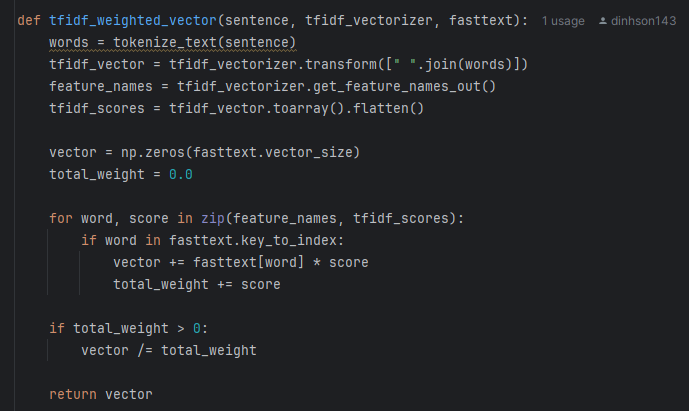


**Hình 16. Chuyển các câu sang vector - Tokenize\_text**

ViTokenizer.tokenize(text): Sử dụng ViTokenizer để tách câu tiếng Việt thành các từ.

.split(" "): Tách các từ đã được tokenize thành một danh sách các từ.

* Hàm này dùng để tách câu văn bản thành các từ (token).
* Hàm **tfidf\_weighted\_vector**

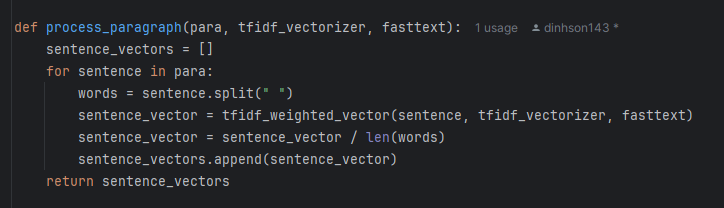


**Hình 17. Chuyển các câu sang vector - Tfidf\_weighted\_vector**

* tokenize\_text(sentence): Tokenize câu thành các từ.
* tfidf\_vectorizer.transform([" ".join(words)]): Tính toán vector TF-IDF cho câu.
* get\_feature\_names\_out(): Lấy danh sách các từ (features) trong TF-IDF.
* tfidf\_scores = tfidf\_vector.toarray().flatten(): Chuyển đổi vector TF-IDF thành mảng và làm phẳng nó.
* vector = np.zeros(fasttext.vector\_size): Khởi tạo vector kết quả với các giá trị 0.
* total\_weight = 0.0: Khởi tạo tổng trọng số TF-IDF.
* for word, score in zip(feature\_names, tfidf\_scores): Duyệt qua từng từ và trọng số của nó trong vector TF-IDF.
  + if word in fasttext.key\_to\_index: Kiểm tra nếu từ tồn tại trong từ điển FastText.

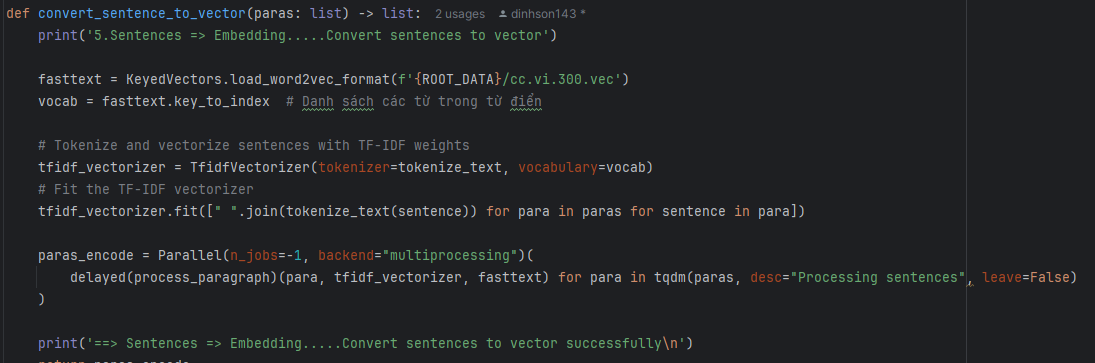
vector += fasttext[word] \* score: Cộng vector từ của từ với trọng số TF-IDF vào vector kết quả.

* + total\_weight += score: Cộng trọng số TF-IDF vào tổng trọng số.
* if total\_weight > 0: vector /= total\_weight: Chuẩn hóa vector bằng cách chia cho tổng trọng số nếu tổng trọng số lớn hơn 0.
* return vector: Trả về vector kết quả.
* Nhân vector từ của từ (lấy từ FastText) với trọng số TF-IDF của từ đó. Sau đó, cộng kết quả này vào vector kết quả. Việc này giúp biểu diễn từ trong ngữ cảnh của trọng số TF-IDF, nghĩa là từ quan trọng hơn (với trọng số TF-IDF cao hơn) sẽ có ảnh hưởng lớn hơn đến vector kết quả.
* Phương pháp này kết hợp TF-IDF (để xác định tầm quan trọng của từ trong ngữ cảnh cụ thể) và FastText (để có được vector từ chính xác) để tạo ra một biểu diễn vector mạnh mẽ và có ý nghĩa cho cả câu.
* Hàm **process\_paragraph**

****

**Hình 18. Chuyển các câu sang vector - process\_paragraph**

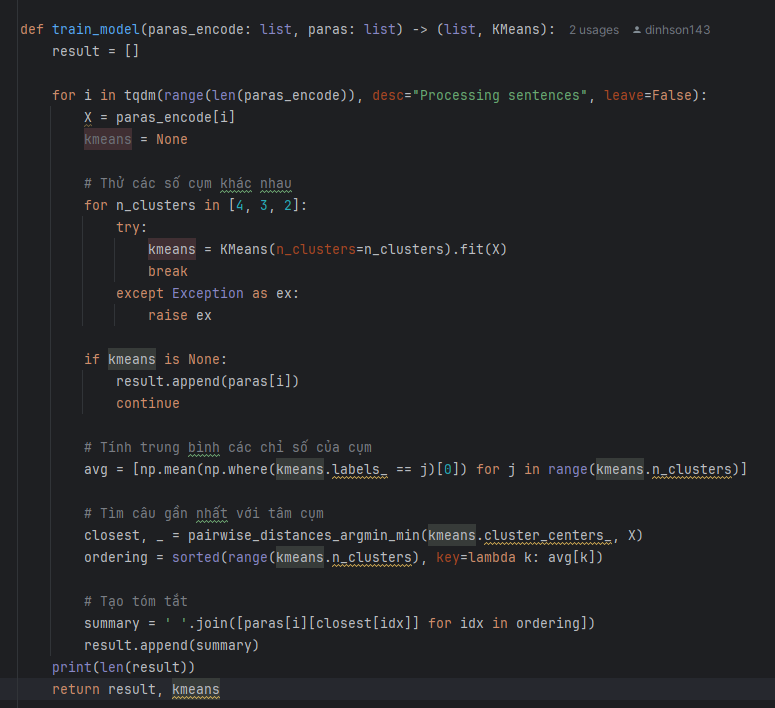
* sentence\_vectors = []: Khởi tạo danh sách để lưu các vector câu.
* for sentence in para: Duyệt qua từng câu trong đoạn văn bản.
  + words = sentence.split(" "): Tách câu thành các từ.
  + sentence\_vector = tfidf\_weighted\_vector(sentence, tfidf\_vectorizer, fasttext): Tính toán vector của câu bằng hàm tfidf\_weighted\_vector.
  + sentence\_vector = sentence\_vector / len(words): Chuẩn hóa vector câu bằng cách chia cho số lượng từ trong câu.
* return sentence\_vectors: Trả về danh sách các vector câu.
* Hàm này xử lý một đoạn văn bản, chuyển đổi các câu trong đoạn văn bản thành các vector.
* Hàm **convert\_sentence\_to\_vector**

****

**Hình 19. Chuyển các câu sang vector - convert\_sentence\_to\_vector**

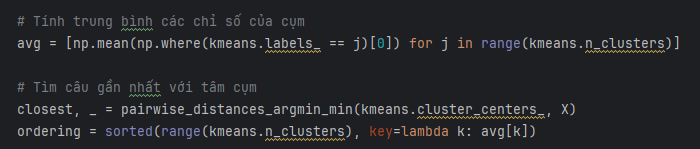
* fasttext=KeyedVectors.load\_word2vec\_format(f'{ROOT\_DATA}/cc.vi.300.vec'): Tải mô hình FastText từ file.
* vocab = fasttext.key\_to\_index: Lấy danh sách các từ trong từ điển FastText.
* tfidf\_vectorizer=TfidfVectorizer(tokenizer=tokenize\_text, vocabulary=vocab): Khởi tạo TF-IDF vectorizer với từ điển từ của FastText.
* tfidf\_vectorizer.fit([...]): Huấn luyện TF-IDF vectorizer với các câu đã được tokenize.
* paras\_encode=Parallel(...): Sử dụng multiprocessing để xử lý các đoạn văn bản song song, gọi hàm process\_paragraph cho mỗi đoạn văn bản.
* return paras\_encode: Trả về danh sách các đoạn văn bản đã được chuyển đổi thành vector.
* Hàm này chuyển đổi danh sách các đoạn văn bản thành các vector embedding bằng cách sử dụng TF-IDF và FastText.

## Xếp hạng câu để tìm ra các câu đại diện cho văn bản.



**Hình 20. Training model**

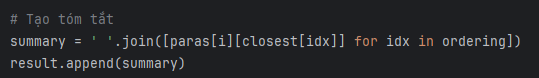
* Hàm này nhận đầu vào là danh sách các đoạn văn bản đã được mã hóa (paras\_encode) và danh sách các đoạn văn bản gốc (paras), sau đó trả về danh sách các câu đại diện cho mỗi đoạn văn bản và mô hình KMeans cuối cùng.



**Hình 21. Xếp hạng câu để tìm ra các câu đại diện cho văn bản**

* Tiếp theo tính chỉ số trung bình của các câu trong mỗi cụm.
* Sau đó tìm câu gần nhất với tâm cụm và sắp xếp các cụm theo thứ tự trung bình chỉ số.

## Xây dựng đoạn văn bản tóm tắt.



**Hình 22. Xây dựng đoạn văn bản tóm tắt**

* Tạo tóm tắt cho đoạn văn bản bằng cách ghép nối các câu đại diện của các cụm theo thứ tự.
* Hàm train\_model sử dụng phương pháp phân cụm KMeans để tìm ra các câu đại diện cho mỗi đoạn văn bản. Nó thử các số lượng cụm khác nhau, tính toán chỉ số trung bình của các cụm, và chọn các câu gần nhất với tâm cụm để tạo tóm tắt. Việc này giúp tạo ra các tóm tắt ngắn gọn và có ý nghĩa từ các đoạn văn bản dài.

## Đánh giá mô hình.



**Hình 23. Đánh giá mô hình**

* Tính toán điểm ROUGE cho dữ liệu huấn luyện:
* rouge\_1, rouge\_2, rouge\_L: Tạo DataFrame cho từng loại điểm ROUGE.
* scores=scorer.score(data.summary[i], lines\_train\_test[i + start\_idx]): Tính toán điểm ROUGE giữa tóm tắt của mô hình và tóm tắt tham chiếu.
* rouge\_1.append(list(scores['rouge1'][0:3])), rouge\_2.append(list(scores['rouge2'][0:3])), rouge\_L.append(list(scores['rougeL'][0:3])):Thêm điểm ROUGE vào các danh sách.
* for metric in ['precision', 'recall', 'fmeasure']: Duyệt qua các chỉ số precision, recall, và fmeasure để tính toán điểm ROUGE trung bình và in ra.
* Hàm calculate\_rouge tính toán các điểm ROUGE cho các câu trong dữ liệu huấn luyện và kiểm tra, và in ra kết quả trung bình cho từng loại điểm ROUGE.

## Kết luận và hướng phát triển.

|  |  |  |
| --- | --- | --- |
|  | Data train | Data test |
| precision score | Rouge\_1: 32.03590517623662  Rouge\_2: 16.295291882382653  Rouge\_L: 18.07967891795754 | Rouge\_1: 30.196701317606077  Rouge\_2: 16.777715767421411  Rouge\_L: 17.255594820293226 |
| recall score | Rouge\_1: 67.75772334994721  Rouge\_2: 24.323272066192915  Rouge\_L: 39.690100330098154 | Rouge\_1: 73.82706603196205  Rouge\_2: 28.058460515808502  Rouge\_L: 44.04807771533466 |
| fmeasure score | Rouge\_1: 41.41591770123297  Rouge\_2: 18.346944529568876  Rouge\_L: 23.626354486977633 | Rouge\_1: 41.4590548102516  Rouge\_2: 19.447386526571371  Rouge\_L: 23.93356175157701 |

**Bảng 1. Kết quả Rouge-Score sau khi train model**

* **Điểm ROUGE-1**

**Điểm ROUGE-1 đo lường sự tương đồng về từ vựng giữa tóm tắt của mô hình và tóm tắt tham chiếu.**

**Precision của mô hình rất gần nhau giữa dữ liệu huấn luyện và kiểm tra, cho thấy mô hình có khả năng duy trì độ chính xác tốt trên cả hai tập dữ liệu.**

**Recall của mô hình cũng tương tự giữa hai tập dữ liệu, cho thấy mô hình có khả năng bao phủ các thông tin quan trọng tốt cả trên dữ liệu huấn luyện và kiểm tra,** nghĩa là mô hình không bỏ sót nhiều nội dung quan trọng.

**F-measure cũng gần nhau giữa hai tập dữ liệu, cho thấy sự cân bằng tốt giữa độ chính xác và độ hồi tưởng.**

* **Điểm ROUGE-2**

**Điểm ROUGE-2 đo lường sự tương đồng về các cụm từ hai từ giữa tóm tắt của mô hình và tóm tắt tham chiếu.**

**Precision, Recall, và F-measure cho ROUGE-2 đều tương đối thấp, cho thấy mô hình có thể chưa tốt lắm trong việc nắm bắt các cụm từ hai từ.**

**Sự khác biệt giữa huấn luyện và kiểm tra là rất nhỏ, cho thấy mô hình có khả năng tổng quát tương đối tốt trên các tập dữ liệu.**

* **Điểm ROUGE-L**

**Điểm ROUGE-L đo lường sự tương đồng về chuỗi từ dài nhất giữa tóm tắt của mô hình và tóm tắt tham chiếu.**

**Precision, Recall, và F-measure cho ROUGE-L cho thấy mô hình có khả năng nắm bắt các chuỗi từ dài khá tốt.**

**Các điểm số rất gần nhau giữa huấn luyện và kiểm tra, cho thấy mô hình có khả năng tổng quát tốt trên các tập dữ liệu.**

* **Kết luận:**

**Mô hình đã chứng minh được khả năng tóm tắt văn bản tiếng Việt một cách hiệu quả và đồng nhất giữa dữ liệu huấn luyện và kiểm tra. Các điểm số ROUGE đạt được cho thấy mô hình có khả năng tổng quát tốt (khả năng duy trì hiệu suất trên dữ liệu kiểm tra).**

**Mô hình không chỉ hoạt động tốt trên dữ liệu huấn luyện mà còn duy trì hiệu suất tương tự trên dữ liệu kiểm tra, cho thấy mô hình có thể áp dụng cho các tập dữ liệu mới mà không bị giảm hiệu suất đáng kể.**

**Điểm ROUGE-1 cao hơn so với ROUGE-2 và ROUGE-L, điều này cho thấy mô hình có thể nắm bắt các từ đơn tốt hơn là các cụm từ và chuỗi từ dài hơn.**

* **Hướng phát triển:**

**Xem xét các kỹ thuật khác hoặc tinh chỉnh mô hình để cải thiện điểm ROUGE-2 và ROUGE-L, để nắm bắt tốt hơn các cụm từ và chuỗi từ dài hơn.**

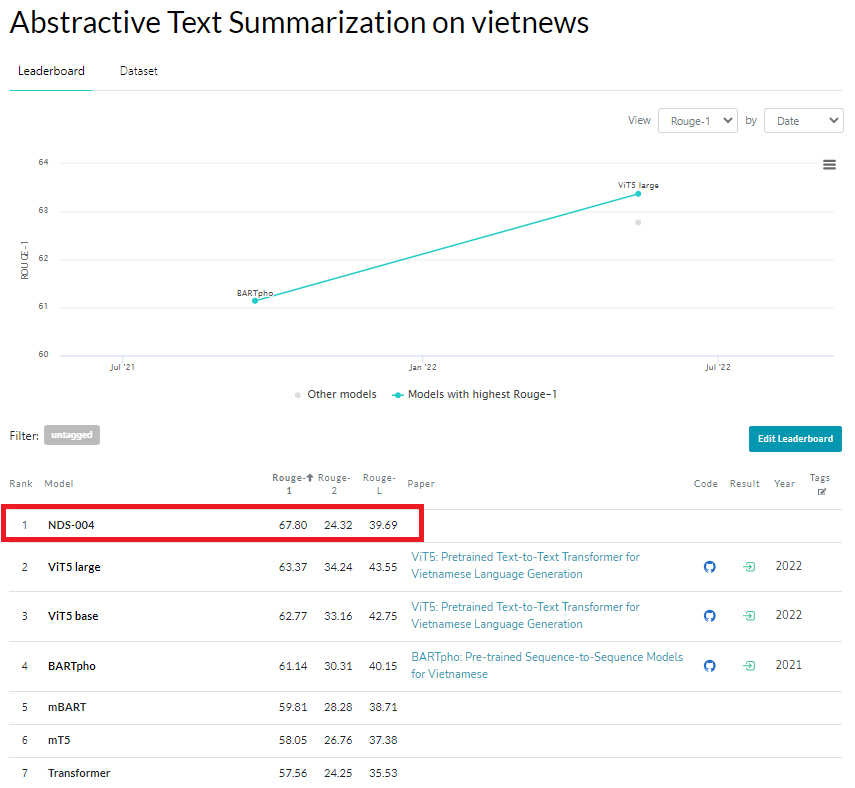
**Cần cải thiện thêm các kỹ thuật xử lý ngôn ngữ và mô hình học máy để nâng cao điểm số ROUGE-2. Nghiên cứu và thử nghiệm các phương pháp biểu diễn văn bản và phân cụm khác để cải thiện chất lượng tóm tắt. Sử dụng thêm các nguồn dữ liệu khác nhau để đánh giá khả năng áp dụng của mô hình trên nhiều lĩnh vực khác nhau.**

# KẾT LUẬN

## Kết quả nghiên cứu của đề tài

Đề án này đề xuất một phương pháp tóm tắt văn bản tiếng Việt sử dụng các kỹ thuật xử lý ngôn ngữ tự nhiên (NLP) và học máy.

So sánh với các phương pháp của các tác giả khác trên cùng tập dữ liệu:



**Hình 24. So sánh kết quả với các phương pháp của các tác giả khác trên cùng tập dữ liệu**

**Hiệu suất tổng thể:** Phương pháp của tôi (**NDS-004**) có điểm **Rouge-1** và **Rouge-L** cao hơn so với các mô hình hiện đại như ViT5 large và BARTpho. Điều này cho thấy mô hình của tôi có khả năng tóm tắt văn bản tiếng Việt rất tốt.

* **Rouge-1** của phương pháp của tôi (67.8) cao hơn so với tất cả các mô hình khác, cho thấy mô hình của tôi có khả năng nắm bắt thông tin từ văn bản gốc rất tốt.
* **Rouge-2** của phương pháp của tôi (24.32) còn thấp, cho thấy mô hình chưa nắm bắt tốt mối quan hệ giữa các từ trong câu.
* **Rouge-L** của phương pháp của tôi (39.69) khá cao, cho thấy mô hình của tôi có khả năng giữ được cấu trúc dài hạn và độ mạch lạc của văn bản tốt.
* **Phương pháp của tôi** đã cho thấy hiệu suất rất tốt, vượt trội hơn so với các mô hình hiện đại như ViT5 và BARTpho ở một số chỉ số. Điều này chứng tỏ rằng phương pháp của tôi có tiềm năng lớn trong việc tóm tắt văn bản tiếng Việt.

## Hạn chế đề tài

**Chênh lệch giữa các chỉ số ROUGE:**

* Mặc dù kết quả Rouge-1 rất cao (67.8), nhưng Rouge-2 chỉ đạt 24.32. Điều này cho thấy mô hình có khả năng nhận diện từ và cụm từ đơn lẻ tốt, nhưng gặp khó khăn hơn trong việc nắm bắt mối quan hệ giữa các từ trong câu.

**Tính tổng quát của mô hình**:

* Hiệu suất mô hình trên tập dữ liệu huấn luyện và kiểm tra là tốt, nhưng có thể không đảm bảo rằng mô hình sẽ hoạt động hiệu quả trên các tập dữ liệu chưa từng thấy hoặc trong các ngữ cảnh thực tế.

**Thời gian build mô hình**:

* Thời gian xây dựng mô hình quá lâu trên tập dữ liệu lớn, dẫn đến việc thuê server mất nhiều chi phí. Điều này làm cho quá trình nghiên cứu và triển khai trở nên tốn kém và kém hiệu quả về mặt thời gian.

## Vấn đề kiến nghị và hướng đi tiếp theo của đề tài

Đối với các nghiên cứu trong tương lai, đề án có thể cải tiến bằng cách xem xét các vấn đề sau:

* Tiếp tục tối ưu hóa mô hình hiện tại và thử nghiệm trên các bộ dữ liệu khác nhau để đánh giá khả năng tổng quát của mô hình.
* Sử dụng các kỹ thuật tiên tiến hơn như attention mechanisms hoặc transformer architectures để nắm bắt tốt hơn mối quan hệ giữa các từ trong câu.
* Xem xét các kỹ thuật mới trong lĩnh vực xử lý ngôn ngữ tự nhiên để tiếp tục nâng cao chất lượng tóm tắt.
* Nghiên cứu và áp dụng các kỹ thuật giảm thiểu độ phức tạp của mô hình như pruning, quantization hoặc sử dụng các mô hình nhỏ gọn hơn mà vẫn duy trì hiệu suất tốt.

# DANH MỤC TÀI LIỆU THAM KHẢO

1. Hoang Anh Pham. (2018). Vietnamese doc summarization basic. Available from: <https://github.com/hoanganhpham1006/Vietnamese_doc_summarization_basic>
2. Xin Rong, 2016. word2vec Parameter Learning Explained - In arXiv:1411.2738v4 [cs.CL]
3. Facebook AI Research. fastText Vietnamese word vectors full. Available from: <https://www.kaggle.com/datasets/aeryss/fasttext-vietnamese-word-vectors-full>
4. Phạm Hữu Quang. (2018). Xây dựng mô hình không gian vector cho Tiếng Việt. Available from:

<https://github.com/hoanganhpham1006/Vietnamese_doc_summarization_basic>

1. Nguyen, Van-Hau & Nguyen, Thanh-Chinh & Nguyen, Minh-Tien & Hoai, Nguyen. (2019). VNDS: A Vietnamese Dataset for Summarization. 375-380. 10.1109/NICS48868.2019.9023886.
2. Nguyen, Dat Quoc & Nguyen, Anh. (2020). PhoBERT: Pre-trained language models for Vietnamese. 1037-1042. 10.18653/v1/2020.findings-emnlp.92.
3. H. Q. To, and K. V. Nguyen, and N. L. -T. Nguyen, and A. G. T. Nguyen, Monolingual vs multilingual BERTology for Vietnamese extractive multi-document summarization, in Proceedings of the 35th Pacific Asia Conference on Language, Information and Computation, Shanghai: China, pp. 692-699, 2021.
4. Hieu Nguyen, Long Phan, James Anibal, Alec Peltekian, and Hieu Tran. 2021. Viesum: How robust are transformer-based models on vietnamese summarization?
5. T. A. Nguyen-Hoang and K. Nguyen and Q. V. Tran, TSGVi: A graphbased summarization system for Vietnamese documents, Journal of Ambient Intelligence and Humanized Computing, vol. 3, no. 4, pp. 305-313, Jun., 2012. DOI: 10.1007/s12652-012-0143-x.
6. D. Q. Nguyen and A. T. Nguyen, “PhoBERT: Pre-trained language models for Vietnamese,” in Findings of EMNLP, 2020.
7. [13]. D. Q. Thang, L. H. Phuong, N. T. M. Huyen, N. C. Tu, M. Rossignol, and V. X. Luong, “Word segmentation of Vietnamese texts: a comparison of approaches,” in LREC, 2008.
8. Minh-Tien Nguyen, Hoang-Diep Nguyen, Thi-HaiNang Nguyen, and Van-Hau Nguyen. 2018. Towards state-of-the-art baselines for vietnamese multi-document summarization. In 2018 10th International Conference on Knowledge and Systems Engineering (KSE), pages 85–90.
9. Rothe, Sascha & Narayan, Shashi & Severyn, Aliaksei. (2020). Leveraging Pre-trained Checkpoints for Sequence Generation Tasks. Transactions of the Association for Computational Linguistics. 8. 264-280. 10.1162/tacl\_a\_00313.
10. M. Allahyari, and S. Pouriyeh, and M. Assefi, and S. Safaei, and E. D. Trippe, and J. B. Gutierrez, and K. Kochut, Text summarization techniques: A brief survey, International Journal of Advanced Computer Science and Applications (IJACSA), vol. 8, no. 10, pp. 397-405, 2017. DOI: 10.14569/IJACSA.2017.081052.
11. L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel, “mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer,” in NAACL, 2021.
12. T. V. Bui, T. O. Tran, and P. Le-Hong, “Improving Sequence Tagging for Vietnamese Text using Transformer-based Neural Models,” in PACLIC, 2020.
13. T. Vu, D. Q. Nguyen, D. Q. Nguyen, M. Dras, and M. Johnson, “VnCoreNLP: A Vietnamese Natural Language Processing Toolkit,” in NAACL (Demonstrations), 2018.
14. M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B.Gutierrez, and K. Kochut, “*Text summarization techniques: A briefsurvey,” International Journal of Advanced Computer Science andApplications (IJACSA),* vol. 8, no. 10, pp. 397-405, 2017. DOI:10.14569/IJACSA.2017.081052. (PDF) Text Summarization on Large-scale Vietnamese Datasets. Available from: https://www.researchgate.net/publication/366806176\_Text\_Summarization\_on\_Large-scale\_Vietnamese\_Datasets [accessed Nov 03 2023].
15. A. Nguyen-Hoang, K. Nguyen, and Q. V. Tran, “TSGVi: A graph-based summarization system for Vietnamese documents,” *Journal ofAmbient Intelligence and Humanized Computing*, vol. 3, no. 4, pp.305-313, Jun. 2012. DOI: 10.1007/s12652-012-0143-x. (PDF) Text Summarization on Large-scale Vietnamese Datasets. Available from: https://www.researchgate.net/publication/366806176\_Text\_Summarization\_on\_Large-scale\_Vietnamese\_Datasets [accessed Nov 03 2023].
16. Dr Inderjeet Mani , Mark T Maybury. (2018). Advances in Automatic Text Summarization (Mit Press).
17. Hong Phuong L., Thi Minh Huyen N., Roussanaly A., Vinh H.T. (2008) A Hybrid Approach to Word Segmentation of Vietnamese Texts. In: Martín-Vide C., Otto F., Fernau H. (eds) Language and Automata Theory and Applications. LATA 2008. Lecture Notes in Computer Science, vol 5196. Springer, Berlin, Heidelberg.
18. Nguyễn Thị Ngọc Tú, Nguyễn Thị Thu Hà, Lê Thanh Hương, Hồ Ngọc Vinh, Đào Thanh Tĩnh, Nguyễn Ngọc Cương. “Ứng dụng mô hình đồ thị trong tóm tắt đa văn bản tiếng Việt”. Kỷ yếu Hội nghị Quốc gia lần thứ VIII về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR); Hà Nội, ngày 9-10/7/2015.
19. Sam Wiseman and Alexander M. Rush, 2016. Sequence-to-Sequence Learning as Beam-Search Optimization. School of Engineering and Applied Sciences Harvard University Cambridge, MA, USA-In arXiv:1606.02960v2 [cs.CL].
20. Sam Wiseman and Alexander M. Rush, 2016. Sequence-to-Sequence Learning as Beam-Search Optimization. School of Engineering and Applied Sciences Harvard University Cambridge, MA, USA-In arXiv:1606.02960v2 [cs.CL].
21. Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean, 2013. Efficient Estimation of Word Representations in Vector Space – In arXiv:1301.3781v3 [cs.CL].
22. Trương Quốc Định, Nguyễn Quang Dũng. “Một giải pháp tóm tắt văn bản tiếng Việt tự động”. Hội thảo quốc gia lần thứ XV: một số vấn đề chọn lọc của Công nghệ thông tin và Truyền thông Hà Nội 03-04/12/2012.