ĐẠI HỌC QUỐC GIA TP.HCM
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH

BÁO CÁO THỰC TẬP 1

# Bài toán xác định nội dung trùng lặp


GVHD: PGS. TS. Quản Thành Thơ
HVTH: Lê Nguyên Khang - 2370618

Thành phố Hồ Chí Minh, 06 2024

# Lời cam đoan

Tôi xin cam đoan rằng, bài báo cáo thực tập 1 'Bài toán xác định nội dung trùng lặp' là sản phẩm nghiên cứu của tôi dưới sự hướng dẫn của thầy PGS. TS. Quản Thành Thơ, chú trọng vào việc giải quyết thách thức thực tiễn trong xử lý dữ liệu đầu vào cho Chatbot hỏi đáp closed-domain.

Ngoại trừ những thông tin tham khảo rõ ràng từ các công trình nghiên cứu khác, các nội dung trong luận văn là kết quả của quá trình nghiên cứu chủ thể của tôi và chưa từng được công bố trước đây dưới mọi hình thức.

Tôi chấp nhận hoàn toàn trách nhiệm về nội dung và chất lượng của luận văn. Mọi sáng tạo và kết quả đều xuất phát từ công lao và sự cố gắng không ngừng của chính tôi. Trong trường hợp có bất kỳ sự đạo văn hay vi phạm bản quyền nào, tôi xác nhận sẽ chịu trách nhiệm và đảm bảo sửa chữa ngay lập tức.

Tôi cam kết tuân thủ nguyên tắc đạo đức nghiên cứu và tuân thủ các quy định của Trường Đại học Bách khoa - Đại học Quốc gia TP.HCM. Bản cam kết này không làm ảnh hưởng đến uy tín của trường và không tạo ra bất kỳ vấn đề pháp lý nào liên quan đến việc sử dụng thông tin hay kết quả nghiên cứu của tôi.

TP Hồ Chí Minh, Tháng 6/2024

Tác giả

Lê Nguyên Khang

# Lời cảm ơn

Tôi xin gửi lời cảm ơn chân thành và tri ân nhất đến thầy PGS.TS Quản Thành Thơ, người đã dành thời gian và tâm huyết hướng dẫn tôi trong quá trình thực hiện đề tài. Sự đồng hành và sự tận tâm chỉ dẫn của thầy không chỉ giúp tôi có một cái nhìn toàn diện hơn về đề tài mà còn nâng cao chất lượng của công trình nghiên cứu.

Tôi muốn bày tỏ lòng biết ơn sâu sắc đến tất cả các thầy, cô và giảng viên Khoa Khoa học và Kỹ thuật Máy tính cũng như Trường Đại học Bách Khoa - Đại học Quốc gia TP.HCM. Kiến thức quý báu mà tôi đã được học từ các thầy, cô đã đóng góp quan trọng vào việc hoàn thành đề tài và phát triển năng lực chuyên môn.

Mặc dù đã cố gắng hết sức để hoàn thiện đề tài, tôi nhận thức rằng vẫn còn những hạn chế và thiếu sót. Tôi mong muốn nhận được những lời nhận xét, góp ý từ thầy cô và bạn bè để bài báo cáo này có thể ngày càng được hoàn thiện và phát triển.

# Tóm tắt

Hiện nay, với sự tiến bộ của các kỹ thuật Trí tuệ nhân tạo (Artificial Intelligence), sự phát triển của các hệ thống Chatbot thông minh ngày càng thu hút sự chú ý, đặc biệt là với tính hiệu quả của chúng trong việc thay thế con người ở nhiều lĩnh vực. Trong bối cảnh xu hướng hiện nay, người dùng có xu hướng ưa chuộng sự sử dụng ngôn ngữ tự nhiên bởi tính thân thiện và tính dễ sử dụng của nó. Tuy nhiên, xử lý dữ liệu đầu vào là một bài toán cần giải quyết trong các hệ thống Chatbot.

Để phục vụ cho chatbot hỏi đáp closed-domain, ví dụ ở trường Đại học Bách khoa Thành phố Hồ Chí Minh, các crawler đã cào dữ liệu từ các website của trường, của khoa và một số nơi khác liên quan tới lĩnh vực giáo dục. Tuy nhiên, các bài viết thường dài và nội dung trùng lặp lẫn nhau khá nhiều. Việc xử lý thủ công là không thể.

Đề tài này tập trung vào công việc tìm giải pháp để có thể tách các đoạn văn, bài viết dài thành các đoạn ngắn hơn, đồng thời kiểm tra và lưu một phiên bản duy nhất với cùng một nội dung trong cơ sở dữ liệu. Việc giảm thiểu thời gian xử lý và đảm bảo về mặt ngữ nghĩa là một trong những thách thức quan trọng được giải quyết.

# Mục lục

# Danh sách hình vẽ

# Danh sách bảng

# Chương 1

# Giới thiệu

Trong chương này, tôi sẽ giới thiệu tổng quan về nội dung của đề tài cùng với mục tiêu đề ra trong quá trình thực hiện đề tài. Ngoài ra, ta sẽ xác định các giai đoạn cụ thể trong quá trình thực hiện đề tài để đảm bảo sự tiến độ và đạt được kết quả như mong đợi. Các giai đoạn này sẽ được phân tích kỹ hơn để có thể phân bổ thời gian và nguồn lực phù hợp cho từng giai đoạn.

## 1.1 Đặt vấn đề

QA là một lĩnh vực nghiên cứu sôi động và nhiều hướng nghiên cứu. Nó là sự giao thoa kết hợp của Xử lý ngôn ngữ tự nhiên (NLP), Truy xuất thông tin (Information Retrieval - IR), Suy luận logic (Logical Reasoning), Biểu diễn tri thức (Knowledge Representation), Học máy máy (Machine Learning), Tìm kiếm Ngữ nghĩa (Semantic search).

CQAS là một hướng nghiên cứu quan trọng trong Xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP) và là hướng mở rộng hơn của bài toán QAS. Do đặc trưng linh hoạt của ngôn ngữ tự nhiên mà các câu hỏi rất không có cấu trúc đồng thời do sự mơ hồ trong chính câu hỏi có thể dẫn đến các câu trả lời sai, CQAS có thể giảm bớt sự nhập nhằng bằng cách đặt thêm một số câu hỏi phụ cho người hỏi để làm rõ hơn về ngữ cảnh. Cụ thể hơn, nhiệm vụ CQAS là xây dựng một phần mềm có thể tự trả lời một loạt câu hỏi bằng ngôn ngữ tự nhiên có tính liên kết với nhau xuất hiện trong một cuộc hội thoại. Nó cũng có thể duy trì một cuộc đối thoại mạch

lạc và phù hợp với người dùng, thay vì chỉ cung cấp các câu trả lời đứt đoạn.

Một trong những công việc cần thiết cho các công tác nghiên cứu QA và CQAS là thu thập dữ liệu đầu vào. Tại Trường Đại học Bách khoa Thành phố Hồ Chí Minh, để phục vụ cho chatbot hỏi đáp closed-domain, các crawler đã cào dữ liệu từ các website của trường, của khoa và một số nơi khác liên quan tới lĩnh vực giáo dục. Tuy nhiên, các bài viết thường dài và nội dung trùng lặp lẫn nhau khá nhiều. Việc xử lý thủ công là không thể.

Nhận thấy vấn đề trên, tôi quyết định thực hiện đề tài này nhằm tạo ra một giải pháp có khả năng giảm thiểu việc xử lý thủ công, đem lại khả năng phân tách cách văn bản, bài viết thành các đoạn văn ngắn hơn và đảm bảo lưu trữ một bản duy nhất để giúp việc tìm kiếm trở nên hiệu quả.

## 1.2 Các giai đoạn thực hiện

Với vấn đề và mục tiêu đã đặt ra bên trên, công trình nghiên cứu ban đầu sẽ được chia thành 4 giai đoạn chính:

- Giai đoạn 1: Thu thập và xử lý dữ liệu, văn bản đầu vào về giáo dục.
- Giai đoạn 2: Phân tích và thiết kế hệ thống.
- Giai đoạn 3: Xây dựng và thử nghiệm giải pháp trên tập dữ liệu về của Trường Đại học Bách khoa Thành phố Hồ Chí Minh.
- Giai đoạn 4: Hoàn thiện giải pháp với các chức năng ban đầu đã đặt ra và thực hiện kiểm thử, đánh giá cho hệ thống.

## 1.3 Phạm vi nghiên cứu

Trong bài nghiên cứu này, tôi sẽ thiết kế và xây dựng một giải pháp hướng tới giải quyết vấn đề như sau:

- Tôi đã thu thập và sử dụng dữ liệu cho đề tài từ các nguồn thông tin, các văn bản chính thống từ các trang web, hệ thống lưu trữ thuộc Trường Đại học Bách

Khoa - Đại học Quốc gia Thành phố Hồ Chí Minh. Các dữ liệu dùng để thử nghiệm trong bài báo cáo này thuộc các thể loại như thủ tục hành chính, các bài viết giới thiệu, các quy chế, quy định,...

- Giải pháp cung cấp khả năng phân chia một các văn bản dài thành các đoạn văn bản ngắn nhưng vẫn đảm bảo về mặt ngữ nghĩa.
- Xác định đoan văn này có trùng trong cơ sở dữ liệu đã có không. Việc lưu trữ một bản duy nhất sẽ giúp việc tìm kiếm và trả lời thông tin được hiệu quả.

## 1.4 Tổng quan về báo cáo

Có tất cả 5 chương được trình bày trong bài cáo báo đồ án này:

- Chương 1: Giới thiệu tổng quan về nội dung đề tài, mục tiêu và các giai đoạn đặt ra của đề tài.
- Chương 2: Trình bày các kiến thức nền tảng được nghiên cứu và sẽ sử dụng trong đề tài.
- Chương 3: Giới thiệu một số công trình liên quan đến đề tài.
- Chương 4: Trình bày về giải pháp mà tôi đã nghiên cứu và thực hiện đề tài này.
- Chương 5: Tổng kết, đánh giá giải pháp và đề ra hướng phát triển cho các giai đoạn tiếp theo trong tương lai.

# Chương 2

# Kiến thức nền tảng

Để nghiên cứu và triển khai đề tài, cần tìm hiểu về mô hình phân loại chủ đề và các kỹ thuật so sánh các phân phối, so sánh vector.

## 2.1 Mô hình phân bố Dirichlet tiềm ẩn

Latent Dirichlet Allocation (LDA) Model[1] - Mô hình phân bố Dirichlet tiềm ẩn là lớp mô hình sinh (generative model) cho phép xác định một tợp hợp các chủ đề tưởng tượng (imaginary topics) mà mỗi topic sẽ được biểu diễn bởi tập hợp các từ. Mục tiêu của LDA là mapping toàn bộ các văn bản sang các topics tương ứng sao cho các từ trong mỗi một văn bản sẽ thể hiện những topic tưởng tượng ấy.

Một số định nghĩa trong mô hình LDA:

- Từ (Word): là đơn vị cơ bản nhất trong mô hình LDA. Trong ngữ cảnh LDA, một từ được biểu diễn bằng một chỉ số index trong từ điển.Một từ thứ $w_i$ trong từ điển được biểu diễn dưới dạng một vector one-hot $\mathbf{w}_i$ sao cho phần tử thứ $i$ của vector bằng 1 và các phần tử còn lại bằng 0.
  - **Ví dụ:** Với từ điển gồm 5 từ, nếu từ "apple"có chỉ số là 2, thì vector one-hot của "apple"là $\mathbf{w}_{apple} = [0, 1, 0, 0, 0]$.
- Văn bản (Document): là một tập hợp của $N$ từ được ký hiệu bởi $\mathbf{d} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_N\}$. Mỗi vector $\mathbf{w}_i$ đại diện cho một từ trong văn bản.

  - **Ví dụ:** Một văn bản có thể là: "apple dog apple cat", được biểu diễn bởi 4 vector one-hot tương ứng với các từ trong văn bản.

- Bộ văn bản (Corpus): là một tập hợp của $M$ văn bản ký hiệu bởi $\mathcal{D} = \{\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_M\}$.

  - **Ví dụ:** Một bộ văn bản gồm 3 văn bản có thể là:

    1. "apple dog apple cat"
    2. "banana orange dog"
    3. "dog cat fish"

    Ký hiệu: $\mathcal{D} = \{\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3\}$.

- Topic ẩn (Latent Topic): là những chủ đề ẩn được xác định dựa trên phân phối của các từ và làm trung gian để biểu diễn các văn bản theo chủ đề. Số lượng chủ đề ẩn được xác định trước và ký hiệu là $K$. Mỗi chủ đề $k$ được biểu diễn bởi một phân phối xác suất trên các từ trong từ điển. Chủ đề $k$ có phân phối $\beta_k = (\beta_{k1}, \beta_{k2}, \ldots, \beta_{kV})$, trong đó $\beta_{kv}$ là xác suất của từ $v$ trong chủ đề $k$.

  - **Ví dụ:** Với $K = 2$, có thể có các chủ đề như sau:
    * Chủ đề 1: { "apple": 0.4, "orange": 0.3, "banana": 0.3 }
    * Chủ đề 2: { "dog": 0.5, "cat": 0.3, "fish": 0.2 }

Các tham số mô hình:

- Số lượng chủ đề $K$: là một tham số được xác định trước và biểu diễn số lượng chủ đề mà mô hình sẽ tìm ra từ tập dữ liệu, nhằm xác định số lượng chủ đề giúp mô hình phân loại và nhóm các từ trong các văn bản thành $K$ chủ đề ẩn.

- Tham số Dirichlet $\alpha$: là một vector của các giá trị siêu tham số (hyperparameter) cho phân phối Dirichlet trên các chủ đề cho mỗi văn bản, nhằm kiểm soát mức độ hỗn hợp của các chủ đề trong mỗi văn bản. Giá trị lớn của $\alpha$ sẽ làm cho các văn bản có nhiều chủ đề, trong khi giá trị nhỏ của $\alpha$ sẽ làm cho mỗi văn bản chỉ chứa một vài chủ đề.

- Tham số Dirichlet $\eta$: là một vector của các giá trị siêu tham số cho phân phối Dirichlet trên các từ cho mỗi chủ đề, nhằm kiểm soát mức độ hỗn hợp của các từ trong mỗi chủ đề. Giá trị lớn của $\eta$ sẽ làm cho các chủ đề chứa nhiều từ, trong khi giá trị nhỏ của $\eta$ sẽ làm cho mỗi chủ đề chỉ chứa một vài từ.

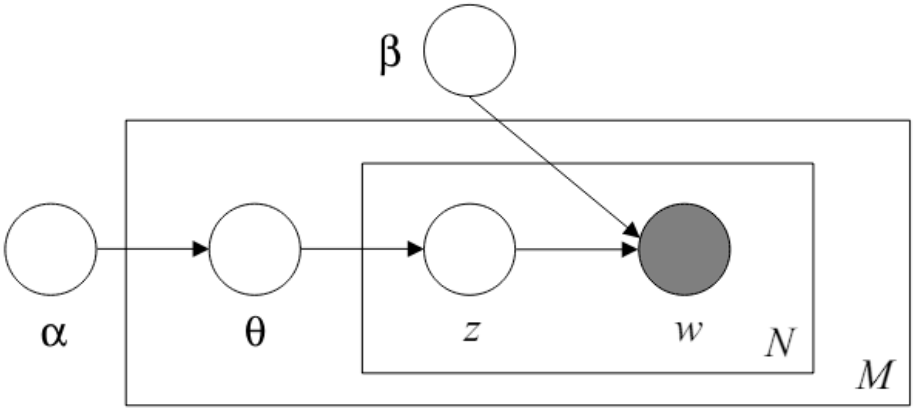
- Phân phối chủ đề $\theta$: là phân phối xác suất trên các chủ đề cho văn bản $d$. Phân phối này cho biết xác suất của mỗi chủ đề xuất hiện trong một văn bản cụ thể. Được sinh ra từ phân phối Dirichlet với tham số $\alpha$.

- Phân phối từ $\beta$: là phân phối xác suất trên các từ cho chủ đề $k$. Phân phối này cho biết xác suất của mỗi từ xuất hiện trong một chủ đề cụ thể. Được sinh ra từ phân phối Dirichlet với tham số $\eta$.

- Biến chủ đề $z$: là biến ẩn đại diện cho chủ đề của từ $n$ trong văn bản $d$. Biến này cho biết từ nào thuộc về chủ đề nào trong một văn bản cụ thể.

- Từ $w$: Từ $w_{dn}$ là từ thứ $n$ trong văn bản $d$. Đây là các từ quan sát được trong văn bản, là dữ liệu đầu vào của mô hình.

Quá trình sinh mô hình LDA gồm hai bước chính: bước sinh dữ liệu và bước suy luận.




**Hình 2.1:** Biểu diễn quá trình sinh của mô hình LDA. Vòng bên ngoài tượng trưng cho tài liệu, trong khi vòng bên trong tượng trưng cho sự lựa chọn lặp lại chủ đề và từ ngữ trong tài liệu.

**Bước sinh dữ liệu (Generative Process):** Quá trình sinh dữ liệu giả định cách các văn bản được tạo ra từ các chủ đề ẩn như sau:

- **Bước 1:** Với mỗi chủ đề $k$, chọn một phân phối xác suất trên các từ từ phân phối Dirichlet $\beta_k \sim \text{Dirichlet}(\eta)$.
- **Bước 2:** Với mỗi văn bản $d$ trong bộ văn bản:
  - Chọn một phân phối xác suất trên các chủ đề từ phân phối Dirichlet $\theta_d \sim \text{Dirichlet}(\alpha)$.
  - Với mỗi từ thứ $n$ trong văn bản $d$:
    - Chọn một chủ đề $z_{dn} \sim \text{Multinomial}(\theta_d)$.
    - Chọn một từ $w_{dn}$ từ phân phối xác suất của chủ đề $z_{dn}$: $w_{dn} \sim \text{Multinomial}(\beta_{z_{dn}})$.

**Bước suy luận (Inference Process):** Trong bước này, chúng ta suy luận ra các phân phối ẩn (ẩn chủ đề của mỗi từ và phân phối các chủ đề trong mỗi văn bản) từ dữ liệu quan sát được (các văn bản và từ trong chúng). Mục tiêu là tìm ra các giá trị tốt nhất cho các tham số $\theta$, $\beta$, và $z$.

- **Phương pháp suy luận:**
  - **Gibbs Sampling:** Một phương pháp lấy mẫu tuần tự để ước lượng các phân phối ẩn.
  - **Variational Inference:** Một phương pháp tối ưu hóa để xấp xỉ các phân phối ẩn.

Mô hình LDA (Latent Dirichlet Allocation) cung cấp các kết quả chính sau:

- Phân phối chủ đề cho mỗi văn bản ($\theta$): cho biết mức độ mà mỗi chủ đề đóng góp vào văn bản $d$. **Ví dụ:** Giả sử có 3 chủ đề (K=3) và một văn bản $d$ có phân phối chủ đề $\theta_d = [0.2, 0.5, 0.3]$. Điều này có nghĩa là chủ đề 1 đóng góp 20%, chủ đề 2 đóng góp 50%, và chủ đề 3 đóng góp 30% vào văn bản $d$.

- Phân phối từ cho mỗi chủ đề ($\beta$): Vector $\beta_k$ cho biết xác suất của mỗi từ trong từ điển xuất hiện trong chủ đề $k$. **Ví dụ:** Giả sử từ điển có 5 từ và chủ đề $k$ có phân phối từ $\beta_k = [0.1, 0.2, 0.3, 0.1, 0.3]$. Điều này có nghĩa là xác suất của từ thứ nhất trong chủ đề $k$ là 10%, từ thứ hai là 20%, và tương tự cho các từ còn lại.

- Gán chủ đề cho các từ trong văn bản ($z$): Biến $z_{dn}$ cho biết từ $w_{dn}$ thuộc về chủ đề nào trong văn bản $d$. **Ví dụ:** Giả sử văn bản $d$ có các từ {w1, w2, w3} và các biến chủ đề tương ứng là {z1, z2, z3}, với $z1 = 2$, $z2 = 1$, và $z3 = 3$. Điều này có nghĩa là từ thứ nhất thuộc về chủ đề 2, từ thứ hai thuộc về chủ đề 1, và từ thứ ba thuộc về chủ đề 3.

- Tập hợp các chủ đề: giúp hiểu rõ hơn về các nội dung chính được phân tích từ tập dữ liệu. **Ví dụ:** Chủ đề 1 có thể bao gồm các từ như {"apple"(0.4), "orange"(0.3), "banana"(0.3)}, chủ đề 2 có thể bao gồm các từ như {"dog"(0.5), "cat"(0.3), "fish"(0.2)}.

## 2.2 Divergence Kullback-Leibler

Divergence Kullback-Leibler[1](KL divergence) là một khái niệm quan trọng trong lý thuyết thông tin và xác suất thống kê. Nó đo lường sự khác biệt giữa hai phân phối xác suất. Đặc biệt, KL divergence được sử dụng để đánh giá mức độ mà một phân phối xác suất $Q$ lệch khỏi một phân phối xác suất tham chiếu $P$.

Toán tử KL divergence giữa hai phân phối xác suất $P$ và $Q$ được định nghĩa như sau:

$$D_{\mathrm{KL}}(P\|Q) = \sum_{x \in X} P(x) \log \frac{P(x)}{Q(x)}$$

trong trường hợp phân phối rời rạc, hoặc:

$$D_{\mathrm{KL}}(P\|Q) = \int_{-\infty}^{\infty} P(x) \log \frac{P(x)}{Q(x)}\, dx$$

[1]Kullback–Leibler divergence `https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence`

trong trường hợp phân phối liên tục.
Trong đó:

- $P(x)$: Xác suất của $x$ theo phân phối $P$.
- $Q(x)$: Xác suất của $x$ theo phân phối $Q$.
- $X$: Không gian mẫu (tất cả các giá trị có thể của biến ngẫu nhiên).

Đặc điểm chính:

- *Không đối xứng*: KL divergence không đối xứng, tức là $D_{\mathrm{KL}}(P\|Q) \neq D_{\mathrm{KL}}(Q\|P)$.
- *Không phải là một khoảng cách*: Mặc dù KL divergence đo lường sự khác biệt giữa hai phân phối, nó không thỏa mãn các điều kiện của một khoảng cách trong không gian hình học (ví dụ, tính đối xứng và bất đẳng thức tam giác).
- *Không âm*: $D_{\mathrm{KL}}(P\|Q) \geq 0$, và chỉ bằng 0 khi $P$ và $Q$ là giống nhau trên toàn bộ không gian mẫu.

## 2.3 Độ đo cosine

*Độ tương tự cosine* đo lường độ tương tự giữa hai vector của không gian tích bên trong. Độ đo cosine được xác định bằng cosine của góc giữa hai vectơ và xác định xem hai vectơ có hướng gần giống nhau hay không và thường được sử dụng để đo độ tương tự của tài liệu trong phân tích văn bản.[2]

Trong phân tích văn bản, các vector thường được sử dụng để biểu diễn các tài liệu hoặc các đoạn văn bản. Mỗi chiều của vector thường tương ứng với một từ hoặc thuật ngữ cụ thể, và giá trị ở mỗi chiều có thể biểu diễn tần suất hoặc mức độ quan trọng của từ đó trong tài liệu.

Độ tương đồng cosine đo lường cosine của góc giữa hai vector. Nếu các vector cùng hướng, góc giữa chúng là 0 độ và cosine của 0 độ là 1. Nếu các vector vuông góc với nhau (góc 90 độ), cosine là 0, cho thấy không có sự tương đồng. Nếu các vector hướng ngược chiều nhau, cosine là -1, cho thấy hoàn toàn không tương đồng.

Độ tương đồng cosine giữa hai vector $\mathbf{A}$ và $\mathbf{B}$ được cho bởi:

$$\text{cosine similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|}$$

trong đó:

- $\mathbf{A} \cdot \mathbf{B}$ là tích vô hướng của các vector.
- $\|\mathbf{A}\|$ và $\|\mathbf{B}\|$ là độ lớn (hoặc độ dài) của các vector.

Một trong những ưu điểm chính của độ tương đồng cosine là nó không bị ảnh hưởng bởi độ dài của các vector. Điều này có nghĩa là không quan trọng các vector dài bao nhiêu (chứa bao nhiêu từ), nó chỉ xem xét hướng của các vector. Điều này làm cho nó đặc biệt hữu ích khi so sánh các tài liệu có độ dài khác nhau.

# Chương 3

# Công trình liên quan

Sau khi tiến hành tìm hiểu và nghiên cứu, tôi đã tìm được một vài ứng dụng và kỹ thuật có tính ứng dụng và cách hoạt động có thể giúp ích cho hướng nghiên cứu của tôi.

## 3.1 BERT

BERT (Bidirectional Encoder Representations from Transformers)[3] là một mô hình học sâu được phát triển bởi Google và được công bố vào năm 2018. BERT là một phần của họ mô hình Transformers, và nó đã tạo ra bước tiến lớn trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP).

Đặc điểm chính của BERT:

- **Bidirectional**: Không giống như các mô hình trước đó chỉ xử lý văn bản từ trái sang phải hoặc từ phải sang trái, BERT xử lý văn bản theo cả hai chiều. Điều này cho phép BERT nắm bắt được ngữ cảnh đầy đủ của một từ dựa trên cả văn bản trước và sau từ đó.
- **Pre-training và Fine-tuning**: BERT được huấn luyện trước (pre-trained) trên một lượng lớn dữ liệu văn bản từ Wikipedia và BooksCorpus, sau đó có thể được tinh chỉnh (fine-tuned) cho các nhiệm vụ cụ thể như phân loại văn bản, trả lời câu hỏi, và nhiều nhiệm vụ khác.

- **Transformers**: BERT dựa trên kiến trúc Transformer, một kiến trúc mạng neural mạnh mẽ cho phép mô hình xử lý mối quan hệ giữa các từ trong một câu mà không cần đến sự phụ thuộc tuần tự như các mô hình Recurrent Neural Network (RNN).

PhoBERT [4] là một mô hình ngôn ngữ dựa trên BERT được phát triển đặc biệt cho tiếng Việt. Giống như BERT, PhoBERT cũng sử dụng kiến trúc Transformer và kỹ thuật học sâu để nắm bắt ngữ cảnh của từ trong câu, nhưng nó được huấn luyện trên một lượng lớn dữ liệu tiếng Việt.

Đặc điểm chính của PhoBERT:

- **Ngôn ngữ đặc thù**: PhoBERT được phát triển riêng cho tiếng Việt, tận dụng các đặc trưng ngữ pháp và từ vựng của tiếng Việt để cải thiện hiệu suất cho các tác vụ NLP liên quan đến tiếng Việt.
- **Corpus huấn luyện**: PhoBERT được huấn luyện trên một lượng lớn dữ liệu văn bản tiếng Việt từ nhiều nguồn khác nhau như báo chí, sách, và các trang web. Điều này giúp mô hình hiểu rõ hơn về ngữ cảnh và ý nghĩa của từ trong tiếng Việt.
- **Kiến trúc**: PhoBERT sử dụng kiến trúc BERT cơ bản, bao gồm các tầng encoder của Transformer, giúp mô hình học được mối quan hệ giữa các từ trong câu theo cả hai chiều (trái sang phải và phải sang trái).

## 3.2 Underthesea Toolkit

Underthesea[1]là bộ dữ liệu module Python nguồn mở và các hướng dẫn hỗ trợ nghiên cứu và phát triển về Xử lý ngôn ngữ tự nhiên tiếng Việt. Nó cung cấp API cực kỳ dễ dàng để nhanh chóng áp dụng các mô hình NLP đã được huấn luyện trước cho văn bản tiếng Việt của bạn, chẳng hạn như phân đoạn từ, gắn thẻ một phần giọng nói (PoS), nhận dạng thực thể được đặt tên (NER), phân loại văn bản và phân tích cú pháp phụ thuộc.

Underthesea được hỗ trợ bởi một trong những thư viên học sâu phổ biến nhất, Pytorch, giúp nó dễ dàng train các mô hình học sâu và thử nghiệp các phương pháp tiếp cận mới bằng cách sử dụng các Module và Class của Underthesea.

Underthesea được công bố theo giấy phép GNU General Public License v3.0. Các quyền của giấy phép này có điều kiện là cung cấp mã nguồn hoàn chỉnh của các tác phẩm được cấp phép và sửa đổi, bao gồm các tác phẩm lớn hơn sử dụng tác phẩm được cấp phép, theo cùng một giấy phép.

[1]Underthesea `https://github.com/undertheseanlp/underthesea`

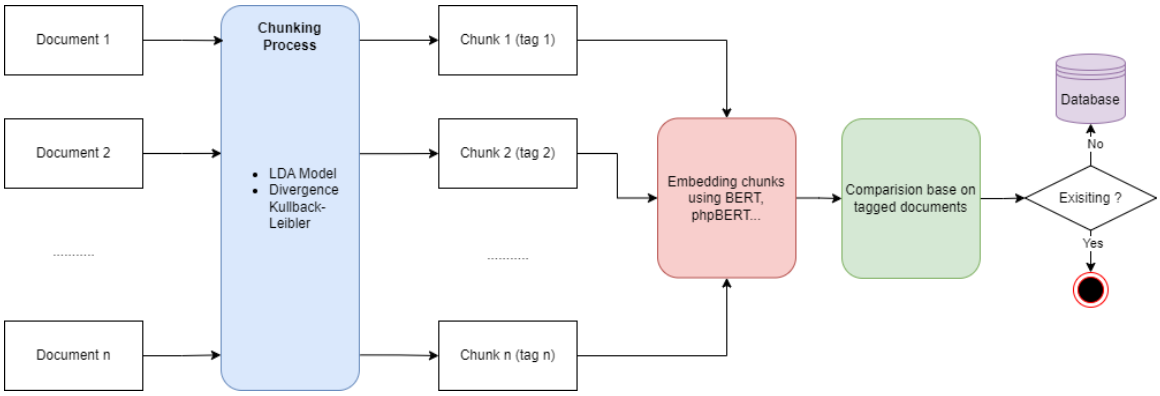
# Chương 4

# Giải pháp thực hiện đề tài

Trong chương này, tôi sẽ trình hướng giải quyết bày bài toán xác định nội dung trùng lặp. Tôi sẽ giới thiệu các thách thức và vấn đề cần giải quyết, cũng như các giải pháp được đề xuất để đạt được mục tiêu đề ra.

## 4.1 Tổng quan về kiến trúc

Để giải quyết tác vụ chia các bài viết dài thành các đoạn ngắn nhưng vẫn phải đảm bảo đầy đủ ngữ nghĩa, đồng thời xác định đoạn văn này có trùng trong cơ sở dữ liệu đã có hay không để đảm bảo việc lưu trữ một bản duy nhất sẽ giúp việc tìm kiếm và trả lời thông tin được hiệu quả, tôi đề xuất một cách tiếp cận được biểu diễn ở hình bên dưới.




**Hình 4.1:** Chiến lược theo dõi sự thay đổi chủ đề văn bản

Đầu tiên, với một đoạn văn bản dài, tôi dùng phương pháp phân tách đoạn dựa trên sự thay đổi của chủ đề để chia văn bản thành các phần, các đoạn ngắn (chunk). Sau đó với mỗi đoạn văn thu được, tôi tiến hành gán nhãn và embed đoạn văn thành vector. Cuối cùng, tôi thực hiện công tác so sánh các vector vừa thu được với các vector có trong cơ sở dữ liệu dựa trên các nhãn tương ứng, và thực hiện lưu văn bản vào cơ sở dữ liệu nếu đoạn văn chưa tồn tại (về mặt ngữ nghĩa, nội dung).

## 4.2 Phân đoạn văn bản

### 4.2.1 Một số phương pháp phân đoạn văn bản

Phân đoạn là một quá trình nhằm mục đích tách một văn bản dài thành nhiều phần nội dung ngắn nhiễu nhất có thể trong khi vẫn duy trì mức độ liên quan về mặt ngữ nghĩa. Quá trình này đặc biệt hữu ích trong tìm kiếm ngữ nghĩa, trong đó mỗi tài liệu chứa thông tin có giá trị về một chủ đề cụ thể.

Một số phương pháp có thể dùng để phân đoạn văn bản như:

- Phân đoạn theo kích thước cố định:
  - Là một cách tiếp cận đơn giản để phân đoạn văn bản, chia văn bản thành các phần có kích thước cố định được coi là các khối. Trong phương pháp này, văn bản được phân chia dựa trên số lượng ký tự hoặc câu, giúp việc triển khai trở nên đơn giản.
  - Tuy nhiên, phương pháp này bộc lộ những hạn chế nhất định. Một nhược điểm đáng kể là thiếu kiểm soát chính xác kích thước ngữ cảnh. Tính chất nghiêm ngặt và có kích thước cố định có thể dẫn đến việc cắt các từ, câu hoặc đoạn văn ở giữa, điều này có thể cản trở khả năng hiểu và làm gián đoạn luồng thông tin.
  - Hơn nữa, phương pháp này không tính đến ngữ nghĩa, không đảm bảo rằng đơn vị ngữ nghĩa của văn bản nắm bắt một ý tưởng hoặc suy nghĩ nhất định sẽ được gói gọn một cách chính xác trong một đoạn. Do đó, một đoạn có thể không khác biệt về mặt ngữ nghĩa với một đoạn khác.
- Phân tách nhận biết cấu trúc đệ quy:
  - Là một cách tiếp cận kết hợp để phân đoạn văn bản, kết hợp các phần tử của phương pháp cửa sổ trượt có kích thước cố định và phương pháp phân tách nhận biết cấu trúc. Phương pháp này cố gắng tạo ra các khối có kích thước gần như cố định, bằng ký tự hoặc mã thông báo, đồng thời cố gắng giữ nguyên các đơn vị văn bản gốc như từ, câu hoặc đoạn văn.

– Trong phương pháp này, văn bản được phân chia đệ quy bằng cách sử dụng nhiều dấu phân cách khác nhau chẳng hạn như ngắt đoạn, dòng mới hoặc dấu cách, chỉ chuyển sang mức độ chi tiết tiếp theo khi cần thiết. Điều này cho phép phương pháp cân bằng nhu cầu về kích thước khối cố định với mong muốn tôn trọng ranh giới ngôn ngữ tự nhiên của văn bản.

– Tuy nhiên, phương pháp này đòi hỏi văn bản có cấu trúc tốt và không phù hợp với văn bản có sự phân chia cấu trúc không nhất quán hoặc không rõ ràng. Đồng thời, các đoạn văn được tách đôi khi cũng có thể quá dài không phù hợp với yêu cầu ban đầu.

### 4.2.2 Phương pháp phân đoạn dựa trên thay đổi của chủ đề đoạn văn

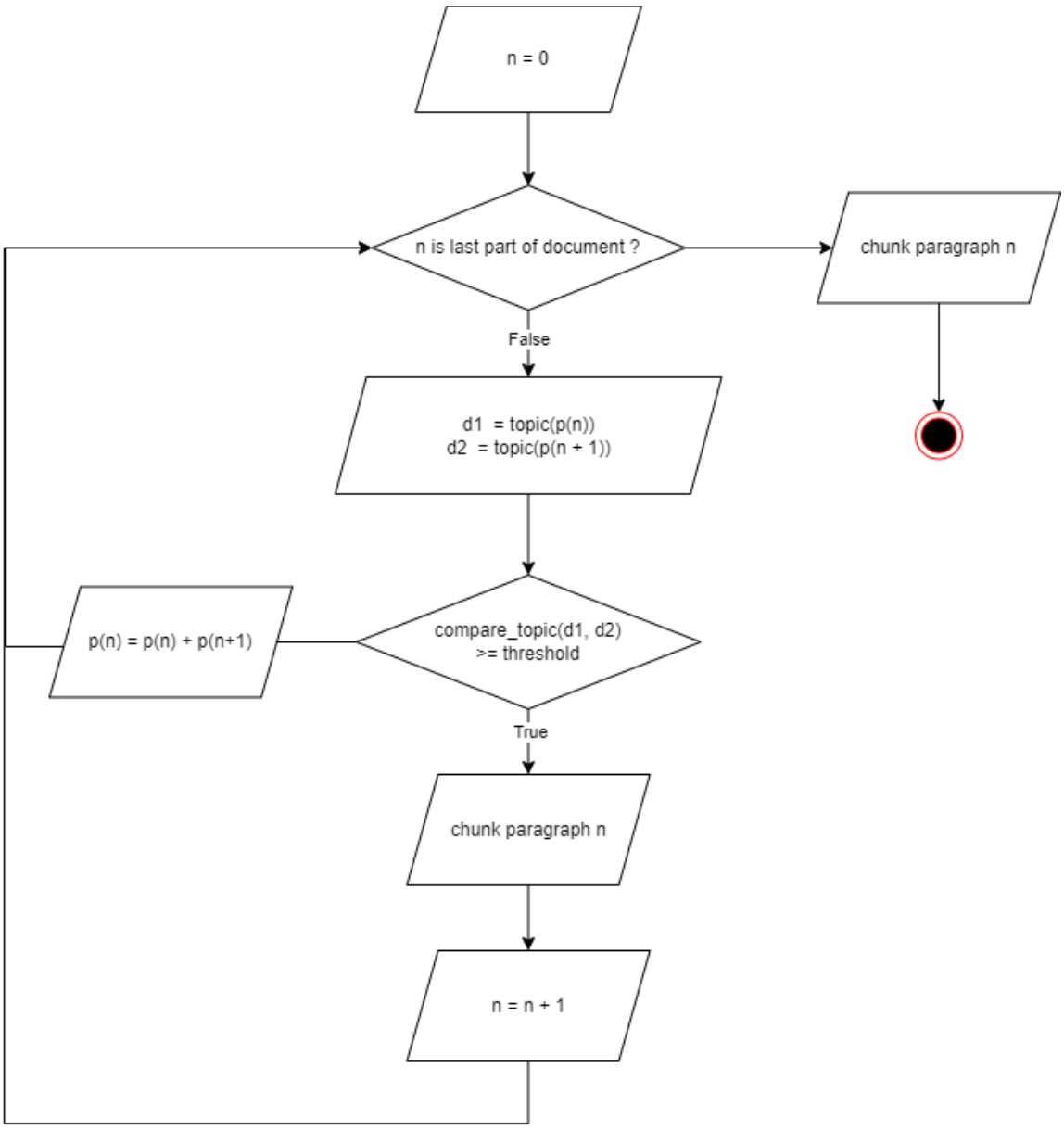
Với các cách tiếp cận đã trình bày trên, mỗi cách đều có ưu và nhược điểm nhất định, nhưng tóm chung lại để thực hiện công việc phân tách có thể chưa cho ra được các đoạn văn có chất tượng tốt, nhất là việc đảm bảo, duy trì tương đối về mặt ngữ nghĩa. Vì vậy, tôi đề xuất một phương pháp phân đoạn dựa trên sự thay đổi chủ đề giữa các phần văn bản.

Để tiến hành phân tách văn bản thành các đoạn văn ngắn dựa trên sự thay đổi chủ để, tôi sử dụng một cửa sổ trượt (slide window) để lần lượt trượt qua toàn bộ văn bản. Kích thước của cửa sổ trượt có thể là 2, 3, 4 ,... câu nhưng thường sẽ không có kích thước quá lớn. Thông thường, một đoạn văn tốt sẽ có kích thước từ 3 đến 10 câu[1].

Với mỗi phần văn bản được trượt qua, tôi sẽ tiến hành đánh giá sự khác nhau giữa chủ đề giữa phần hiện tại và phần trước đó (hoặc các phần trước đó). Nếu chủ đề giữa hai phần có sự khác nhau (vượt ngưỡng chỉ định) thì tiến hành tách đoạn tại điểm hiện tại của cửa sổ trượt. Ngược lại, nếu chủ đề giữa hai phần không có sự khác biệt tương đối, phần văn bản hiện tại ở cửa sổ trượt sẽ được kết hợp với đoạn trước đó để so sánh với đoạn tiếp theo.

Giải thuật phân tách đoạn theo chủ đề được thể hiện ở hình bên dưới (Hình 4.2).

[1]Paragraphs - Writing Guide `https://www.usu.edu/markdamen/writingguide/15paragr.htm`




**Hình 4.2:** Giải thuật phân tách đoạn theo chủ đề

Trong đó:

- p(n) là phần văn bản thứ n được theo dõi bởi cửa sổ trượt.
- d(i) là phân phối chủ đề của văn bản thứ i.

Vì đơn vị của cửa sổ trượt là câu nên tôi đã kết hợp với một số công cụ hỗ trợ để phân tách văn bản thành cách câu trong quá trình xử lý. Một số công cụ hỗ trợ tôi đã sử dụng như Underthesea Toolkit...

Ở đây, tôi sử dụng LDA model được trình bày ở mục 2.1 để xác định phân phối chủ đề của hai phần văn bản được phân chia bởi cửa sổ trượt vừa đề cập ở trên, sau đó dùng độ đo Kullback-Leibler trình bày ở mục 2.2 để tiến hành đánh giá sự khác biệc giữa hai phân phối chủ đề văn bản vừa tìm được.

## 4.3 Paragraph Embedding

Sau khi thực hiện quá trình phân tách đoạn, tôi sử dụng BERT để vector hóa các cho các đoạn văn bản tiếng Anh và phoBERT để vector hóa cho các đoạn văn bản sử dụng tiếng Việt.

## 4.4 Gán nhãn các đoạn văn bản đã được xử lý

Gán nhãn cho đoạn văn là một công việc cần thiết được thực hiện sau khi có các đoạn văn ngắn (chunk). Mục đích tôi gán nhãn cho các chunk để tiện cho việc truy vấn sau này, đồng thời giảm thời gian xử lý cho quá trình so sánh các chunk. Bài viết này sẽ không tập trung vào nghiên cứu chuyên sâu cho công tác gán nhãn. Các phương pháp gán nhãn đã được sử dụng trong quá trình nghiên cứu:

- Gán nhãn thủ công.
- Gán nhãn đoạn văn bản sử dụng K-Nearest Neighbors (KNN).

## 4.5 So sánh trùng lặp đoạn văn bản

Sau khi thực hiện toàn bộ các bước ở trên, tôi tiến hành công việc kiểm tra đoạn văn bản đã được lưu ở trong cơ sở dữ liệu hay chưa, nhằm mục đích đảm bảo lưu trữ một phiên bản duy nhất cho cùng một nội dung.

Tôi thực hiện công tác so sánh các vector embedding từ các đoạn văn bản có được từ quá trình 4.3 với các đoạn văn bản cùng một nhãn.

Lúc này, độ đo cosine sẽ được sử dụng. Độ tương tự cosine đo lường độ tương tự giữa các tài liệu bằng cách đo cosine của góc giữa hai vector. Kết quả độ tương tự cosine có thể nhận giá trị từ 0 đến 1. Nếu các vector hướng cùng hướng thì độ tương tự là 1, nếu các vector hướng ngược nhau thì độ tương tự là 0. Với giá trị của độ đo cosine cao hơn một ngưỡng nhất định (threshold) thì đoạn văn bản sẽ không được lưu vào cơ sở dữ liệu và ngược lại,

Điểm hay của độ tương tự cosine là nó tính toán hướng giữa các vector chứ không phải độ lớn. Do đó, nó sẽ nắm bắt được sự tương đồng giữa hai tài liệu tương tự nhau mặc dù có kích thước khác nhau.

Tuy nhiên, việc sử dụng độ đo cosine sẽ có nhược điểm là nó chỉ có thể được thực hiện theo cặp, do đó cần phải so sánh nhiều hơn. Điều này có thể làm giảm hiệu suất của hệ thống, nhất là khi số lượng đoạn văn trong cơ sở dữ liệu trở nên quá lớn.

## 4.6 Kiểm thử phương pháp

Sau khi hoàn thành việc triển khai hệ thống, sẽ thực hiện quá trình chạy thử và kiểm thử bằng một loạt các testcases. Đây là các văn bản giáo dục được lấy từ kho dữ liệu cho Chatbot, từ các website chính thống của trường Đại học Bách Khoa Thành phố Hồ Chí Minh. Mục tiêu là đảm bảo rằng giải pháp đề ra sẽ đáp ứng đúng yêu cầu ban đầu.

Việc đánh giá sẽ dựa trên kết quả thu được từ quá trình tách đoạn và kết quả so sánh các đoạn văn trùng lặp trong cơ sở dữ liệu để xác định xem testcases nào làm đúng, testcase nào cần điều chỉnh. Bài báo cáo sẽ tập trung vào việc phát hiện, tìm hiểu nguyên nhân và khắc phục các sai sót xuất hiện trong quá trình thử nghiệm.

Các tham số được sử dụng để kiểm thử trong hệ thống:

**Bảng 4.1:** Tham số sử dụng kiểm thử.

| **Tham số** | **Ý nghĩa** | **Giá trị** |
|---|---|---|
| chunk_slide_window_size | Kích thước cửa sổ trượt qua văn bản trong phân đoạn (đơn vị: Câu) | [2; 5] |
| kl_threshold | Ngưỡng mà tại đó xác định phân phối chủ đề giữa 2 đoạn văn bản là khác nhau | [0,7; 0,9] |
| cosine_threshold | Ngưỡng mà tại đó 2 đoạn văn bản được xem là khác nhau về ý nghĩa | 0,9 |

Đối với đánh giá kết quả cho phần phân tách đoạn văn bản, kết quả sẽ được đánh giá chủ quan dựa trên 4 mức độ tăng dần: Kém, Trung bình, Khá, Tốt.

Thực hiện công tác đánh giá trên 50 văn bản dài, 70% trong số đó cho kết quả ở mức Khá trở lên, với cửa sổ trượt có kích thước 2 câu hoặc 3 câu và ngưỡng so sánh với độ đo Kullback-Leibler dao động từ 0,75 đến 0,8. Đối với những văn bản cho kết quả Trung bình hoặc Kém là những văn bản chứa kiểu dữ liệu dạng bảng, có nhiều header, footnote, trích dẫn,... Cần phải tiền xử lý những dạng dữ liệu này trong văn bản trước khi tiến hành chạy giải thuật.

## 4.7 Định hướng tương lai

### 4.7.1 Sử dụng bộ dữ liệu lớn

Hiện tại, tôi chỉ sử dụng một phần dữ liệu là các văn bản thuộc hệ thống của Trường Đại học Bách khoa Thành phố Hồ Chí Minh. Tuy nhiên, nhận thức về sự quan trọng của việc mở rộng và tận dụng toàn bộ nguồn dữ liệu, tôi dự định tiếp tục xử lý hết tất cả các tệp văn bản khác trong tương lai.

Bằng cách này, tôi có thể theo dõi cũng có nhìn nhận tổng quan, đánh giá chính xác về hệ thống. Việc mở rộng nguồn dữ liệu cũng sẽ hỗ trợ trong việc giải quyết những trường hợp ngoại lệ và mang lại tính linh hoạt cao hơn trong việc xử lý các tình huống đặc biệt và đa dạng.

### 4.7.2 Xây dựng hệ thống đánh giá khách quan

Do công việc phân tách đoạn cũng như đánh giá các đoạn văn có cùng ngữ nghĩa hay không là công việc mang tích chất chủ quan, do đó khi cung cấp một phương pháp để thực hiện các hoạt động trên cho một số lượng lớn văn bản, các kết quả đạt được

có thể tốt hoặc chưa tốt tùy vào sự đánh giá của mỗi cá nhân. Do đó để có thể tạo ra được một tiêu chuẩn đánh giá tốt là một thách thức cần được giải quyết thời gian tới.

### 4.7.3 Kết hợp với mô hình có khả năng học

Việc phải liên tục thay đổi tham số để điều chỉnh ra được các kết quả tốt cho công tác phân đoạn với từng nhóm, từng loại văn bản khác nhau là một mối quan tâm đặt biệt với tôi. Trong thời gian sắp, việc nghiên cứu và đưa ra hướng giải quyết cho vấn đề này là điều hoàn toàn cần thiết và cần phải thực hiện nếu muốn tăng độ linh hoạt trong việc xử lý của hệ thống.

# Chương 5

# Tổng kết

Trong bài báo cáo thực tập 1 này, tôi đã tìm hiểu về các mô hình, các công trình nghiên cứu liên quan để xây dựng một công cụ cho phép phân tách và so sánh các đoạn văn phục vụ công tác tiền xử lý dữ liệu, đồng thời cũng đã chạy thử với những trường hợp cơ bản được đặt ra. Hệ thống hiện tại đã có thể phân tách một văn bản có kích thước lớn thành các đoạn văn ngắn và so sánh với các đoạn văn khác có trong cơ sở dữ liệu để đảm bảo chỉ lưu một phiên bản duy nhất với cùng một nội dung. Tuy nhiêu vẫn còn một vài hạn chế sau đây:

**Thực hiện chưa tốt với xử lý hoàn toàn tự động:** Một trong những vấn đề mà tôi gặp phải là phải thực hiện tinh chỉnh các tham số nhiều lần cho một hoặc một nhóm các văn bản khác nhau. Điều này có thể tạo ra trải nghiệm không tốt cho người dùng và giảm hiệu suất của hệ thống.

**Kết quả phân tách đôi khi không chính xác:** Vì sử dụng mô hình LDA cùng với một số công cụ hỗ trợ như Underthesea, nên đôi khi tôi gặp phải vấn đề về sự không chính xác trong việc cho ra các đoạn văn bản tốt. Điều này có thể làm giảm độ tin cậy của giải pháp.

**Thiếu hệ thống đánh giá khách quan.**

Trong tương lai, tôi sẽ tiếp tục tìm hiểu những mô hình, công nghệ mới để giúp giải quyết những vấn đề trên đồng thời tiến hành chạy trên bộ dữ liệu lớn để phục vụ công tác xử lý dữ liệu cho Hệ thống hỏi đáp và Hệ thống hỏi đáp hội thoại có tương tác tại Trường Đại học Bách khoa Thành phố Hồ Chí Minh.

# Danh sách tham khảo


[1] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *In: Journal of Machine Learning Research 3*, 2003.

[2] J. Han, M. Kamber, and J. Pei, "Data mining: Concepts and techniques," *In: Elsevier Inc*, 2012.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *In: Google AI Language*, p. 1.

[4] N. Q. Dat and N. T. Anh, "Phobert: Pre-trained language models for vietnamese," *In: Findings of the Association for Computational Linguistics: EMNLP 2020*, p. 1037–1042, 2020.