

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

**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 … tháng … 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 … tháng … năm 2024

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

# MỤC LỤC

[LỜI CAM ĐOAN 1](#_Toc174886878)

[LỜI CẢM ƠN 2](#_Toc174886879)

[DANH MỤC CÁC THUẬT NGỮ, CHỮ VIẾT TẮT 4](#_Toc174886880)

[DANH SÁCH BẢNG 5](#_Toc174886881)

[DANH SÁCH HÌNH VẼ 6](#_Toc174886882)

[MỞ ĐẦU 1](#_Toc174886883)

[1. Tính cấp thiết của đề tài 1](#_Toc174886884)

[2. Tổng quan về vấn đề nghiên cứu 1](#_Toc174886885)

[2.1. Khái quát ngắn gọn tổng quan về vấn đề nghiên cứu 1](#_Toc174886886)

[2.2. Khảo sát các công trình liên quan 2](#_Toc174886887)

[2.3. Mục đích nghiên cứu 4](#_Toc174886888)

[3. Đối tượng và phạm vi nghiên cứu 4](#_Toc174886889)

[3.1 Đối tượng nghiên cứu 4](#_Toc174886890)

[3.2 Phạm vi nghiên cứu 4](#_Toc174886891)

[4. Phương pháp nghiên cứu 5](#_Toc174886892)

[5. Bố cục đề án 6](#_Toc174886893)

[CHƯƠNG 1: NGHIÊN CỨU TỔNG QUAN 7](#_Toc174886894)

[1.1. Tổng quan về Text Mining 7](#_Toc174886895)

[1.1.1 Text Mining là gì? 7](#_Toc174886896)

[1.1.2 Text mining và Text analytics 8](#_Toc174886897)

[1.1.3 Các kỹ thuật khai thác văn bản 9](#_Toc174886898)

[1.1.4 Ứng dụng của khai thác văn bản 12](#_Toc174886899)

[1.2. Tổng quan về NLP 13](#_Toc174886900)

[1.2.1 NLP là gì? 14](#_Toc174886901)

[1.2.2 Lợi ích của NLP 15](#_Toc174886902)

[1.2.3 Những thách thức với NLP 15](#_Toc174886903)

[1.2.4 Cách hoạt động của NLP 16](#_Toc174886904)

[1.2.5 Các tác vụ NLP 19](#_Toc174886905)

[1.2.6 Các trường hợp sử dụng NLP 20](#_Toc174886906)

[CHƯƠNG 2: CÁC THUẬT TOÁN LIÊN QUAN 23](#_Toc174886907)

[2.1 Các bước xử lý Text Mining 23](#_Toc174886908)

[2.1.1 Thu thập dữ liệu 23](#_Toc174886909)

[2.1.2 Tiền xử lý dữ liệu 24](#_Toc174886910)

[2.1.3 Biểu diễn văn bản 24](#_Toc174886911)

[2.1.4 Phân tích và khai phá dữ liệu 24](#_Toc174886912)

[2.1.5 Đánh giá kết quả 24](#_Toc174886913)

[2.2 Thuật toán và độ đo 25](#_Toc174886914)

[2.2.1 FastText 25](#_Toc174886915)

[2.2.2 Kmean Clustering. 26](#_Toc174886916)

[2.2.3 Distance Measurement. 28](#_Toc174886917)

[2.2.4 Term Frequency-Inverse Document Frequency. 28](#_Toc174886918)

[2.2.5 ROUGE Score. 29](#_Toc174886919)

[2.3 Giải pháp đề xuất tóm tắt văn bản 30](#_Toc174886920)

[2.3.1 Mô hình tóm tắt văn bản đề xuất 30](#_Toc174886921)

[2.3.2 Tập dữ liệu 32](#_Toc174886922)

[2.3.3 Phương pháp đánh giá 32](#_Toc174886923)

[2.3.4 Cấu trúc thí nghiệm 33](#_Toc174886924)

[CHƯƠNG 3: TỔNG KẾT VÀ THỰC NGHIỆM CHƯƠNG TRÌNH 34](#_Toc174886925)

[3.1 Thực nghiệm chương trình. 34](#_Toc174886926)

[3.1.1 Đọc dữ liệu 34](#_Toc174886927)

[3.1.2 Kiểm tra dữ liệu 34](#_Toc174886928)

[3.1.3 Tạo token từ câu 34](#_Toc174886929)

[3.1.4 Chuyển đổi câu thành vector 34](#_Toc174886930)

[3.1.5 Huấn luyện mô hình Kmeans 35](#_Toc174886931)

[3.1.6 Lưu mô hình 35](#_Toc174886932)

[3.2 Kết luận và hướng phát triển. 35](#_Toc174886933)

[KẾT LUẬN 42](#_Toc174886934)

[1. Kết quả nghiên cứu của đề tài 42](#_Toc174886935)

[2. Hạn chế đề tài 42](#_Toc174886936)

[3. Vấn đề kiến nghị và hướng đi tiếp theo của đề tài 42](#_Toc174886937)

[DANH MỤC TÀI LIỆU THAM KHẢO 44](#_Toc174886938)

[PHỤ LỤC 49](#_Toc174886939)

[Phụ lục 1: Mã Code Mô Hình Training 49](#_Toc174886940)

[Tiền xử lý văn bản. 49](#_Toc174886941)

[Tách các câu trong văn bản. 49](#_Toc174886942)

[Chuyển các câu sang vector. 50](#_Toc174886943)

[Xếp hạng câu để tìm ra các câu đại diện cho văn bản. 54](#_Toc174886944)

[Xây dựng đoạn văn bản tóm tắt. 55](#_Toc174886945)

[Đánh giá mô hình. 55](#_Toc174886946)

# 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 3. 1: Kết quả Rouge-Score sau khi train model 36](#_Toc174886656)

# DANH SÁCH HÌNH VẼ

[Hình 1: Kết quả nghiên cứu của các mô hình trên tập dữ liệu VNDS 6](#_Toc174886699)

[Hình 2: Tiền xử lí dữ liệu - Đọc files 49](#_Toc174886700)

[Hình 3: Tiền xử lí dữ liệu – Kiểm tra files 49](#_Toc174886701)

[Hình 4: Tách các câu trong văn bản 50](#_Toc174886702)

[Hình 5: Chuyển các câu sang vector - Tokenize\_text 51](#_Toc174886703)

[Hình 6: Chuyển các câu sang vector - Tfidf\_weighted\_vector 51](#_Toc174886704)

[Hình 7: Chuyển các câu sang vector - process\_paragraph 52](#_Toc174886705)

[Hình 8: Chuyển các câu sang vector - convert\_sentence\_to\_vector 53](#_Toc174886706)

[Hình 9: Training model 54](#_Toc174886707)

[Hình 10: Xếp hạng câu để tìm ra các câu đại diện cho văn bản 55](#_Toc174886708)

[Hình 11: Xây dựng đoạn văn bản tóm tắt 55](#_Toc174886709)

[Hình 12: Đánh giá mô hình 56](#_Toc174886710)

[Hình 1. 1: Tổng quan về Text Mining 7](#_Toc174886755)

[Hình 1. 2: Tổng quan về NLP 14](#_Toc174886756)

[Hình 2. 1: Các bước xử lý Text Mining 23](#_Toc174886803)

[Hình 2. 2: SkipGram với thông tin subword (character n-gram size=2) 25](#_Toc174886804)

[Hình 2. 3: Mô hình tóm tắt văn bản đề xuất 31](#_Toc174886805)

[Hình 3. 1: So sánh kết quả của mô hình đề xuất với các phương pháp của các tác giả khác trên cùng tập dữ liệu 37](#_Toc174886810)

# 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 nhằm:

* **Tăng cường khả năng xử lý thông tin trong bối cảnh dữ liệu lớn**:
* Với sự bùng nổ của dữ liệu số trong thời đại hiện nay, việc xử lý và quản lý thông tin trở thành một thách thức lớn. Tóm tắt văn bản tự động là một công cụ mạnh mẽ giúp người dùng nhanh chóng nắm bắt được nội dung chính của một tài liệu hoặc tập hợp tài liệu lớn mà không cần phải đọc toàn bộ. Điều này đặc biệt quan trọng trong các lĩnh vực như báo chí, nghiên cứu khoa học, và quản lý tài liệu, nơi mà khối lượng thông tin cần xử lý là rất lớn.
* **Phát triển và ứng dụng của trí tuệ nhân tạo trong ngôn ngữ tự nhiên:**
* Ngôn ngữ tự nhiên (Natural Language Processing - NLP) là một trong những lĩnh vực quan trọng của trí tuệ nhân tạo (AI), với nhiều ứng dụng trong thực tế như dịch máy, tìm kiếm thông tin, phân tích cảm xúc, và tóm tắt văn bản. Việc nghiên cứu và phát triển các mô hình tóm tắt văn bản tự động đóng góp vào sự phát triển chung của NLP, đồng thời mở ra các cơ hội ứng dụng mới trong nhiều ngành công nghiệp.
* **Ứng dụng thực tiễn trong nhiều lĩnh vực:**
* Tóm tắt văn bản tự động có thể được ứng dụng trong nhiều lĩnh vực khác nhau như giáo dục, kinh doanh, y tế, và công nghệ thông tin. Ví dụ, trong giáo dục, nó giúp học sinh, sinh viên nắm bắt nhanh nội dung tài liệu học tập. Trong kinh doanh, việc tóm tắt thông tin hỗ trợ quản lý thông tin từ báo cáo, nghiên cứu thị trường. Trong y tế, việc tóm tắt thông tin giúp các bác sĩ và nhà nghiên cứu nhanh chóng nắm bắt thông tin từ các tài liệu y khoa phức tạp.

### 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

Mục tiêu chính của nghiên cứu này là phát triển một mô hình tóm tắt văn bản tự động hiệu quả cho tiếng Việt, đáp ứng các yêu cầu sau:

* Phát triển mô hình tóm tắt văn bản trích xuất dành riêng cho tiếng Việt:
* Thiết kế và xây dựng một mô hình tóm tắt văn bản trích xuất có khả năng lựa chọn và kết hợp các câu quan trọng nhất từ văn bản gốc để tạo ra bản tóm tắt ngắn gọn nhưng vẫn giữ được nội dung chính của văn bản.
* Đánh giá và so sánh với các mô hình hiện tại:
* So sánh mô hình được phát triển với các mô hình tóm tắt văn bản hiện có, để đánh giá khả năng áp dụng và hiệu quả của mô hình đối với 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:

* Tiền xử lý văn bản: nltk và pyvi
* Chuyển đổi văn bản thành vector: FastText và TF-IDF
* Xử lý và tóm tắt văn bản: KMeans
* Đánh giá mô hình tóm tắt: ROUGE

### Phạm vi nghiên cứu

Bộ dữ liệu**: 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.

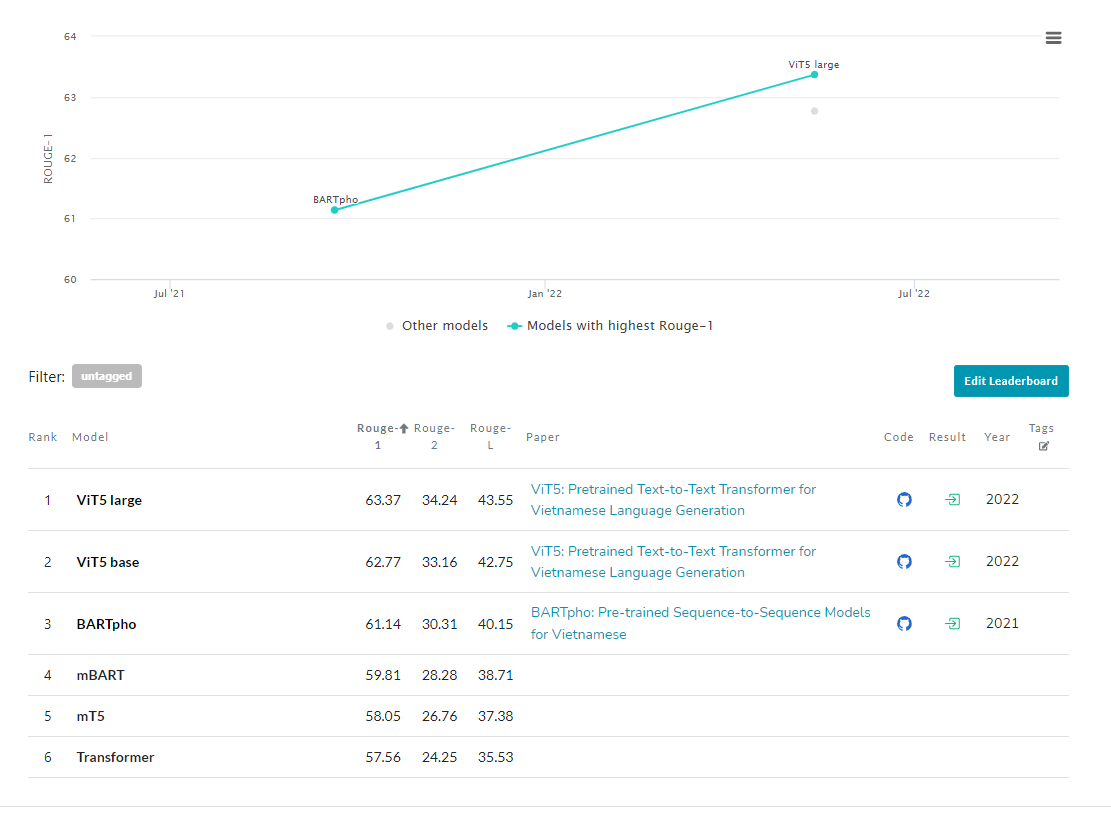
Các tài liệu trong VNDS được chú trọng để đảm bảo rằng chúng bao gồm thông tin chính và ý nghĩa tổng thể của văn bản nguồn. Bộ dữ liệu này cung cấp cơ hội cho các nhà nghiên cứu tiếng Việt nghiên cứu các phương pháp tóm tắt văn bản và đánh giá hiệu suất của chúng trên các tài liệu tiếng Việt thực tế.

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 trên bộ dữ liệu **VNDS (VNDS: A Vietnamese Dataset for Summarization).** Học viên muốn tích hợp các 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 1: Kết quả nghiên cứu của các mô hình trên tập dữ liệu VNDS**

(Nguồn: VNDS: A Vietnamese Dataset for Summarization - Kaggle.com)

## 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 3 chương chính như sau:

Chương 1: Nghiên cứu tổng quan

Chương 2: Các thuật toán liên quan

Chương 3: Tổng kết và thực nghiệm chương trình

# CHƯƠNG 1: NGHIÊN CỨU TỔNG QUAN

## Tổng quan về Text Mining



**Hình 1. 1: Tổng quan về Text Mining**

(Nguồn: Text Mining – Concepts techniques and workflows - imspatial)

### Text Mining là gì?

Khai thác văn bản (text mining), còn được gọi là khai thác dữ liệu văn bản, là quá trình chuyển đổi văn bản không cấu trúc thành định dạng có cấu trúc để xác định các mẫu ý nghĩa và thông tin mới. Chúng ta có thể sử dụng khai thác văn bản để phân tích các tập hợp lớn các tài liệu văn bản nhằm nắm bắt các khái niệm chính, xu hướng và mối quan hệ ẩn.

Bằng cách áp dụng các kỹ thuật phân tích nâng cao, chẳng hạn như Naïve Bayes, Support Vector Machines (SVM) và các thuật toán học sâu khác, các công ty có thể khám phá và phát hiện các mối quan hệ ẩn trong dữ liệu không cấu trúc của họ.

Văn bản là một trong những loại dữ liệu phổ biến nhất trong cơ sở dữ liệu. Tùy thuộc vào cơ sở dữ liệu, dữ liệu này có thể được tổ chức theo các định dạng sau:

* **Dữ liệu có cấu trúc**: Dữ liệu này được chuẩn hóa thành định dạng bảng với nhiều hàng và cột, giúp dễ lưu trữ và xử lý cho phân tích và các thuật toán học máy. Dữ liệu có cấu trúc có thể bao gồm các thông tin như tên, địa chỉ và số điện thoại.
* **Dữ liệu không cấu trúc**: Dữ liệu này không có định dạng dữ liệu được xác định trước. Nó có thể bao gồm văn bản từ các nguồn như mạng xã hội hoặc đánh giá sản phẩm, hoặc các định dạng phương tiện phong phú như video và tệp âm thanh.
* **Dữ liệu bán cấu trúc**: Như tên gọi, dữ liệu này là sự kết hợp giữa các định dạng dữ liệu có cấu trúc và không cấu trúc. Mặc dù nó có một số tổ chức, nhưng không đủ cấu trúc để đáp ứng yêu cầu của cơ sở dữ liệu quan hệ. Các ví dụ về dữ liệu bán cấu trúc bao gồm các tệp XML, JSON và HTML.

Vì khoảng 80% dữ liệu trên thế giới tồn tại dưới dạng không cấu trúc, khai thác văn bản là một thực hành vô cùng giá trị trong các tổ chức. Các công cụ khai thác văn bản và các kỹ thuật xử lý ngôn ngữ tự nhiên (NLP), như trích xuất thông tin, cho phép chúng ta chuyển đổi các tài liệu không cấu trúc thành định dạng có cấu trúc để phân tích và tạo ra những thông tin chất lượng cao. Điều này, lần lượt, cải thiện quy trình ra quyết định của các tổ chức, dẫn đến kết quả kinh doanh tốt hơn.

### Text mining và Text analytics

**Text Mining**

* **Định nghĩa**: Khai thác văn bản là quá trình chuyển đổi dữ liệu văn bản không cấu trúc thành định dạng có cấu trúc để phát hiện các mẫu, xu hướng và mối quan hệ tiềm ẩn. Quá trình này thường sử dụng các phương pháp xử lý ngôn ngữ tự nhiên (NLP) để phân tích nội dung văn bản và trích xuất thông tin có giá trị.
* **Mục tiêu**: Tìm kiếm và khám phá thông tin ẩn trong dữ liệu văn bản, như các chủ đề, mối quan hệ giữa các đối tượng, và các xu hướng tiềm năng.
* **Phương pháp**: Các kỹ thuật bao gồm phân tích ngữ nghĩa, trích xuất thông tin, phân loại văn bản, và phân cụm văn bản. Công cụ phổ biến bao gồm các thuật toán học máy và các phương pháp thống kê.

**Text analytics**

* **Định nghĩa:** Phân tích văn bản là quá trình phân tích các dữ liệu văn bản đã được chuẩn bị để rút ra các thông tin định lượng và chi tiết. Nó sử dụng các kỹ thuật phân tích và trực quan hóa dữ liệu để chuyển đổi thông tin văn bản thành những hiểu biết có thể hành động.
* **Mục tiêu:** Tạo ra các báo cáo và biểu đồ trực quan từ dữ liệu văn bản, cung cấp thông tin chi tiết và định lượng về các mẫu và xu hướng trong dữ liệu.
* **Phương pháp:** Sử dụng các công cụ phân tích thống kê, mô hình hóa dữ liệu, và trực quan hóa dữ liệu để hiểu và trình bày thông tin từ dữ liệu văn bản.

### Các kỹ thuật khai thác văn bản

Quá trình khai thác văn bản bao gồm nhiều hoạt động cho phép chúng ta suy luận thông tin từ dữ liệu văn bản không cấu trúc. Trước khi áp dụng các kỹ thuật khai thác văn bản khác nhau, chúng ta cần bắt đầu với bước tiền xử lý văn bản, là thực hành làm sạch và chuyển đổi dữ liệu văn bản thành định dạng có thể sử dụng. Đây là một khía cạnh cốt lõi của xử lý ngôn ngữ tự nhiên (NLP) và thường bao gồm các kỹ thuật như nhận diện ngôn ngữ, phân tách từ (tokenization), gán nhãn phần của lời (part-of-speech tagging), phân đoạn (chunking) và phân tích cú pháp (syntax parsing) để định dạng dữ liệu phù hợp cho phân tích. Khi hoàn thành bước tiền xử lý văn bản, chúng ta có thể áp dụng các thuật toán khai thác văn bản để rút ra thông tin từ dữ liệu. Một số kỹ thuật khai thác văn bản phổ biến bao gồm:

**Tìm kiếm thông tin (Information Retrieval)**

Tìm kiếm thông tin (IR) trả về thông tin hoặc tài liệu liên quan dựa trên một tập hợp các truy vấn hoặc cụm từ được xác định trước. Hệ thống IR sử dụng các thuật toán để theo dõi hành vi của người dùng và xác định dữ liệu liên quan. Tìm kiếm thông tin thường được sử dụng trong các hệ thống danh mục thư viện và các công cụ tìm kiếm phổ biến như Google. Một số nhiệm vụ phụ của IR bao gồm:

* **Phân tách từ (Tokenization)**: Là quá trình tách văn bản dài thành các câu và từ gọi là “tokens”. Những token này sau đó được sử dụng trong các mô hình như bag-of-words để phân cụm văn bản và các nhiệm vụ khớp tài liệu.
* **Rút gọn từ (Stemming)**: Là quá trình tách các tiền tố và hậu tố khỏi từ để tìm ra dạng gốc và ý nghĩa của từ. Kỹ thuật này cải thiện việc tìm kiếm thông tin bằng cách giảm kích thước của các tệp chỉ mục.

**Xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP)**

Xử lý ngôn ngữ tự nhiên, phát triển từ ngôn ngữ học tính toán, sử dụng các phương pháp từ nhiều lĩnh vực như khoa học máy tính, trí tuệ nhân tạo, ngôn ngữ học và khoa học dữ liệu, để giúp máy tính hiểu ngôn ngữ của con người cả dưới dạng viết và nói. Bằng cách phân tích cấu trúc câu và ngữ pháp, các nhiệm vụ phụ của NLP cho phép máy tính "đọc". Một số nhiệm vụ phụ phổ biến bao gồm:

* **Tóm tắt văn bản (Summarization):** Kỹ thuật này cung cấp một bản tóm tắt ngắn gọn của các văn bản dài để tạo ra một tóm tắt ngắn gọn, mạch lạc về các điểm chính của tài liệu.
* **Gán nhãn phần của lời (Part-of-Speech - PoS tagging):** Kỹ thuật này gán một thẻ cho từng token trong tài liệu dựa trên phần của lời của nó—tức là các danh từ, động từ, tính từ, v.v. Bước này cho phép phân tích ngữ nghĩa trên văn bản không cấu trúc.
* **Phân loại văn bản (Text Categorization):** Nhiệm vụ này, còn được gọi là phân loại văn bản, chịu trách nhiệm phân tích các tài liệu văn bản và phân loại chúng dựa trên các chủ đề hoặc danh mục đã định sẵn. Nhiệm vụ này đặc biệt hữu ích khi phân loại các từ đồng nghĩa và viết tắt.
* **Phân tích cảm xúc (Sentiment Analysis):** Nhiệm vụ này phát hiện cảm xúc tích cực hoặc tiêu cực từ các nguồn dữ liệu nội bộ hoặc bên ngoài, cho phép chúng ta theo dõi sự thay đổi trong thái độ của khách hàng theo thời gian. Nó thường được sử dụng để cung cấp thông tin về nhận thức của các thương hiệu, sản phẩm và dịch vụ. Những thông tin này có thể giúp các doanh nghiệp kết nối với khách hàng và cải thiện quy trình và trải nghiệm của người dùng.

**Trích xuất thông tin (Information Extraction - IE)**

Trích xuất thông tin (IE) tìm ra các phần dữ liệu liên quan khi tìm kiếm các tài liệu khác nhau. Nó cũng tập trung vào việc trích xuất thông tin có cấu trúc từ văn bản tự do và lưu trữ các thực thể, thuộc tính và thông tin mối quan hệ trong cơ sở dữ liệu. Một số nhiệm vụ phụ của trích xuất thông tin bao gồm:

* **Chọn đặc trưng (Feature Selection),** hay còn gọi là chọn thuộc tính, là quá trình chọn các đặc trưng quan trọng (kích thước) để đóng góp nhiều nhất vào kết quả của mô hình phân tích dự đoán.
* **Trích xuất đặc trưng (Feature Extraction)** là quá trình chọn một tập hợp con của các đặc trưng để cải thiện độ chính xác của nhiệm vụ phân loại. Đây là bước quan trọng trong việc giảm kích thước.
* **Nhận diện thực thể tên (Named-Entity Recognition - NER),** còn được gọi là nhận diện thực thể hoặc trích xuất thực thể, nhằm tìm và phân loại các thực thể cụ thể trong văn bản, chẳng hạn như tên hoặc địa điểm. Ví dụ, NER xác định “California” là một địa điểm và “Mary” là tên của một phụ nữ.

**Khai thác dữ liệu (Data Mining)**

Khai thác dữ liệu là quá trình xác định các mẫu và rút ra thông tin hữu ích từ các tập dữ liệu lớn. Thực hành này đánh giá cả dữ liệu có cấu trúc và không cấu trúc để xác định thông tin mới, và thường được sử dụng để phân tích hành vi của người tiêu dùng trong marketing và bán hàng. Khai thác văn bản về cơ bản là một lĩnh vực con của khai thác dữ liệu vì nó tập trung vào việc đưa cấu trúc cho dữ liệu không cấu trúc và phân tích nó để tạo ra các thông tin mới. Các kỹ thuật đã đề cập ở trên là các hình thức khai thác dữ liệu nhưng thuộc phạm vi phân tích dữ liệu văn bản.

### 1.1.4 Ứng dụng của khai thác văn bản

Phần mềm phân tích văn bản đã tác động đến cách thức hoạt động của nhiều ngành công nghiệp, cho phép họ cải thiện trải nghiệm người dùng sản phẩm cũng như đưa ra các quyết định kinh doanh nhanh hơn và tốt hơn. Một số trường hợp sử dụng bao gồm:

**Dịch vụ khách hàng**: Có nhiều cách mà chúng ta thu thập phản hồi từ khách hàng. Khi kết hợp với các công cụ phân tích văn bản, các hệ thống phản hồi như chatbot, khảo sát khách hàng, NPS (chỉ số khuyến nghị), đánh giá trực tuyến, phiếu hỗ trợ, và hồ sơ trên mạng xã hội, cho phép các công ty cải thiện trải nghiệm khách hàng của mình một cách nhanh chóng. Khai thác văn bản và phân tích cảm xúc có thể cung cấp một cơ chế để các công ty ưu tiên các vấn đề quan trọng đối với khách hàng, giúp doanh nghiệp phản hồi các vấn đề khẩn cấp trong thời gian thực và tăng sự hài lòng của khách hàng. Ví dụ như cách Verizon đang sử dụng phân tích văn bản trong dịch vụ khách hàng.

**Quản lý rủi ro**: Khai thác văn bản cũng có ứng dụng trong quản lý rủi ro, nơi nó có thể cung cấp những thông tin sâu sắc về xu hướng ngành và thị trường tài chính bằng cách theo dõi sự thay đổi trong cảm xúc và trích xuất thông tin từ các báo cáo phân tích và sách trắng. Điều này đặc biệt có giá trị đối với các tổ chức ngân hàng vì dữ liệu này cung cấp sự tự tin hơn khi xem xét các khoản đầu tư kinh doanh trên nhiều lĩnh vực. Ví dụ như cách CIBC và EquBot đang sử dụng phân tích văn bản để giảm thiểu rủi ro.

**Bảo trì**: Khai thác văn bản cung cấp một bức tranh toàn diện về hoạt động và chức năng của sản phẩm và máy móc. Theo thời gian, khai thác văn bản tự động hóa việc ra quyết định bằng cách tiết lộ các mẫu liên quan đến các vấn đề và quy trình bảo trì phòng ngừa và phản ứng. Phân tích văn bản giúp các chuyên gia bảo trì nhanh chóng phát hiện ra nguyên nhân gốc rễ của các thách thức và sự cố.

**Chăm sóc sức khỏe**: Các kỹ thuật khai thác văn bản ngày càng có giá trị đối với các nhà nghiên cứu trong lĩnh vực y sinh, đặc biệt là trong việc phân cụm thông tin. Việc điều tra thủ công các nghiên cứu y khoa có thể tốn kém và mất thời gian; khai thác văn bản cung cấp một phương pháp tự động để trích xuất thông tin có giá trị từ tài liệu y khoa.

**Lọc thư rác**: Thư rác thường là điểm vào cho tin tặc để lây nhiễm hệ thống máy tính bằng phần mềm độc hại. Khai thác văn bản có thể cung cấp một phương pháp để lọc và loại trừ các email này khỏi hộp thư đến, cải thiện trải nghiệm người dùng tổng thể và giảm thiểu nguy cơ bị tấn công mạng cho người dùng cuối.

## Tổng quan về NLP



**Hình 1. 2: Tổng quan về NLP**

(Nguồn: Deep Learning for Natural Language Processing - GippLab)

### 1.2.1 NLP là gì?

Xử lý ngôn ngữ tự nhiên (NLP) là một nhánh của khoa học máy tính và trí tuệ nhân tạo (AI) sử dụng học máy để giúp máy tính hiểu và giao tiếp bằng ngôn ngữ của con người.

NLP cho phép máy tính và các thiết bị kỹ thuật số nhận diện, hiểu và tạo ra văn bản cũng như lời nói bằng cách kết hợp ngôn ngữ học tính toán - việc mô hình hóa ngôn ngữ con người dựa trên quy tắc - cùng với mô hình hóa thống kê, học máy (ML) và học sâu.

Nghiên cứu về NLP đã thúc đẩy sự phát triển của AI thế hệ mới, từ khả năng giao tiếp của các mô hình ngôn ngữ lớn (LLM) đến khả năng hiểu yêu cầu của các mô hình tạo hình ảnh. NLP đã trở thành một phần trong cuộc sống hàng ngày của nhiều người, cung cấp năng lượng cho các công cụ tìm kiếm, chatbot hỗ trợ dịch vụ khách hàng với các lệnh thoại, hệ thống định vị GPS điều khiển bằng giọng nói, và trợ lý kỹ thuật số trên điện thoại thông minh.

NLP cũng đóng vai trò ngày càng quan trọng trong các giải pháp doanh nghiệp, giúp hợp lý hóa và tự động hóa các hoạt động kinh doanh, tăng năng suất của nhân viên và đơn giản hóa các quy trình kinh doanh quan trọng.

### 1.2.2 Lợi ích của NLP

Một hệ thống xử lý ngôn ngữ tự nhiên có thể hoạt động nhanh chóng và hiệu quả: sau khi các mô hình NLP được huấn luyện đúng cách, nó có thể đảm nhận các công việc hành chính, giải phóng nhân viên để tập trung vào công việc năng suất hơn. Những lợi ích có thể bao gồm:

**Khám phá thông tin nhanh hơn:** Các tổ chức có thể tìm thấy các mô hình ẩn, xu hướng và mối quan hệ giữa các phần nội dung khác nhau. Việc truy xuất dữ liệu văn bản hỗ trợ cho việc phân tích sâu hơn, giúp đưa ra quyết định thông minh hơn và nảy sinh những ý tưởng kinh doanh mới.

**Tiết kiệm ngân sách lớn hơn:** Với khối lượng dữ liệu văn bản không có cấu trúc khổng lồ hiện có, NLP có thể được sử dụng để tự động hóa việc thu thập, xử lý và tổ chức thông tin với ít nỗ lực thủ công hơn.

**Truy cập nhanh vào dữ liệu doanh nghiệp:** Một doanh nghiệp có thể xây dựng cơ sở tri thức về thông tin tổ chức để truy cập hiệu quả thông qua tìm kiếm AI. Đối với các đại diện bán hàng, NLP có thể giúp nhanh chóng truy xuất thông tin liên quan, cải thiện dịch vụ khách hàng và hỗ trợ chốt giao dịch.

### 1.2.3 Những thách thức với NLP

Các mô hình NLP không hoàn hảo và có lẽ sẽ không bao giờ đạt đến sự hoàn hảo, giống như việc ngôn ngữ của con người cũng dễ gặp sai sót. Những rủi ro có thể bao gồm:

**Biased training**: Giống như bất kỳ chức năng AI nào, **Biased training** được sử dụng trong quá trình huấn luyện sẽ làm lệch kết quả. Người dùng của một chức năng NLP càng đa dạng, rủi ro này càng trở nên đáng kể, chẳng hạn như trong các dịch vụ chính phủ, y tế và tương tác nhân sự. Các tập dữ liệu huấn luyện được lấy từ web, ví dụ như vậy, rất dễ bị thiên vị.

**Misinterpretation**: Giống như trong lập trình, có rủi ro của nguyên tắc " garbage in, garbage out" (GIGO). Các giải pháp NLP có thể bị nhầm lẫn nếu đầu vào giọng nói là phương ngữ không quen thuộc, nói lắp bắp, quá nhiều tiếng lóng, từ đồng âm, ngữ pháp sai, thành ngữ, câu vụn, phát âm sai hoặc được ghi âm với quá nhiều tiếng ồn nền.

**New vocabulary:**Các từ mới liên tục được tạo ra hoặc du nhập. Các quy ước ngữ pháp có thể phát triển hoặc bị phá vỡ có chủ đích. Trong những trường hợp này, NLP có thể đưa ra phỏng đoán tốt nhất hoặc thừa nhận nó không chắc chắn—và dù thế nào, điều này cũng tạo ra sự phức tạp.Những công việc phổ biến áp dụng NLP.

**Tone of voice**: Khi con người nói, cách phát âm hoặc ngôn ngữ cơ thể của họ có thể mang lại ý nghĩa hoàn toàn khác so với từ ngữ. Việc phóng đại để tạo hiệu ứng, nhấn mạnh từ để nhấn mạnh hoặc sự mỉa mai có thể khiến NLP hiểu lầm, làm cho việc phân tích ngữ nghĩa trở nên khó khăn và ít đáng tin cậy hơn.

Ngôn ngữ của con người chứa đầy những điều mơ hồ khiến cho việc lập trình phần mềm để xác định chính xác ý nghĩa của văn bản hoặc dữ liệu giọng nói trở nên khó khăn. Ngôn ngữ của con người có thể mất nhiều năm để con người học, và nhiều người không bao giờ ngừng học. Nhưng sau đó, các lập trình viên phải dạy các ứng dụng dựa trên ngôn ngữ tự nhiên để nhận ra và hiểu các điểm không đều để các ứng dụng của họ có thể chính xác và hữu ích.

### 1.2.4 Cách hoạt động của NLP

NLP kết hợp sức mạnh của ngôn ngữ học tính toán cùng với các thuật toán học máy và học sâu. Ngôn ngữ học tính toán là một lĩnh vực của ngôn ngữ học sử dụng khoa học dữ liệu để phân tích ngôn ngữ và giọng nói. Nó bao gồm hai loại phân tích chính: phân tích cú pháp và phân tích ngữ nghĩa. Phân tích cú pháp xác định ý nghĩa của một từ, cụm từ hoặc câu bằng cách phân tích cú pháp của các từ và áp dụng các quy tắc ngữ pháp được lập trình trước. Phân tích ngữ nghĩa sử dụng đầu ra của phân tích cú pháp để rút ra ý nghĩa từ các từ và giải thích ý nghĩa của chúng trong cấu trúc câu.

Việc phân tích cú pháp của các từ có thể diễn ra dưới hai hình thức. Phân tích cú pháp phụ thuộc xem xét mối quan hệ giữa các từ, chẳng hạn như xác định danh từ và động từ, trong khi phân tích cú pháp cấu trúc tạo ra một cây phân tích cú pháp (hoặc cây cú pháp): một đại diện gốc và có thứ tự của cấu trúc cú pháp của câu hoặc chuỗi từ. Các cây phân tích cú pháp kết quả này là nền tảng cho các chức năng của trình dịch ngôn ngữ và nhận diện giọng nói. Lý tưởng nhất, phân tích này giúp đầu ra dù là văn bản hay giọng nói trở nên dễ hiểu đối với cả mô hình NLP và con người.

Học tự giám sát (Self-supervised learning - SSL) đặc biệt hữu ích cho việc hỗ trợ NLP vì NLP yêu cầu lượng lớn dữ liệu gán nhãn để huấn luyện các mô hình trí tuệ nhân tạo (AI) tiên tiến. Vì các tập dữ liệu được gán nhãn này yêu cầu sự chú thích tốn thời gian, một quá trình liên quan đến việc gán nhãn thủ công bởi con người, việc thu thập đủ dữ liệu có thể trở nên cực kỳ khó khăn. Các phương pháp tiếp cận tự giám sát có thể tiết kiệm thời gian và chi phí hơn, vì chúng thay thế một phần hoặc toàn bộ dữ liệu huấn luyện được gán nhãn thủ công.

**Ba cách tiếp cận khác nhau với NLP bao gồm:**

**NLP dựa trên quy tắc:** Các ứng dụng NLP đầu tiên là các cây quyết định if-then đơn giản, yêu cầu các quy tắc được lập trình sẵn. Chúng chỉ có thể cung cấp câu trả lời để đáp ứng các lời nhắc cụ thể, chẳng hạn như phiên bản gốc của Moviefone. Vì không có khả năng học máy hoặc AI trong NLP dựa trên quy tắc, chức năng này bị giới hạn và không thể mở rộng.

**NLP thống kê:** Phát triển sau này, NLP thống kê tự động trích xuất, phân loại và gán nhãn các yếu tố của văn bản và dữ liệu giọng nói, sau đó gán xác suất thống kê cho mỗi ý nghĩa có thể của các yếu tố đó. Điều này dựa vào học máy, cho phép phân tích chi tiết phức tạp về ngôn ngữ học như gán nhãn phần của từ.

**NLP học sâu:** Gần đây, các mô hình học sâu đã trở thành phương thức chủ yếu của NLP bằng cách sử dụng khối lượng lớn dữ liệu thô, không cấu trúc gồm cả văn bản và giọng nói, để ngày càng trở nên chính xác hơn. Học sâu có thể được xem như là một sự tiến hóa tiếp theo của NLP thống kê, với sự khác biệt là nó sử dụng các mô hình mạng nơ-ron. Có một số loại mô hình con:

* Mô hình Sequence-to-Sequence (seq2seq): Dựa trên mạng nơ-ron hồi quy (RNN), chúng chủ yếu được sử dụng cho dịch máy bằng cách chuyển đổi một cụm từ từ một miền (chẳng hạn như ngôn ngữ Đức) thành cụm từ của một miền khác (chẳng hạn như tiếng Anh).
* Mô hình Transformer: Chúng sử dụng việc chia nhỏ ngôn ngữ (vị trí của mỗi token - từ hoặc từ con) và sự tự chú ý (ghi lại sự phụ thuộc và mối quan hệ) để tính toán mối quan hệ của các phần ngôn ngữ khác nhau với nhau. Các mô hình Transformer có thể được huấn luyện hiệu quả bằng cách sử dụng học tự giám sát trên các cơ sở dữ liệu văn bản khổng lồ. Một dấu mốc trong các mô hình Transformer là BERT của Google, đây là cơ sở của cách mà công cụ tìm kiếm của Google hoạt động và vẫn đang tiếp tục phát triển.
* Mô hình tự hồi quy: Loại mô hình Transformer này được huấn luyện đặc biệt để dự đoán từ tiếp theo trong một chuỗi, điều này đại diện cho một bước nhảy vọt trong khả năng tạo văn bản. Các ví dụ về các mô hình tự hồi quy LLM bao gồm GPT, Llama, Claude và mã nguồn mở Mistral.
* Mô hình nền tảng (Foundation models): Các mô hình nền tảng được xây dựng sẵn và tuyển chọn có thể tăng tốc việc khởi chạy một nỗ lực NLP và tăng cường độ tin cậy trong hoạt động của nó. Ví dụ, các mô hình nền tảng Granite™ của IBM được áp dụng rộng rãi trên các ngành công nghiệp. Chúng hỗ trợ các nhiệm vụ NLP bao gồm tạo nội dung và trích xuất thông tin chi tiết. Ngoài ra, chúng hỗ trợ khung tạo dựa trên truy xuất, một khung để cải thiện chất lượng phản hồi bằng cách liên kết mô hình với các nguồn kiến thức bên ngoài. Các mô hình này cũng thực hiện nhận diện thực thể có tên, bao gồm việc xác định và trích xuất thông tin chính trong văn bản.

### 1.2.5 Các tác vụ NLP

Nhiều tác vụ NLP thường giúp xử lý dữ liệu văn bản và giọng nói của con người theo cách mà máy tính có thể hiểu được. Một số tác vụ này bao gồm:

**Các tác vụ ngôn ngữ học**

* Xác định đồng tham chiếu (Coreference resolution): Đây là nhiệm vụ xác định xem hai từ có đề cập đến cùng một thực thể hay không và khi nào. Ví dụ phổ biến nhất là xác định người hoặc đối tượng mà một đại từ nhất định đề cập đến (chẳng hạn như, "cô ấy" = "Mary"). Nhưng nó cũng có thể xác định một phép ẩn dụ hoặc thành ngữ trong văn bản (chẳng hạn như trường hợp "bear" không phải là một con vật, mà là một người to lớn và nhiều lông).
* **Nhận dạng thực thể có tên (Named entity recognition - NER)**: Nhiệm vụ này xác định các từ hoặc cụm từ là các thực thể có ích. Ví dụ, NER xác định “London” là một địa điểm hoặc “Maria” là tên của một người.
* **Gán nhãn từ loại (Part-of-speech tagging)**: Còn được gọi là gán nhãn ngữ pháp, đây là quá trình xác định từ loại của một từ hoặc một phần của văn bản, dựa trên ngữ cảnh và cách sử dụng. Ví dụ, gán nhãn từ loại xác định “make” là động từ trong “I can make a paper plane,” và là danh từ trong “What make of car do you own?”
* **Phân biệt nghĩa của từ (Word sense disambiguation)**: Đây là việc chọn nghĩa của từ có nhiều nghĩa khác nhau. Quá trình này sử dụng phân tích ngữ nghĩa để xem xét từ trong ngữ cảnh. Ví dụ, phân biệt nghĩa của từ giúp phân biệt nghĩa của động từ “make” trong “make the grade” (đạt được) so với “make a bet” (đặt cược). Việc giải mã “I will be merry when I marry Mary” yêu cầu một hệ thống NLP tinh vi.

**Các tác vụ hỗ trợ người dùng**

* **Nhận diện giọng nói (Speech recognition):** Còn được gọi là chuyển giọng nói thành văn bản, đây là nhiệm vụ chuyển đổi dữ liệu giọng nói thành dữ liệu văn bản một cách chính xác. Nhận diện giọng nói là một phần của bất kỳ ứng dụng nào theo dõi lệnh giọng nói hoặc trả lời các câu hỏi được nói. Điều làm cho việc nhận diện giọng nói trở nên đặc biệt khó khăn là cách mọi người nói—nhanh, nối liền các từ, với nhiều cách nhấn âm và ngữ điệu khác nhau.
* **Tạo ngôn ngữ tự nhiên (Natural language generation - NLG):** NLG có thể được mô tả như là quá trình ngược lại của nhận diện giọng nói hoặc chuyển giọng nói thành văn bản: NLG là nhiệm vụ chuyển đổi thông tin có cấu trúc thành ngôn ngữ tự nhiên của con người. Nếu không có NLG, máy tính sẽ khó có cơ hội vượt qua bài kiểm tra Turing, nơi mà máy tính cố gắng mô phỏng một cuộc hội thoại của con người. Các trợ lý ảo như Alexa của Amazon và Siri của Apple đang thực hiện tốt nhiệm vụ này và hỗ trợ khách hàng trong thời gian thực.
* **Hiểu ngôn ngữ tự nhiên (Natural language understanding -NLU):** Đây là một nhánh của NLP tập trung vào việc phân tích ý nghĩa đằng sau các câu. NLU cho phép phần mềm tìm ra các ý nghĩa tương tự trong các câu khác nhau hoặc xử lý các từ có nhiều nghĩa khác nhau.
* **Phân tích cảm xúc (Sentiment analysis):** Nhiệm vụ này cố gắng trích xuất các phẩm chất chủ quan - thái độ, cảm xúc, sự mỉa mai, nhầm lẫn hoặc nghi ngờ từ văn bản. Điều này thường được sử dụng để định tuyến các giao tiếp đến hệ thống hoặc người có khả năng đưa ra phản hồi tiếp theo.

### 1.2.6 Các trường hợp sử dụng NLP

Các tổ chức có thể sử dụng NLP để xử lý các hình thức giao tiếp bao gồm email, tin nhắn SMS, âm thanh, video, nguồn tin tức và mạng xã hội. NLP là động lực đằng sau AI trong nhiều ứng dụng thực tế hiện đại. Dưới đây là một vài ví dụ:

**Hỗ trợ khách hàng**: Các doanh nghiệp có thể triển khai chatbot hoặc trợ lý ảo để nhanh chóng phản hồi các câu hỏi và yêu cầu của khách hàng. Khi các câu hỏi trở nên quá khó đối với chatbot hoặc trợ lý ảo, hệ thống NLP sẽ chuyển khách hàng sang nhân viên hỗ trợ khách hàng thực sự.

**Câu hỏi thường gặp (FAQ)**: Không phải ai cũng muốn đọc để tìm câu trả lời. Và NLP có thể cải thiện các FAQ: Khi người dùng đặt câu hỏi, chức năng NLP sẽ tìm câu trả lời phù hợp nhất trong số các câu trả lời có sẵn và hiển thị cho người dùng. Nhiều câu hỏi của khách hàng thuộc loại ai/cái gì/khi nào/ở đâu, vì vậy chức năng này có thể tiết kiệm công sức của nhân viên khỏi việc phải trả lời các câu hỏi thường gặp lặp đi lặp lại.

**Chỉnh sửa ngữ pháp**: Các quy tắc ngữ pháp có thể được áp dụng trong các chương trình xử lý văn bản hoặc các chương trình khác, nơi chức năng NLP được đào tạo để phát hiện ngữ pháp sai và đề xuất các cách diễn đạt sửa đổi.

**Dịch máy**: Google Dịch là một ví dụ về công nghệ NLP có sẵn. Dịch máy thực sự hữu ích đòi hỏi nhiều hơn là chỉ thay thế từ trong một ngôn ngữ bằng từ trong ngôn ngữ khác. Dịch máy hiệu quả nắm bắt chính xác ý nghĩa và tông giọng của ngôn ngữ gốc và dịch chúng thành văn bản có cùng ý nghĩa và tác động mong muốn trong ngôn ngữ đích.

**Xóa thông tin cá nhân nhận diện (PII)**: Các mô hình NLP có thể được đào tạo để nhanh chóng xác định thông tin cá nhân trong các tài liệu có thể nhận diện cá nhân. Các ngành công nghiệp xử lý lượng lớn thông tin nhạy cảm như: tài chính, y tế, bảo hiểm và pháp lý, có thể nhanh chóng tạo các phiên bản đã xóa PII.

**Phân tích cảm xúc**: Sau khi được đào tạo về ngôn ngữ ngành hoặc ngôn ngữ doanh nghiệp cụ thể, một mô hình NLP có thể nhanh chóng quét văn bản nhập vào để tìm các từ khóa và cụm từ nhằm đánh giá tâm trạng của khách hàng theo thời gian thực như tích cực, trung lập hoặc tiêu cực. Tâm trạng của giao tiếp sẽ giúp xác định cách xử lý. Và giao tiếp không nhất thiết phải trực tiếp: NLP cũng có thể được sử dụng để phân tích phản hồi của khách hàng hoặc các bản ghi cuộc gọi trung tâm.

**Phát hiện thư rác**: Nhiều người có thể không nghĩ rằng phát hiện thư rác là một giải pháp NLP, nhưng công nghệ phát hiện thư rác tốt nhất sử dụng khả năng phân loại văn bản của NLP để quét email nhằm phát hiện ngôn ngữ chỉ ra thư rác hoặc lừa đảo. Các chỉ số này có thể bao gồm lạm dụng các thuật ngữ tài chính, ngữ pháp xấu đặc trưng, ngôn ngữ đe dọa, khẩn cấp không phù hợp hoặc tên công ty bị viết sai chính tả.

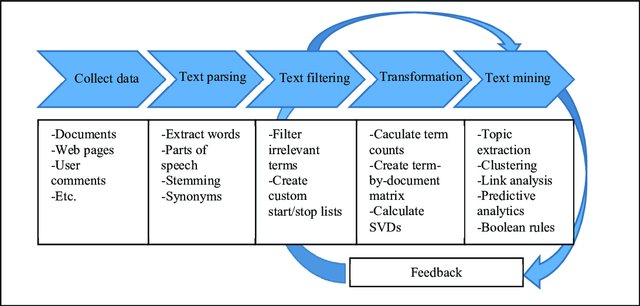
**Tạo văn bản**: NLP giúp đưa "tạo sinh" vào AI tạo sinh. NLP cho phép máy tính tạo văn bản hoặc giọng nói tự nhiên và thực tế đến mức có thể bị nhầm lẫn với giao tiếp của con người. Ngôn ngữ tạo sinh có thể được sử dụng để tạo bản nháp ban đầu của blog, mã máy tính, thư từ, ghi nhớ hoặc tweet. Với hệ thống cấp doanh nghiệp, chất lượng ngôn ngữ tạo sinh có thể đủ tốt để sử dụng trong thời gian thực cho các chức năng tự động hoàn thành, chatbot hoặc trợ lý ảo.

**Tóm tắt văn bản**: Tóm tắt văn bản sử dụng các kỹ thuật NLP để xử lý lượng lớn văn bản kỹ thuật số và tạo ra các bản tóm tắt và tóm lược cho các chỉ số, cơ sở dữ liệu nghiên cứu, dành cho những người đọc bận rộn không có thời gian đọc toàn bộ văn bản. Các ứng dụng tóm tắt văn bản tốt nhất sử dụng lý luận ngữ nghĩa và tạo ngôn ngữ tự nhiên (NLG) để thêm bối cảnh hữu ích và kết luận cho các bản tóm tắt.

# CHƯƠNG 2: CÁC THUẬT TOÁN LIÊN QUAN

## 2.1 Các bước xử lý Text Mining

Văn bản là ngôn ngữ viết trên nền tảng kỹ thuật số. Lượng văn bản hiện có ngày nay đang không ngừng gia tăng. Các kỹ thuật khai phá dữ liệu (DM) truyền thống không đủ để phân tích dữ liệu phi cấu trúc. Chúng ta cần sử dụng một số phương pháp tiếp cận ngôn ngữ học. Quá trình khai phá văn bản (TM) xử lý thông tin phi cấu trúc, trích xuất các chỉ số số học có ý nghĩa từ văn bản, và từ đó, giúp cho thông tin chứa trong văn bản có thể truy cập được bởi các thuật toán khai phá dữ liệu khác nhau, bao gồm cả thống kê và học máy. Các doanh nghiệp sử dụng khai phá dữ liệu và khai phá văn bản để phân tích dữ liệu khách hàng và đối thủ cạnh tranh nhằm cải thiện khả năng cạnh tranh. Lợi ích của khai phá văn bản rất rõ ràng trong các lĩnh vực nơi mà một lượng lớn dữ liệu văn bản được thu thập từ các giao dịch kinh doanh. **Hình 2.1** minh họa quy trình khai phá văn bản.



**Hình 2. 1: Các bước xử lý Text Mining**

(Nguồn: Mining Unstructured Turkish Economy News Articles - Procedia Economics and Finance 16)

### 2.1.1 Thu thập dữ liệu

* Thu thập dữ liệu từ nhiều nguồn khác nhau như văn bản, trang web, cơ sở dữ liệu, mạng xã hội.

### 2.1.2 Tiền xử lý dữ liệu

* Loại bỏ nhiễu: Loại bỏ các thành phần không cần thiết như HTML tags, ký tự đặc biệt, số, hoặc các từ dừng (stopwords).
* Chuẩn hóa văn bản: Chuyển đổi văn bản về dạng thống nhất, như viết thường tất cả chữ cái, chuyển đổi số thành từ.
* Tách từ: Chia văn bản thành các từ hoặc cụm từ có ý nghĩa.
* Gán nhãn từ loại: Xác định loại từ của từng từ trong văn bản (danh từ, động từ, tính từ).

### 2.1.3 Biểu diễn văn bản

* **Bag of Words** (BoW): Biểu diễn văn bản dưới dạng vector số, dựa trên tần suất xuất hiện của các từ.
* **TF-IDF** (Term Frequency-Inverse Document Frequency): Phương pháp đánh trọng số cho mỗi từ trong văn bản dựa trên tần suất xuất hiện và tần suất xuất hiện trong toàn bộ tập dữ liệu.
* **Word Embeddings:** Sử dụng các mô hình như Word2Vec, GloVe hoặc BERT để biểu diễn từ dưới dạng vector với ngữ nghĩa liên quan.

### 2.1.4 Phân tích và khai phá dữ liệu

* Phân cụm văn bản (Text Clustering): Nhóm các văn bản tương tự nhau thành các cụm.
* Phân loại văn bản (Text Classification): Xác định nhãn hoặc chủ đề cho các văn bản dựa trên nội dung.
* Trích xuất thực thể có tên (Named Entity Recognition - NER): Xác định và phân loại các thực thể như tên người, địa điểm, tổ chức, v.v.
* Phân tích cảm xúc (Sentiment Analysis): Xác định cảm xúc hoặc thái độ của người viết trong văn bản.

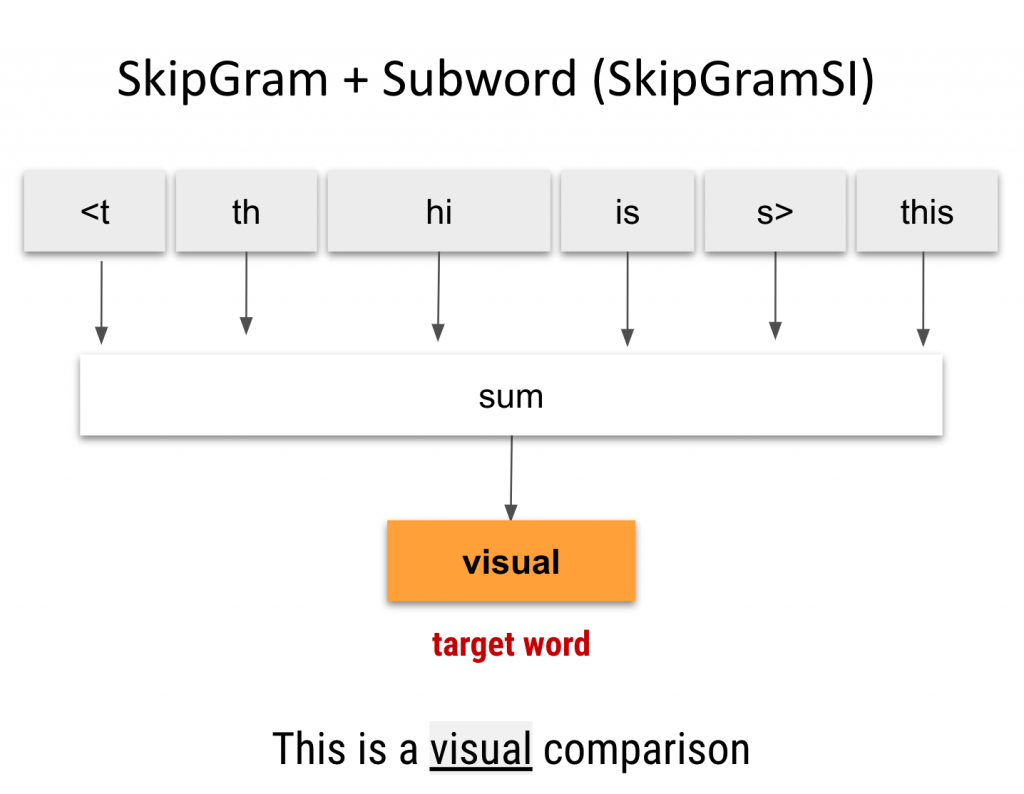
### 2.1.5 Đánh giá kết quả

* Sử dụng các chỉ số đánh giá như độ chính xác, độ phủ, F1-score, ROUGE, để đánh giá hiệu quả của mô hình.

## 2.2 Thuật toán và độ đo

### 2.2.1 FastText

FastText là một kỹ thuật nhúng từ nâng cao do Facebook AI Research (FAIR) phát triển, mở rộng từ mô hình Word2Vec. Khác với Word2Vec, FastText không chỉ xem xét từ nguyên vẹn mà còn kết hợp thông tin về các thành phần con của từ như các n-grams. Cách tiếp cận này cho phép xử lý các ngôn ngữ có cấu trúc hình thái phong phú và nắm bắt thông tin về cấu trúc của từ hiệu quả hơn.



**Hình 2. 2: SkipGram với thông tin subword (character n-gram size=2)**

(Nguồn: Advanced Word Embeddings: Word2Vec, GloVe, and FastText- The Complete NLP Guide: Text to Context)

**Thông tin về thành phần con của từ:** FastText đại diện cho mỗi từ dưới dạng một tập hợp các n-grams ký tự, ngoài từ nguyên vẹn. Điều này có nghĩa là từ “apple” được đại diện bởi chính từ đó và các n-grams thành phần như “ap”, “pp”, “pl”, “le”, v.v. Cách tiếp cận này giúp nắm bắt ý nghĩa của các từ ngắn hơn và cung cấp sự hiểu biết tốt hơn về hậu tố và tiền tố.

**Huấn luyện mô hình:** Tương tự như Word2Vec, FastText có thể sử dụng kiến trúc CBOW hoặc Skip-gram. Tuy nhiên, nó tích hợp thông tin về các thành phần con của từ trong quá trình huấn luyện. Mạng nơ-ron trong FastText được huấn luyện để dự đoán từ (trong CBOW) hoặc ngữ cảnh (trong Skip-gram) không chỉ dựa trên các từ mục tiêu mà còn dựa trên các n-grams này.

**Xử lý các từ hiếm và từ chưa biết:** Một lợi thế lớn của FastText là khả năng tạo ra các biểu diễn từ tốt hơn cho các từ hiếm hoặc thậm chí là các từ chưa từng xuất hiện trong quá trình huấn luyện. Bằng cách phân tách từ thành các n-grams, FastText có thể xây dựng các biểu diễn có ý nghĩa cho các từ này dựa trên các thành phần con của chúng.

**Ưu điểm và nhược điểm:**

**Ưu điểm:**

* Biểu diễn tốt hơn cho các từ hiếm.
* Khả năng xử lý các từ ngoài từ vựng.
* Biểu diễn từ phong phú hơn nhờ thông tin thành phần con của từ.

**Nhược điểm:**

* Tăng kích thước mô hình do thông tin n-gram.
* Thời gian huấn luyện lâu hơn so với Word2Vec.

### 2.2.2 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:
   1. Xác định nhãn cho từng điểm dữ liệu 𝑐𝑖 dựa vào khoảng cách tới từng tâm cụm:

(Nguồn: University of Cincinnati Business Analytics. K-Means Cluster Analysis. In: UC Business Analytics R Programming Guide)

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

(Nguồn: University of Cincinnati Business Analytics. K-Means Cluster Analysis. In: UC Business Analytics R Programming Guide)

**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.

### 2.2.3 Distance Measurement.

Kỹ thuật: ***pairwise\_distances\_argmin\_min*** để đo khoảng cách giữa các vector câu và tâm cụm.

KMeans sử dụng khoảng cách Euclidean để đo khoảng cách giữa các điểm và các tâm cụm:

(Nguồn: University of Cincinnati Business Analytics. K-Means Cluster Analysis. In: UC Business Analytics R Programming Guide)

### 2.2.4 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.

(Nguồn: TF-IDF in NLP (Term Frequency Inverse Document Frequency) - Medium)

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

(Nguồn: TF-IDF in NLP (Term Frequency Inverse Document Frequency) - Medium)

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

(Nguồn: TF-IDF in NLP (Term Frequency Inverse Document Frequency) - Medium)

* 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.

### 2.2.5 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:**

(Nguồn: ROUGE: A Package for Automatic Evaluation of Summaries - Chin-Yew Lin)

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

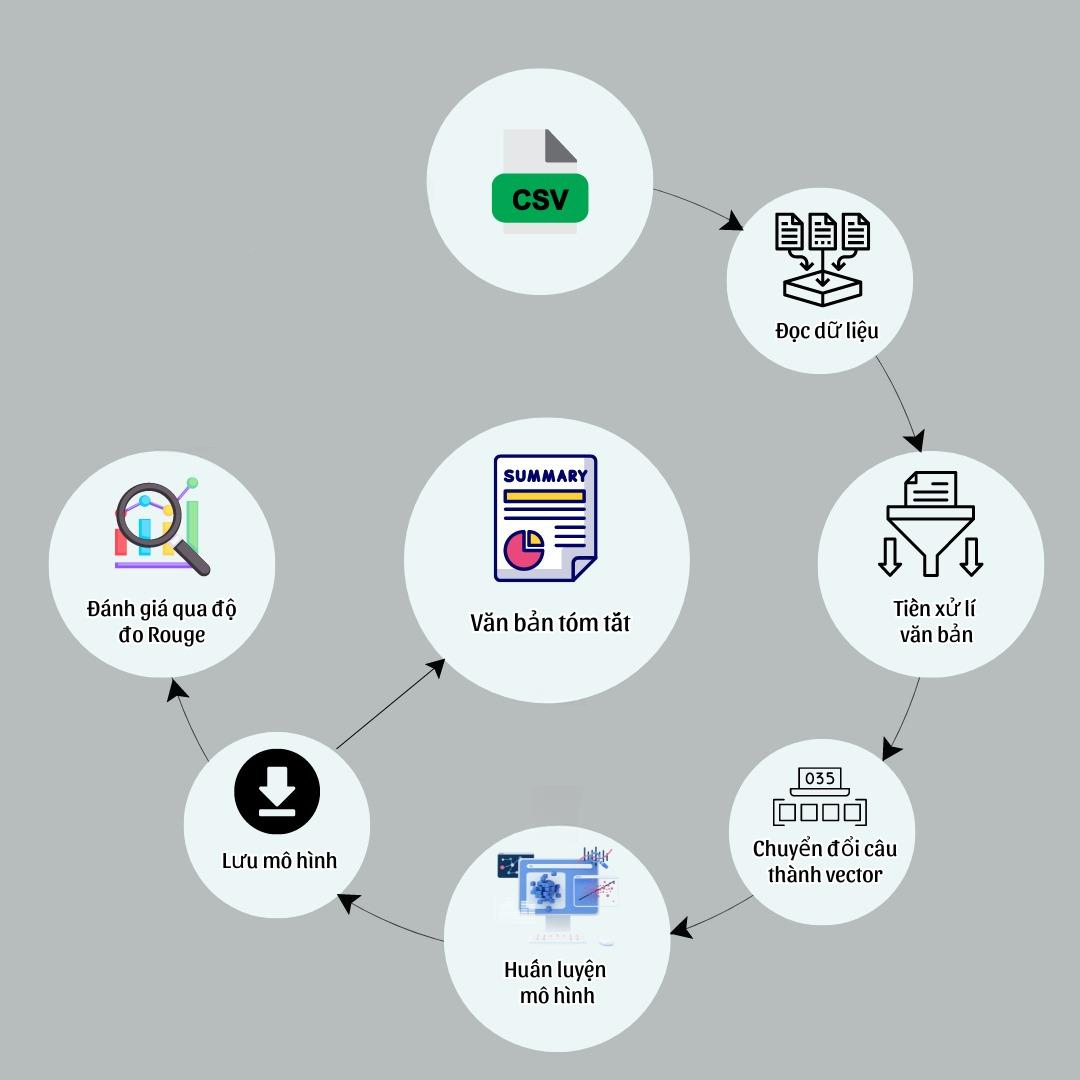
**ROUGE-2**: Đo lường sự trùng khớp của các bigram (cặp từ liên tiếp) giữa bản tóm tắt máy và bản tóm tắt tham chiếu.

* 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.

## 2.3 Giải pháp đề xuất tóm tắt văn bản

### 2.3.1 Mô hình tóm tắt văn bản đề xuất

Mô hình tóm tắt văn bản được đề xuất trong nghiên cứu này dựa trên sự kết hợp của các phương pháp xử lý ngôn ngữ tự nhiên và kỹ thuật học máy, cụ thể là mô hình FastText và K-Means. Mục tiêu của mô hình là tạo ra các bản tóm tắt ngắn gọn và chính xác từ các đoạn văn bản dài, giữ lại những nội dung quan trọng nhất.



**Hình 2. 3: Mô hình tóm tắt văn bản đề xuất**

Các bước chính của mô hình bao gồm:

* **Tiền xử lý văn bản:** Văn bản được chuẩn hóa và tách thành các câu riêng lẻ.
* **Tạo vector đại diện:** Sử dụng mô hình FastText để tạo vector đại diện cho từng từ, sau đó kết hợp với trọng số TF-IDF để tạo ra vector đại diện cho từng câu. Vector cuối cùng của câu là một vector có trọng số, kết hợp thông tin ngữ nghĩa (từ FastText) và mức độ quan trọng (từ TF-IDF) của các từ trong câu. Việc tính vector trung bình có trọng số cho toàn bộ câu theo các bước sau:
* Với mỗi từ trong câu, nhân vector FastText của từ đó với trọng số TF-IDF của nó.
* Tổng hợp các vector từ đã nhân với TF-IDF bằng cách cộng chúng lại.
* Cuối cùng, chia tổng vector này cho tổng trọng số TF-IDF của các từ trong câu để tạo ra vector trung bình có trọng số cho cả câu.
* **Phân cụm bằng K-Means:** Các vector đại diện cho các câu được phân thành các cụm khác nhau bằng mô hình K-Means, từ đó chọn ra những câu tiêu biểu nhất cho mỗi cụm.
* **Tạo tóm tắt:** Các câu tiêu biểu được sắp xếp theo thứ tự xuất hiện trong văn bản gốc để tạo ra bản tóm tắt cuối cùng.
* Sử dụng KMeans Clustering giúp phân nhóm các câu tương tự lại với nhau và chọn câu đại diện từ mỗi nhóm. Điều này có thể tạo ra tóm tắt cô đọng và chính xác hơn so với các phương pháp chỉ chọn câu đơn lẻ.
* Kết hợp trọng số TF-IDF để điều chỉnh độ quan trọng của từ và mô hình FastText để giữ ngữ nghĩa của từ. Sử dụng cả TF-IDF và FastText giúp mô hình không chỉ cân nhắc trọng số của từ mà còn giữ lại ngữ nghĩa từ, tạo ra các vector câu giàu thông tin hơn.

### 2.3.2 Tập dữ liệu

Tập dữ liệu sử dụng trong nghiên cứu này là VNDS (A Vietnamese Dataset for Summarization), một tập dữ liệu dành cho việc tóm tắt văn bản tiếng Việt, được tải về từ Kaggle.com. VNDS bao gồm các văn bản tiếng Việt đa dạng về nội dung, với bản tóm tắt chính thức đi kèm. Điều này tạo điều kiện thuận lợi cho việc huấn luyện và đánh giá các mô hình tóm tắt văn bản.

### 2.3.3 Phương pháp đánh giá

Để đánh giá hiệu quả của mô hình tóm tắt văn bản, chỉ số ROUGE (Recall-Oriented Understudy for Gisting Evaluation) được sử dụng. Cụ thể, các chỉ số ROUGE-1, ROUGE-2, và ROUGE-L được áp dụng để đo lường mức độ trùng khớp giữa các câu tóm tắt được mô hình tạo ra và các câu tóm tắt chính thức trong tập dữ liệu. ROUGE-1 đo lường mức độ trùng khớp về từ đơn, ROUGE-2 đo lường về cụm từ hai từ, và ROUGE-L đo lường chuỗi từ liên tiếp dài nhất giữa các câu tóm tắt.

### 2.3.4 Cấu trúc thí nghiệm

Quá trình thí nghiệm được thực hiện trên một hệ thống máy tính có cấu hình cao, sử dụng AWS để lưu trữ và xử lý dữ liệu lớn. Các bước trong thí nghiệm bao gồm:

* **Tiền xử lý dữ liệu:** Dữ liệu được tải xuống từ Kaggle, sau đó tiền xử lý và phân đoạn thành các câu.
* **Tạo embedding:** Quá trình tạo embedding từ các câu văn bản bằng cách sử dụng mô hình FastText và trọng số TF-IDF.
* **Huấn luyện mô hình KMeans:** Mô hình KMeans được huấn luyện với các số cụm khác nhau để tìm ra cấu trúc tốt nhất.
* **Đánh giá mô hình:** Kết quả tóm tắt được đánh giá bằng chỉ số ROUGE.

# CHƯƠNG 3: TỔNG KẾT VÀ THỰC NGHIỆM CHƯƠNG TRÌNH

## 3.1 Thực nghiệm chương trình.

### 3.1.1 Đọc dữ liệu

* Tải dữ liệu huấn luyện và kiểm tra từ các tệp CSV (Bộ dữ liệu**: VNDS: A Vietnamese Dataset for Summarization)** để chuẩn bị cho các bước xử lý tiếp theo.

### 3.1.2 Kiểm tra dữ liệu

* Kiểm tra kích thước (số dòng và cột) của dữ liệu huấn luyện và kiểm tra để đảm bảo dữ liệu được tải lên đúng cách.

### 3.1.3 Tạo token từ câu

* **Chuẩn hóa văn bản:** Chuyển văn bản về chữ thường, loại bỏ các ký tự không cần thiết (khoảng trắng thừa, ký tự đặc biệt, dấu câu).
* **Tách câu:** Sử dụng nltk.sent\_tokenize để tách văn bản thành các câu.
* **Xử lý từng câu:** Loại bỏ các ký tự không thuộc từ (sử dụng biểu thức chính quy re.sub) và chuẩn hóa khoảng trắng trong câu.
* Chuẩn hóa và tách câu từ văn bản gốc để chuẩn bị cho việc chuyển đổi câu thành vector.

### 3.1.4 Chuyển đổi câu thành vector

* **Tạo TF-IDF vectorizer:** Sử dụng TfidfVectorizer để tính trọng số TF-IDF cho từng từ trong câu, giúp xác định tầm quan trọng của từ đó trong ngữ cảnh của toàn bộ văn bản.
* **Chuyển đổi từ thành vector bằng FastText:** Mỗi từ sau khi được token hóa sẽ được chuyển thành một vector số học (chuỗi các số) bằng mô hình FastText.
* **Nhân vector từ với trọng số TF-IDF:** Mỗi vector từ được nhân với trọng số TF-IDF tương ứng của từ đó. Điều này giúp điều chỉnh độ quan trọng của từ trong câu. Những từ quan trọng hơn sẽ có trọng số TF-IDF cao hơn và sẽ có ảnh hưởng lớn hơn đến vector cuối cùng của câu.
* **Chuẩn hóa vector câu:** Sau khi tính toán xong, vector của câu được chuẩn hóa bằng cách chia cho tổng trọng số TF-IDF của các từ. Điều này đảm bảo rằng vector cuối cùng không quá phụ thuộc vào độ dài của câu, mà tập trung vào ý nghĩa tổng quát của câu.
* Biểu diễn các câu thành các vector số, trong đó mỗi vector phản ánh ý nghĩa ngữ nghĩa của câu dựa trên mô hình từ điển (FastText) và trọng số TF-IDF.

### 3.1.5 Huấn luyện mô hình Kmeans

* **Phân cụm các câu:** Sử dụng KMeans để phân cụm các câu thành 2, 3, hoặc 4 cụm, tùy thuộc vào độ phù hợp của từng số cụm.
* **Chọn câu tiêu biểu:** Tính trung bình các chỉ số của cụm và chọn câu gần nhất với tâm cụm để làm câu đại diện cho cụm đó.
* **Tạo tóm tắt:** Kết hợp các câu đại diện để tạo thành đoạn tóm tắt cho đoạn văn.
* Phân cụm các câu trong đoạn văn thành các nhóm tương tự để tóm tắt đoạn văn.

### 3.1.6 Lưu mô hình

* Lưu mô hình KMeans và các nhãn

## 3.2 Kết luận và hướng phát triển.

* Kết quả đánh giá mô hình:

|  |  |  |
| --- | --- | --- |
|  | 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 3. 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.**

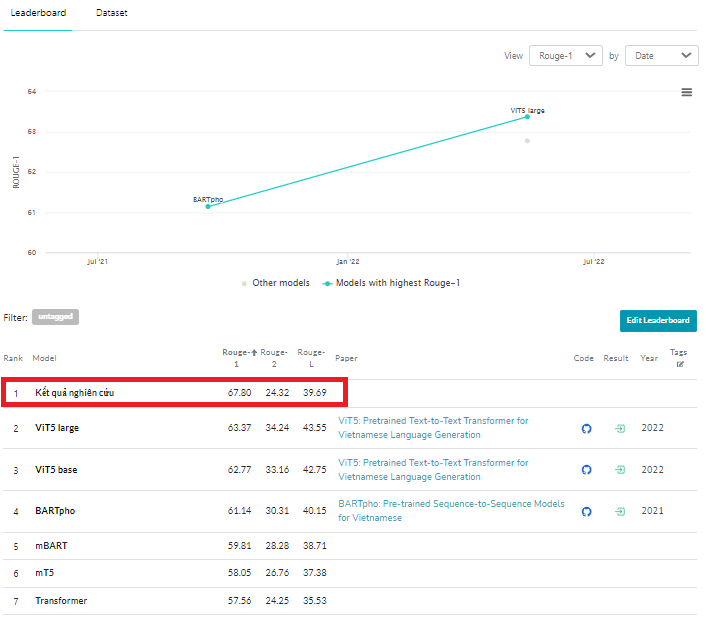
* **Điểm ROUGE-L**

**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.**

So sánh kết quả của mô hình đề xuất với các phương pháp của các tác giả khác trên cùng tập dữ liệu:

****

**Hình 3. 1: So sánh kết quả của mô hình đề xuất với các phương pháp của các tác giả khác trên cùng tập dữ liệu**

* **Kết luận:**

Mặc dù kết quả Rouge-1 khá 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.

Phương pháp được đề xuất 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 có khả năng tóm tắt văn bản tiếng Việt tốt.

* **Rouge-1** (67.8) cao hơn so với tất cả các mô hình khác, cho thấy mô hình có khả năng nắm bắt thông tin từ văn bản gốc rất tốt.
* **Rouge-2** (24.32) còn thấp, cho thấy mô hình chưa nắm bắt tốt mối quan hệ giữa các từ trong câu.
* **Rouge-L** (39.69) khá cao, cho thấy mô hình có khả năng giữ được cấu trúc dài hạn và độ mạch lạc của văn bản tốt.

**Phương pháp được đề xuất** đã cho thấy hiệu suất 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 được đề xuất có tiềm năng lớn trong việc tóm tắt văn bản tiếng Việt.

* **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.

* **Một vài kết quả khi sử dụng mô hình để thực hiện tóm tắt văn bản:**

**Ví dụ 1:**

**Bản gốc:**

**theo báo long\_an , những ngày gần đây , nông\_dân trồng chanh ở huyện bến\_lức , tỉnh long\_an rất phấn\_khởi vì chanh được\_mùa , trúng giá . hiện\_tại , thương\_lái đến tận vườn mua chanh với giá 24.000 đồng / kg . nhiều thương\_lái cho biết , giá chanh tăng cao do đang vào mùa nắng\_nóng ,**

**nhu\_cầu sử\_dụng chanh làm nước giải\_khát tăng , đặc\_biệt là thị\_trường tp. hcm . cụ\_thể , chanh không hạt và có hạt được các thương\_lái thu\_mua tại vườn với giá dao\_động từ 23.000-25.000 đồng / kg , chanh bông tím có giá 24.000 đồng / kg . “ thị\_trường xuất\_khẩu chanh năm nay rộng hơn so năm\_ngoái nên giá chanh tăng và giữ được mức cao trong thời\_gian dài " , ông tám\_sơn ( chủ vựa chanh xã bình\_đức , huyện bến\_lức ) cho biết . theo nhiều người trồng chanh huyện bến\_lức , bình\_quân mỗi ha đất trồng chanh đạt năng\_suất hơn 20 tấn / năm . với mức giá hiện\_tại , sau khi trừ chi\_phí, người trồng chanh lãi vài trăm triệu đồng / ha / năm . nhiều nông\_dân cho biết , những năm trước , giá chanh thường tăng cao vài ngày lại giảm . tuy\_nhiên , năm nay , do thời\_tiết nắng\_nóng kéo\_dài nên giá chanh vẫn ở mức cao . còn theo khảo\_sát của pv báo vnexpress , tại các chợ trên địa\_bàn tp. hcm , giá chanh còn tăng cao hơn cả cam\_sành . chị lan , tiểu\_thương tại chợ xóm mới ( quận gò\_vấp , tp. hcm ) cho biết , giá chanh gần đây có lúc lên tới 45.000 đồng / kg với chanh không hạt , 35.000 đồng với chanh có hạt . như\_vậy , giá đã tăng gấp 4 lần so với cuối năm 2017 . chị kể , nếu bán\_lẻ thì cũng thu 3.000 đồng mỗi quả . “ mỗi ngày tôi lấy khoảng 5-10 kg nhưng đến chiều đã gần hết . giá tăng cao nhưng nắng\_nóng nên khách mua nhiều " , chị lan giải\_thích . chị vân , thương\_lái ở huyện giồng\_trôm ( tỉnh bến\_tre ) chia\_sẻ với tờ vnexpress , năm nay trồng chanh có lãi hơn cả cam . bởi lẽ , một ký cam\_sành loại ngon mua tại vườn chỉ 15.000 - 20.000 đồng / kg , còn chanh lên đến 30.000 đồng / kg . ngọc\_lài ( tổng\_hợp ) chanh tăng\_giá , nông\_dân tích\_cực chăm\_sóc ( ảnh : báo\_long\_an )**

**Bản tóm tắt:**

**theo báo long\_an những ngày gần đây nông\_dân trồng chanh ở huyện bến\_lức tỉnh long\_an rất phấn\_khởi vì chanh được\_mùa trúng giá chị lan tiểu\_thương tại chợ xóm mới quận gò\_vấp tp chị kể nếu bán\_lẻ thì cũng thu 3 000 đồng mỗi quả bởi lẽ một ký cam\_sành loại ngon mua tại vườn chỉ 15 000 20 000 đồng kg còn chanh lên đến 30 000 đồng kg'**

**Ví dụ 2:**

**Bản gốc:**

**sáng 2-2, trong chuyến làm việc tại tp.hcm, chủ tịch nước võ văn thưởng đã đến thăm, tặng quà, chúc tết nsnd kim cương và gs.bs nguyễn chấn hùng. đi cùng chủ tịch nước võ văn thưởng còn có ông lê khánh hải - chủ nhiệm văn phòng chủ tịch nước, ông nguyễn hồ hải - phó bí thư thường trực thành ủy tp.hcm. trao đổi với nsnd kim cương, chủ tịch nước võ văn thưởng tri ân những cống hiến, quá trình lao động của nghệ sĩ kim cương cho nền văn hóa của đất nước. không chỉ vậy, dù tuổi đã cao, bà vẫn luôn chăm lo, động viên đời sống vật chất, tinh thần cho các nghệ sĩ sân khấu, nhất là nghệ sĩ cao tuổi có hoàn cảnh khó khăn, không nơi nương tựa. nghệ sĩ kim cương còn có nhiều hoạt động hỗ trợ người khuyết tật và trẻ em mồ côi của tp.chủ tịch nước võ văn thưởng chúc nsnd kim cương năm mới nhiều niềm vui, may mắn, tiếp tục sống vui sống khỏe để truyền lửa cho thế hệ trẻ. thông qua nsnd kim cương, chủ tịch nước đã gửi lời thăm hỏi đến các văn nghệ sĩ khác tại tp. ông khẳng định song song với phát triển kinh tế, nước ta luôn chú trọng phát triển lĩnh vực văn hóa. trong đó, nhờ hoạt động nghệ thuật, qua các vai diễn, các văn nghệ sĩ đã góp phần rất lớn trong giáo dục truyền thống dân tộc, tuyên truyền chính sách của đảng và nhà nước. đáp lời, nsnd kim cương xúc động khi được lãnh đạo nhà nước quan tâm, thăm hỏi, động viên. bà kỳ vọng thế hệ trẻ với tinh thần nhiệt quyết, năng động sẽ có nhiều chính sách sáng tạo phát triển đất nước**

**Bản tóm tắt:**

**trong đó nhờ hoạt động nghệ thuật qua các vai diễn các văn nghệ sĩ đã góp phần rất lớn trong giáo dục truyền thống dân tộc tuyên truyền chính sách của đảng và nhà nước thông qua nsnd kim cương chủ tịch nước đã gửi lời thăm hỏi đến các văn nghệ sĩ khác tại tp bà kỳ vọng thế hệ trẻ với tinh thần nhiệt quyết năng động sẽ có nhiều chính sách sáng tạo phát triển đất nước sáng 2 2 trong chuyến làm việc tại tp hcm chủ tịch nước võ văn thưởng đã đến thăm tặng quà chúc tết nsnd kim cương và gs bs nguyễn chấn hùng**

# 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.

**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.**

## Hạn chế đề tài

**Thiếu sự đa dạng trong kỹ thuật tóm tắt:**

* Mô hình tóm tắt hiện tại chủ yếu dựa vào việc chọn các câu gần với tâm cụm, điều này có thể dẫn đến sự thiếu đa dạng trong tóm tắt. Các câu được chọn có thể chưa bao quát hết nội dung chính của văn bản, đặc biệt là với các văn bản phức tạp.

**Hiệu quả tính toán**:

* Quá trình huấn luyện mô hình và tính toán embedding tiêu tốn khá nhiều tài nguyên và thời gian, đặc biệt khi xử lý các tập dữ liệu lớn. Điều này có thể hạn chế khả năng mở rộng của mô hình khi áp dụng trên quy mô lớn hơn hoặc trong môi trường thực tế.

**Tính tổng quát của mô hình**:

* Hiệu suất mô hình trên tập dữ liệu huấn luyện và kiểm tra là khá 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.

## 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. Abhishek Jain, (2024). *TF-IDF in NLP (Term Frequency Inverse Document Frequency),* Medium. Truy cập ngày 18 tháng 01 năm 2024.

<https://medium.com/@abhishekjainindore24/tf-idf-in-nlp-term-frequency-inverse-document-frequency-e05b65932f1d>.

1. Arif Romadhan. (2020). *Step by Step to Understanding K-means Clustering and Implementation with sklearn*, October 4,

Available:

https://towardsdatascience.com/step-by-step-to-understanding-k-means-clustering-and-implementation-with-sklearn-527447e0f967.

Truy cập ngày 18 tháng 8 năm 2024.

1. Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., & Kochut, K. (2017). *Text summarization techniques: A brief survey*, *International Journal of Advanced Computer Science and Applications (IJACSA)*, 8(10), pp. 397-405. DOI: 10.14569/IJACSA.2017.081052.
2. Chin-Yew Lin. *ROUGE - A Package for Automatic Evaluation of Summaries*, Information Sciences Institute, University of Southern California, 4676 Admiralty Way, Marina del Rey, CA 90292.
3. Dat Quoc Nguyen, Anh Tuan Nguyen. (2020). *PhoBERT: Pre-trained language models for Vietnamese*, in *Findings of EMNLP*, pp. 1037-1042. DOI: 10.18653/v1/2020.findings-emnlp.92.
4. Dr Inderjeet Mani , Mark T Maybury. (2018). *Advances in Automatic Text Summarization*. MIT Press.
5. Đinh Quang Thắng, Lê Hồng Phương, Nguyễn Thị Minh Huyền, Nguyễn Cẩm Tú, Mathias Rossignol, Vũ Xuân Lương. (2008). *Word segmentation of Vietnamese texts: a comparison of approaches*, in *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC)*.
6. Esra Kahya-Özyirmidokuz. (2014). “Mining Unstructured Turkish Economy News Articles,” *Procedia Economics and Finance* (16), 73-80.
7. Facebook AI Research. (n.d.). *fastText Vietnamese word vectors full*, Kaggle. Truy cập ngày 18 tháng 02 năm 2024.

<https://www.kaggle.com/datasets/aeryss/fasttext-vietnamese-word-vectors-full>.

1. Gipp, B., Ruas, T., & Wahle, J. P. (2023). Deep Learning for Natural Language Processing, Gipp Lab. Truy cập ngày 18 tháng 02 năm 2024. [https://gipplab.org/deep-learning-for-natural-language-processing/](https://gipplab.org/deep-learning-for-natural-language-processing/" \t "_new).
2. Hieu Nguyen, Long Phan, James Anibal, Alec Peltekian, and Hieu Tran. (2021). “Viesum: How robust are transformer-based models on Vietnamese summarization,”
3. Huy To Quoc, Kiet Van Nguyen, Ngan Luu-Thuy Nguyen, Anh Gia-Tuan Nguyen. (2021). “Monolingual vs multilingual BERTology for Vietnamese extractive multi-document summarization,” *Proceedings of the 35th Pacific Asia Conference on Language, Information and Computation* (692-699). Shanghai, China.
4. L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel. (2021). *mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer*, in *NAACL*.
5. Lê Hông Phuong, Nguyên Thi Minh Huyên, Azim Roussanaly & Hô Tuòng Vinh. (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, pp. 415-424. Springer, Berlin, Heidelberg.
6. Minh-Tien Nguyen, Hoang-Diep Nguyen, Thi-HaiNang Nguyen, & Van-Hau Nguyen. (2018). *Towards state-of-the-art baselines for Vietnamese multi-document summarization*, in *Proceedings of the 10th International Conference on Knowledge and Systems Engineering (KSE)*, pp. 85-90.
7. Merve Bayram Durna, M. B. (2024). *Advanced Word Embeddings: Word2Vec, GloVe, and FastText*, Medium. Truy cập ngày 18 tháng 03 năm 2024. Merve Bayram Durna, M. B. (2024). *Advanced Word Embeddings: Word2Vec, GloVe, and FastText*, Medium.

Truy cập ngày 18 tháng 03 năm 2024.

1. Nguyễn Thị Ngọc Tú, Nguyễn Thị Thu Hà, Lê Thanh Hương, Hồ Ngọc Vinh, Đào Thanh Tĩnh, & Nguyễn Ngọc Cương. (2015). “Ứng dụng mô hình đồ thị trong tóm tắt đa văn bản tiếng Việt.” Trong *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.
2. Nguyen, Dat Quoc & Nguyen, Anh. (2020). “PhoBERT: Pre-trained language models for Vietnamese,” *Findings of EMNLP* (1037-1042). DOI: 10.18653/v1/2020.findings-emnlp.92.
3. Nguyen, Van-Hau & Nguyen, Thanh-Chinh & Nguyen, Minh-Tien & Hoai, Nguyen. (2019). “VNDS: A Vietnamese Dataset for Summarization,” *Proceedings of the 2019 IEEE International Conference on Networking, Sensing and Control (ICNSC)*, 375-380. DOI: 10.1109/NICS48868.2019.9023886.
4. Phạm Hữu Quang. 2018. *Xây dựng mô hình không gian vector cho Tiếng Việt*. Truy cập ngày 18 tháng 8 năm 2024.

<https://viblo.asia/p/xay-dung-mo-hinh-khong-gian-vector-cho-tieng-viet-GrLZDXr2Zk0>.

1. Pham Hoang Anh. (2018). *Vietnamese doc summarization basic*, GitHub. Truy cập ngày 18 tháng 8 năm 2024.

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

1. Rothe, Sascha & Narayan, Shashi & Severyn, Aliaksei. (2020). *Leveraging Pre-trained Checkpoints for Sequence Generation Tasks*, *Transactions of the Association for Computational Linguistics*, 8, pp. 264-280. DOI: 10.1162/tacl\_a\_00313.
2. 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. *arXiv:1606.02960v2 [cs.CL]*.
3. Spatial, (2018), *Text Mining – Concepts techniques and workflows,* imspatial.com. Truy cập ngày 18 tháng 01 năm 2024.

<https://imspatial.com/text-mining-concepts-techniques-and-workflows/>.

1. Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean, (2013). “Efficient Estimation of Word Representations in Vector Space.” *arXiv:1301.3781v3 [cs.CL]*.
2. Thanh Vu, Dat Quoc Nguyen, Dai Quoc Nguyen, Mark Dras and Mark Johnson. (2018). *VnCoreNLP: A Vietnamese Natural Language Processing Toolkit*, in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations, NAACL 2018*, pp. 56-60.
3. Trương Quốc Định, Nguyễn Quang Dũng. (2012). “Một giải pháp tóm tắt văn bản tiếng Việt tự động.” Trong *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.
4. Tu-Anh Nguyen-Hoang, Khai Nguyen & Quang-Vinh Tran. (2012). “TSGVi: A graph-based summarization system for Vietnamese documents,” *Journal of Ambient Intelligence and Humanized Computing*, 3(4), 305-313. DOI: 10.1007/s12652-012-0143-x.
5. Viet Bui The, Oanh Tran Thi, Phuong Le-Hong. (2020). *Improving Sequence Tagging for Vietnamese Text using Transformer-based Neural Models*, in *PACLIC*.
6. Rong, X. (2016). *word2vec Parameter Learning Explained*, arXiv. Truy cập ngày 14 tháng 04 năm 2024. [https://arxiv.org/abs/1411.2738](https://arxiv.org/abs/1411.2738" \t "_new).
7. University of Cincinnati (n.d.). *K-means Cluster Analysis*, UC-R. Truy cập ngày 14 tháng 03 năm 2024.

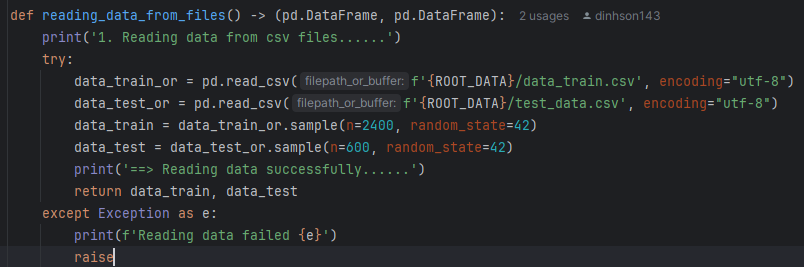
<https://uc-r.github.io/kmeans_clustering>.

# PHỤ LỤC

## Phụ lục 1: Mã Code Mô Hình Training

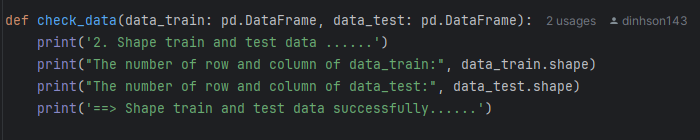
### Tiền xử lý văn bản.

* Đọc dữ liệu từ file CSV:



**Hình 2: 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 3: 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

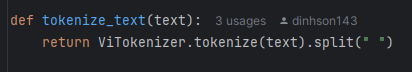


**Hình 4: Tách các câu trong văn bản**

* 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.

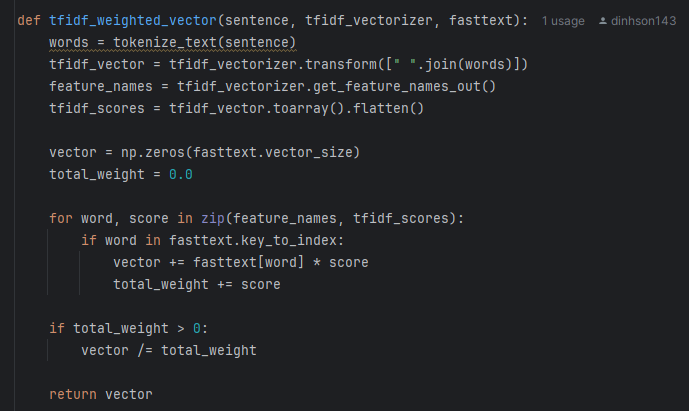
### Chuyển các câu sang vector.

* Hàm **tokenize\_text:**



**Hình 5: 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 6: Chuyển các câu sang vector - Tfidf\_weighted\_vector**

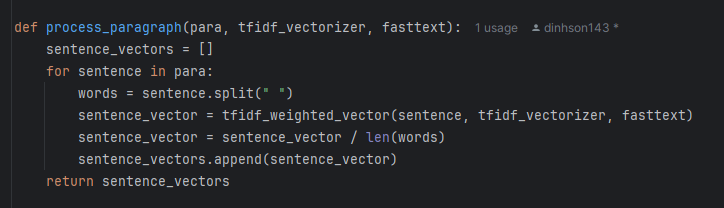
* tokenize\_text(sentence): Tokenize câu thành các từ.
* tfidf\_vectorizer.transform([" ".join(words)]): Tính toán vector TF-IDF cho câu.
* get\_feature\_names\_out(): Lấy danh sách các từ (features) trong TF-IDF.
* tfidf\_scores = tfidf\_vector.toarray().flatten(): Chuyển đổi vector TF-IDF thành mảng 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.

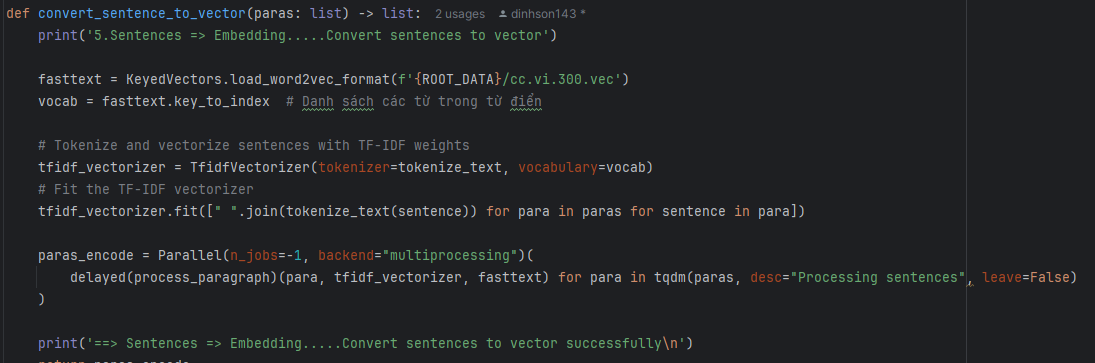
vector += fasttext[word] \* score: Cộng vector từ của từ với trọng số TF-IDF vào vector kết quả.

* + 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 7: Chuyển các câu sang vector - process\_paragraph**

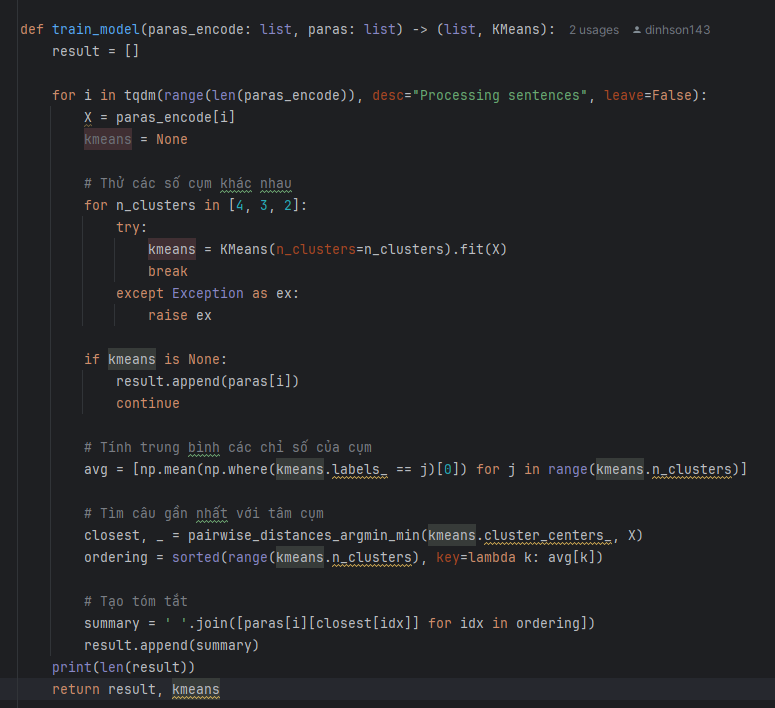
* sentence\_vectors = []: Khởi tạo danh sách để lưu các vector câu.
* for sentence in para: Duyệt qua từng câu trong đoạn văn bản.
  + words = sentence.split(" "): Tách câu thành các từ.
  + sentence\_vector=tfidf\_weighted\_vector (sentence, tfidf\_vectorizer, fasttext): Tính toán vector của câu bằng 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 8: Chuyển các câu sang vector - convert\_sentence\_to\_vector**

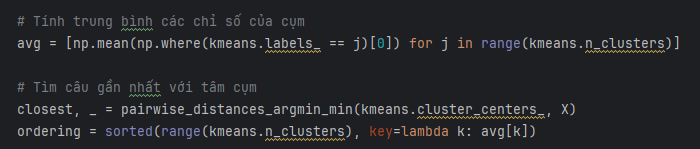
* fasttext=KeyedVectors.load\_word2vec\_format(f'{ROOT\_DATA}/cc.vi.300.vec'): Tải mô hình FastText từ file.
* vocab = fasttext.key\_to\_index: Lấy danh sách các từ trong từ điển FastText.
* tfidf\_vectorizer=TfidfVectorizer(tokenizer=tokenize\_text, vocabulary=vocab): Khởi tạo TF-IDF vectorizer với 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 9: Training model**

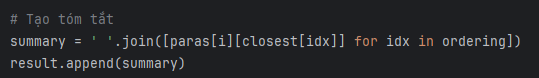
* Hàm này nhận đầu vào là danh sách các đoạn văn bản đã được mã hóa (paras\_encode) và danh sách các đoạn văn bản gốc (paras), sau đó trả về danh sách các câu đại diện cho mỗi đoạn văn bản và mô hình KMeans cuối cùng.



**Hình 10: 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 11: 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 12: 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.