ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

**KHOA CÔNG NGHỆ THÔNG TIN**

**Đào Thị Ngọc Giàu – 21120396**

**Nguyễn Thị Ngọc Châm – 21120417**

**Kiên Đinh Mỹ Hạnh – 21120446**

**Báo cáo cuối kỳ – Ứng dụng mô hình ngôn ngữ lớn trong bài toán xác định tương đồng văn bản xuyên ngữ Anh-Việt**

Môn học: Nhập môn xử lý ngôn ngữ tự nhiên

**GIÁO VIÊN HƯỚNG DẪN**

Thầy Nguyễn Hồng Bửu Long

Thầy Lê Thanh Tùng

Thầy Lương An Vinh

Tp. Hồ Chí Minh, tháng 01/2024

**Lời cảm ơn**

Chúng em xin gửi lời cảm ơn chân thành đến thầy ***Nguyễn Hồng Bửu Long***– giảng viên lý thuyết và thầy ***Lê Thanh Tùng,*** thầy ***Lương An Vinh*** – giảng viên thực hành môn học *Nhập môn xử lý ngôn ngữ tự nhiên* đã trang bị cho nhóm chúng em tài liệu hướng dẫn cùng những kiến thức và kỹ năng cơ bản cần có để hoàn thành bài thực hành này.

Tuy nhiên trong quá trình thực hiện đồ án, vì kiến thức chuyên ngành còn hạn chế nên nhóm chúng em vẫn còn nhiều thiếu sót khi tìm hiểu, đánh giá và trình bày báo cáo. Rất mong nhận được sự quan tâm, góp ý của các thầy để bài báo cáo thực hành của nhóm chúng em được đầy đủ và hoàn chỉnh hơn.

Chúng em xin chân thành cảm ơn!

**Mục lục**

[**I. Đánh giá mức độ hoàn thành** 4](#_Toc155992020)

[**II. Nội dung đồ án** 5](#_Toc155992021)

[**1.** **Giới thiệu đề tài** 5](#_Toc155992022)

[**1.1. Ứng dụng mô hình ngôn ngữ lớn trong bài toán xác định tương đồng văn bản xuyên ngữ Anh-Việt.** 5](#_Toc155992023)

[**1.2. Mô hình mBERT** 6](#_Toc155992024)

[**1.3. Các công trình liên quan** 6](#_Toc155992025)

[**2.** **Phương pháp cài đặt** 7](#_Toc155992026)

[**2.1. Sử mô hình ngôn ngữ lớn cho bài toán xác định tương đồng ngữ nghĩa văn bản xuyên ngữ Anh-Việt.** 7](#_Toc155992027)

[**2.1.1. Các bước tiến hành tạo tập dữ liệu** 8](#_Toc155992028)

[**2.1.2. Fine-tuning mô hình mBERT** 14](#_Toc155992029)

[**2.1.3. Sử dụng mô hình fine-tuned để biểu diễn các đoạn văn bản và đánh giá sự tương đồng bằng độ đo cosine similarity** 15](#_Toc155992030)

[**2.2. Xây dựng ứng dụng cho phép nhập văn bản để so sánh độ tương đồng ngữ nghĩa văn bản xuyên ngữ Anh-Việt** 16](#_Toc155992031)

[**3.** **Mô tả ngữ liệu** 17](#_Toc155992032)

[**4. Thực nghiệm** 19](#_Toc155992033)

[**5.** **Kết luận** 20](#_Toc155992034)

[**6. Tài liệu tham khảo** 20](#_Toc155992035)

# **I. Đánh giá mức độ hoàn thành**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **MSSV** | **Họ và tên** | **Phân công** | **Đóng góp** |
| 1 | 21120396 | Đào Thị Ngọc Giàu | - Xây dựng ứng dụng tích hợp mô hình ngôn ngữ lớn.  - Demo ứng dụng.  - Viết báo cáo | 100% |
| 2 | 21120417 | Nguyễn Thị Ngọc Châm | - Tìm hiểu và sử dụng mô hình ngôn ngữ lớn trong bài toán xác định tương đồng văn bản xuyên ngữ Anh-Việt.  - Thiết kế slide báo cáo.  - Viết báo cáo | 100% |
| 3 | 21120446 | Kiên Đinh Mỹ Hạnh | - Thu thập thập dữ liệu, tạo dataset.  - Báo cáo thuyết trình.  - Viết báo cáo | 100% |

# **II. Nội dung đồ án**

# **Giới thiệu đề tài**

## **1.1. Ứng dụng mô hình ngôn ngữ lớn trong bài toán xác định tương đồng văn bản xuyên ngữ Anh-Việt.**

- Bài toán đánh giá tương đồng văn bản xuyên ngữ Anh-Việt là một thách thức trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP). Mục tiêu chính của bài toán này là đo lường mức độ tương đồng giữa hai đoạn văn bản, một bằng tiếng Anh và một bằng tiếng Việt. Đánh giá tương đồng văn bản có thể áp dụng trong nhiều ứng dụng, bao gồm:

**+ Dịch máy**: Đánh giá mức độ tương đồng giữa văn bản nguồn và văn bản dịch để đảm bảo chất lượng của quá trình dịch.

**+ Tìm kiếm thông tin**: Xác định mức độ tương đồng giữa các đoạn văn bản để cải thiện hiệu suất tìm kiếm và truy xuất thông tin.

**+ Phân loại văn bản**: Đánh giá sự tương đồng giữa các bài báo, bài viết hoặc văn bản khác nhau để hỗ trợ quá trình phân loại.

**+ Kiểm tra đồng nghĩa và chuyển đổi ngôn ngữ**: Đánh giá mức độ tương đồng giữa các biểu diễn ngôn ngữ Anh và Việt để hỗ trợ các ứng dụng như chuyển đổi ngôn ngữ và kiểm tra đồng nghĩa.

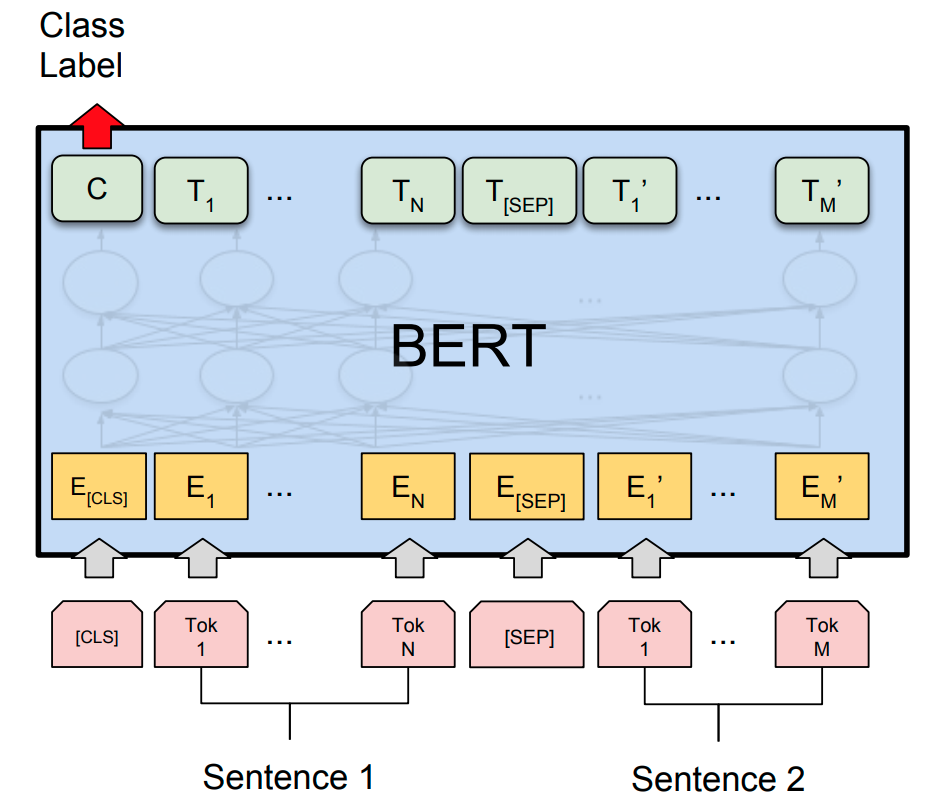
- Cách tiếp cận bài toán này thường sử dụng các kỹ thuật xử lý ngôn ngữ tự nhiên, biểu diễn từ (word embeddings), và mô hình học máy như mô hình BERT, GPT, hoặc các mô hình siamese network. Đối với tiếng Anh và tiếng Việt, có thể sử dụng các bộ dữ liệu song ngữ để huấn luyện mô hình với cặp câu tương đồng và không tương đồng.

- Đánh giá hiệu suất của mô hình thường được thực hiện thông qua các độ đo như precision, recall, F1-score, hoặc cosine similarity giữa các biểu diễn văn bản. Bài toán này đặt ra nhiều thách thức do sự đa dạng của ngôn ngữ và khác biệt về cấu trúc câu trong cả hai ngôn ngữ.

## **1.2. Mô hình mBERT**

- mBERT là một BERT đa ngôn ngữ được đào tạo trước trên 104 ngôn ngữ, được phát hành bởi tác giả của bài báo gốc về kho lưu trữ GitHub chính thức của Google Research: **[google-research / bert](https://github.com/google-research/bert/blob/master/multilingual.md)on** Tháng 11/2018. mBERT tuân theo cấu trúc tương tự của BERT.

- Sự khác biệt duy nhất là rằng mBERT được đào tạo trước về dữ liệu Wikipedia được ghép nối cho 104 ngôn ngữ và nó hoạt động tốt một cách đáng ngạc nhiên so với việc nhúng từ đa ngôn ngữ trên zero-shot chuyển giao đa ngôn ngữ trong tập dữ liệu XNLI, tập dữ liệu **Suy luận ngôn ngữ tự nhiên xuyên ngữ (XNLI)** đã trở thành một tập dữ liệu tiêu chuẩn cho mục đích này. mBERT được sử dụng rộng rãi cho các nhiệm vụ đa ngôn ngữ.

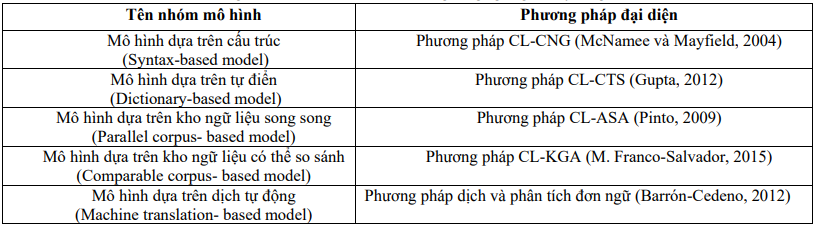


## **1.3. Các công trình liên quan**

- Nghiên cứu của Potthast [1] đã phân loại các phương so sánh độ tương đồng ngữ nghĩa xuyên ngữ theo năm mô hình như trong Bảng 1. Nhóm mô hình dựa trên cấu trúc với phương pháp có phương pháp CL-CNG [2] làm đại diện. Ý tưởng chính của phương pháp này là so sánh các cặp câu sử dụng các n-gram được trích xuất từ các từ liên tiếp nhau trong câu. Phương pháp này không đạt hiệu quả cao trên các cặp ngôn ngữ khác cấu trúc cú pháp hoặc không cùng nhóm ngôn ngữ, nên không áp dụng được hiệu quả cho cặp ngôn ngữ Anh - Việt. Bên cạnh đó, phương pháp CL-CTS [3] đại diện cho nhóm mô hình dựa trên tự điển, có ý tưởng chính là biểu diễn văn bản dưới dạng véctơ khái niệm và tiến hành so sánh độ tương đồng hai văn bản dựa trên hai véctơ của chúng.

- Với nhóm mô hình dựa trên kho ngữ liệu song song, phương pháp CL-ASA [4] được phát triển dựa trên công nghệ dịch máy thống kê. Với hai văn bản d và d’ thuộc hai ngôn ngữ khác nhau L và L’, phương pháp tính toán xác suất mỗi từ ở d là bản dịch của mỗi từ ở d’ dựa trên cặp kho ngữ liệu song song thuộc hai ngôn ngữ L và L’. Từ xác suất các cặp từ là bản dịch của nhau, tính toán xác suất hai văn bản d và d’ là bản dịch của nhau. Mô hình này phụ thuộc nhiều vào chất lượng kho ngữ liệu và mô hình Length và chỉ hiệu quả cao với cặp câu được dịch bởi các chuyên gia hay dịch tự động.

Bảng 1. Các mô hình so sánh độ tương đồng ngữ nghĩa xuyên ngữ



# **Phương pháp cài đặt**

## **2.1. Sử mô hình ngôn ngữ lớn cho bài toán xác định tương đồng ngữ nghĩa văn bản xuyên ngữ Anh-Việt.**

- Ý tưởng thực hiện: Sử dụng mô hình ngôn ngữ lớn huấn luyện trước và thực hiện fine-tuning cho bài toán xác định độ tương đồng giữa 2 văn bản Anh -Việt.

**+ Bước 1:** Chuẩn bị dữ liệu:

* Thu thập và chuẩn bị dữ liệu văn bản gồm những cặp câu tiếng Anh và tiếng Việt, đánh giá độ tương đồng của từng cặp câu.
* Tiền xử lý dữ liệu.
* Phân chia dữ liệu thành các tập huấn luyện (train) và thử nghiệm (test).

**+ Bước 2:** Fine-tuning (Tinh chỉnh): Thực hiện tinh chỉnh trên mô hình ngôn ngữ lớn bằng cách sử dụng dữ liệu huấn luyện đã chuẩn bị.

**+ Bước 3:** Biểu diễn văn bản: Sử dụng mô hình ngôn ngữ lớn để biểu diễn các đoạn văn bản trong tập dữ liệu.

**+ Bước 4:** Đo độ tương đồng: Sử dụng độ đo tương đồng cosine similarity để đo lường sự tương đồng giữa các biểu diễn văn bản.

Ảnh có chứa văn bản, Phông chữ, màu trắng, ảnh chụp màn hình

Mô tả được tạo tự động

**+ Bước 5:** Đánh giá hiệu suất: Sử dụng tập dữ liệu thử nghiệm (test) để đánh giá hiệu suất của mô hình.

### **2.1.1. Các bước tiến hành tạo tập dữ liệu**

- Tập dữ liệu ban đầu được tải về từ hugging face với độ lớn hơn 11.000 cặp câu Anh-Việt tương đồng nhau về mặt ngữ nghĩa. Các cặp câu Anh-Việt được thu thập từ nhiều tập dữ liệu song ngữ như TED2020 v1, wikimedia v20210402, WikiMatrix v1, OpenSubtitiles v2018.

- Cột **“en”** chứa câu tiếng Anh, cột **“vi”**  chứa câu tiếng Việt.

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phông chữ

Mô tả được tạo tự động

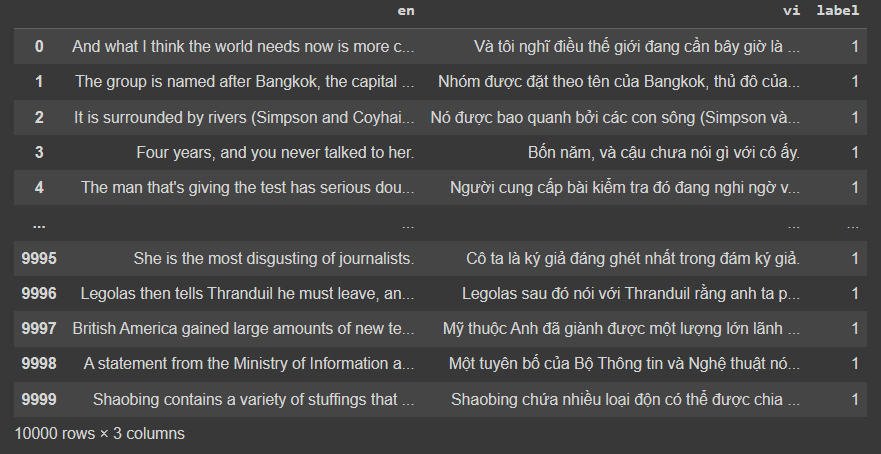
- **Cắt tập dữ liệu còn 10.000 dòng để tiện cho các bước xử lý**

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

- **Thay đổi nội dung tập dữ liệu**

**+ Bước 1:** Thay cột **“source”** bằng cột **“label”**, đánh số 1 (tương đồng) cho toàn bộ cột **“label”**.



- **Bước 2:** Tách tập dữ liệu thành 2 tập dữ liệu con có độ lớn bằng nhau.

**+ Tập dữ liệu 1:**

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phông chữ

Mô tả được tạo tự động

**+ Tập dữ liệu 2:**

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phông chữ

Mô tả được tạo tự động

- **Bước 3:** Xáo trộn các câu ở cột **“vi”** và đánh **“label”** bằng **0** ở tập dữ liệu 1.

**+ Tập dữ liệu 1 trước khi xáo trộn:**

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phông chữ

Mô tả được tạo tự động

**+ Tập dữ liệu 1 sau khi xáo trộn:**

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phông chữ

Mô tả được tạo tự động

* **Bước 4:** Gộp 2 tập dữ liệu đã phân tách trước đó lại thành 1. Ta được tập dữ liệu dùng cho bài toán.

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phông chữ

Mô tả được tạo tự động

* **Bước 5:** Xáo trộn ngẫu nhiên các dòng dữ liệu

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phông chữ

Mô tả được tạo tự động

- **Tiền xử lý dữ liệu văn bản tiếng Anh và tiếng Việt:** Tiền xử lý văn bản làm cho văn bản dễ xử lý hơn và giảm độ phức tạp của dữ liệu. Dưới đây là mô tả chi tiết các bước tiền xử lý:

+ **Bước 1:** Chuyển đổi văn bản thành chữ thường **(`lowercase`**):

🡪 Mục tiêu: Chuyển đổi tất cả các ký tự thành chữ thường để đồng nhất hóa dữ liệu.

+ **Bước 2:** Loại bỏ ký tự không cần thiết (**`[^a-zA-Z0-9\s]`** hoặc **`[^\w\s]`**):

🡪 Mục tiêu: Loại bỏ các ký tự không phải là chữ cái, chữ số hoặc khoảng trắng.

+ **Bước 3:** Tách từ (**`split`** hoặc **`word\_tokenize`**):

🡪 Mục tiêu: Chia văn bản thành các từ riêng lẻ để tiếp tục xử lý từng từ.

+ **Bước** 4: Loại bỏ từ dừng (**`stop words`**):

🡪 Mục tiêu: Loại bỏ các từ không mang ý nghĩa như **"and"**, **"the"**, **"is"**,... để giảm kích thước của dữ liệu và tăng độ chính xác.

+ **Bước** 5: Stemming hoặc Lemmatization (**`PorterStemmer`** hoặc **`WordNetLemmatizer`**):

🡪 Mục tiêu: Chuyển đổi từ về dạng gốc để giảm số lượng biến thể và làm cho từ ngữ đồng nhất hóa hơn.

+ **Stemming (`PorterStemmer`)**: Cắt bớt các hậu tố của từ để đạt được hình thức gốc.

+ **Lemmatization (`WordNetLemmatizer`)**: Chuyển đổi từ về dạng từ điển (định danh) của từ.

+ **Bước 6:** Ghép từ sau khi xử lý lại thành câu **(`join`):**

🡪 Mục tiêu: Tạo lại câu từ các từ đã được xử lý.

- Dưới đây là mô tả chi tiết mỗi bước áp dụng cho tiếng Anh **(`language='en'`)** và tiếng Việt **(`language='vi'`):**

+ Tiếng Anh **(`language='en'`)**:

* Chuyển đổi văn bản thành chữ thường **(`lower()`).**
* Loại bỏ ký tự không cần thiết **(`[^a-zA-Z0-9\s]`).**
* Tách từ **(`split`).**
* Loại bỏ từ dừng **(`stop words`).**
* Stemming **(`PorterStemmer`).**
* Lemmatization **(`WordNetLemmatizer`).**

+ Tiếng Việt **(`language='vi'`):**

* Chuyển đổi văn bản thành chữ thường **(`lower()`).**
* Loại bỏ ký tự không cần thiết **(`[^\w\s]`).**
* Tách từ **(`word\_tokenize`).**
* Loại bỏ từ dừng **(`stop words`).**
* Nhận diện thực thể **(`ner(word)[0][0]`).**

Cuối cùng, áp dụng các bước tiền xử lý này cho cả cột **“en”** (tiếng Anh) và **“vi”** (tiếng Việt) trong DataFrame **“df”**. Kết quả cuối cùng là DataFrame được hiển thị với văn bản đã được tiền xử lý.



### **2.1.2. Fine-tuning mô hình mBERT**

- Quá trình fine-tuning mô hình sử dụng PyTorch và Hugging Face Transformers

**+ Token hóa dữ liệu:**

* Dữ liệu đào tạo **(train\_texts)** và dữ liệu phát triển **(dev\_texts)** được token hóa bằng **tokenizer** được tạo từ mô hình **mBERT** đã chọn.
* Thực hiện việc cắt (**truncation**) và thêm đệm (**padding**) để đảm bảo mọi đầu vào có cùng độ dài.

**+ Tạo Dataset cho PyTorch:**

* Sử dụng các mã hóa từ bước trước để tạo ra các đối tượng **CustomDataset** cho cả tập dữ liệu đào tạo và tập dữ liệu phát triển.
* Mỗi mục trong dataset bao gồm các đặc trưng đầu vào (**encodings**) và nhãn tương ứng (**labels**).

**+ Tạo DataLoader cho PyTorch:**

* Sử dụng **DataLoader** để tạo các **mini**-**batch** từ dataset.
* **Batch size** được chọn là **16**, và dữ liệu được xáo trộn trong quá trình huấn luyện.

**+ Chuẩn bị GPU:**

* Kiểm tra xem GPU có sẵn hay không và đặt mô hình lên GPU.
* Ở đây nhóm sử dụng **T4 GPU 16GB VRAM** của Google Colab.

**+ Tiến hành Fine-tuning:**

* Sử dụng tối ưu hóa **AdamW** với **learning rate** là **2e-5**.
* Sử dụng giảm **learning rate** theo chu kỳ **(OneCycleLR)** để cập nhật **learning rate** trong suốt quá trình huấn luyện.
* Huấn luyện mô hình qua một số **epochs** được chọn (ở đây **epochs = 3**).
* Với mỗi **epoch**:
* Đặt mô hình vào chế độ huấn luyện.
* Với mỗi **batch** trong tập dữ liệu đào tạo, thực hiện các bước sau:
* Chuyển dữ liệu lên GPU nếu cần.
* Đặt **gradients** bằng **0** **(optimizer.zero\_grad()).**
* Chạy dữ liệu qua mô hình (**outputs = model(\*\*inputs)).**
* Tính **loss** và thực hiện **backpropagation (loss.backward()).**
* Cập nhật trọng số mô hình **(optimizer.step()).**
* Cập nhật **learning rate** **(scheduler.step()).**

Bằng cách thực hiện những bước này qua mỗi **epoch**, quá trình **fine-tuning** giúp mô hình tối ưu hóa các trọng số dựa trên dữ liệu đào tạo, với mong muốn là mô hình sẽ học được biểu diễn tốt cho nhiệm vụ đánh giá độ tương đồng văn bản xuyên ngữ Anh-Việt.

### **2.1.3. Sử dụng mô hình fine-tuned để biểu diễn các đoạn văn bản và đánh giá sự tương đồng bằng độ đo cosine similarity**

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

- Sau khi tính độ đo tương đồng cosine similarity, ta gán nhãn độ tương đồng ứng với từng mức độ:

+ Không tương đồng: CosineSimilarity = [0;0.5).

+ Khá tương đồng: CosineSimilarity = [0.5;0.8).

+ Rất tương đồng: CosineSimilarity = [0.8;1.0).

## **2.2. Xây dựng ứng dụng cho phép nhập văn bản để so sánh độ tương đồng ngữ nghĩa văn bản xuyên ngữ Anh-Việt**

- Ý tưởng thực hiện: Tạo web server, dùng HTML/CSS để tạo giao diện người dùng sau đó tích hợp mô hình đánh giá tương đồng văn bản vào ứng dụng web.

- Mô hình được xây dựng bằng framework FastAPI và được lưu trữ trong file model.py

Các endpoint API

/: Trả về trang web chính sử dụng template Jinja2.

/measure\_dis: Nhận 2 câu văn bản, tính toán độ tương đồng theo cosine, manhattan, euclidean, trả về kết quả dưới dạng JSON.

/cluster: Nhận một tập văn bản và số lượng cụm, thực hiện phân cụm KMeans, trả về kết quả HTML hiển thị biểu đồ phân cụm.

- Tương tác với mô hình

+ Hàm get\_model được sử dụng để khởi tạo và cung cấp mô hình cho các endpoint API

+ Các endpoint API gọi các phương thức của mô hình để thực hiện các tác vụ xử lý văn bản và trả về kết quả.

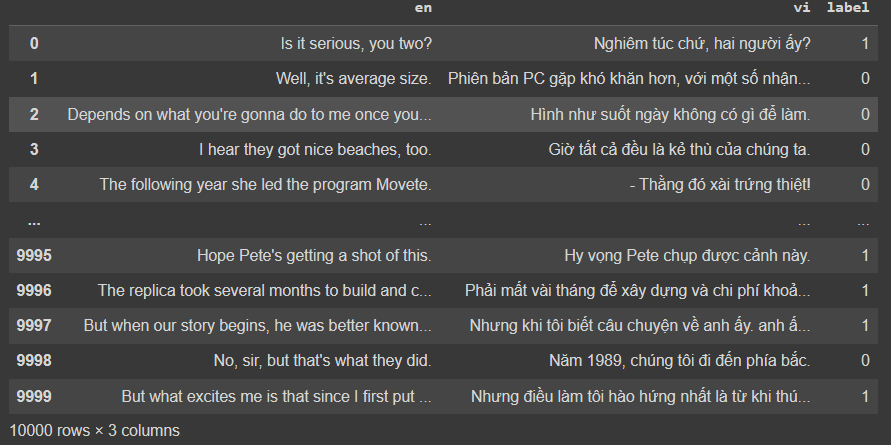
# **Mô tả ngữ liệu**

- Tập ngữ liệu gồm 10.000 cặp câu Anh-Việt được gán nhãn:

+ Cột **“en”** chứa văn bản tiếng Anh.

+ Cột **“vi”** chứa văn bản tiếng Việt.

+ Cột **“label”** đánh giá sự tương đồng của 2 câu Anh-Việt (1: Tương đồng, 0: Không tương đồng).



**- Tập ngữ liệu qua tiền xử lý:**

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm

Mô tả được tạo tự động

**- Chia tập huấn luyện:**

+ Số lượng samples:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

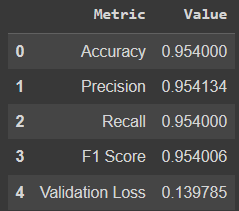
**- Tỷ lệ phân chia nhãn:**

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động

# **4. Thực nghiệm**

**- Kết quả các độ đo đánh giá:**

****

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

- **Kết quả so sánh sự tương đồng:**

**+ Test 1:**

Ảnh có chứa văn bản, Phông chữ, thuật in máy

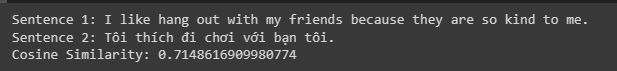
Mô tả được tạo tự động

**+ Test 2:**

**Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động**

**+ Test 3:**

****

# **Kết luận**

Qua đề tài nhóm đã được hiểu thêm về các mô hình ngôn ngữ lớn và ứng dụng chúng trong các nhiệm vụ xử lý ngôn ngữ tự nhiên. Tuy nhiên do tài nguyên còn hạn chế nên nhóm chưa thực sự hoàn thiện được mô hình, cần tinh chỉnh thêm với bộ dữ liệu lớn hơn và điều chỉnh các tham số để đạt hiệu quả cao nhất. Ngoài ra ứng dụng tích hợp mô hình ngôn ngữ lớn còn khá đơn giản và chưa ổn định, chỉ mới được chạy trên máy ảo Google colab, chưa sử dụng được như một ứng dụng độc lập trên nhiều thiết bị khác nhau và có thể sẽ xảy ra lỗi trong quá trình thử nghiệm.

# **6. Tài liệu tham khảo**

Emerging Cross-lingual Structure in Pretrained Language Models: Shijie Wu, Alexis Conneau, Haoran Li, Luke Zettlemoyer, Veselin Stoyanov - Department of Computer Science, Johns Hopkins University (2020).

XÁC ĐỊNH TƯƠNG ĐỒNG XUYÊN NGỮ ANH - VIỆT SỬ DỤNG MÔ HÌNH ĐỒ THỊ: Lê Thành Nguyên, Trần Gia Trọng Nhân, Trần Công Hậu, Đinh Điền - Trường Đại học Khoa học Tự Nhiên, Đại học Quốc gia Thành phố Hồ Chí Minh (2019).

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova (2019)