ĐẠI HỌC BÁCH KHOA HÀ NỘI

PROJECT 3

Ứng dụng mô hình học sâu trong tìm kiếm câu hỏi tương đồng

Nguyễn Văn Thọ 20204694
Tho.NV204694@sis.hust.edu.vn

Giảng viên hướng dẫn: TS. Trần Văn Đặng
Khoa: Khoa học máy tính
Trường: Công nghệ thông tin và Truyền thông

Hà Nội, 1/2024

LỜI CẢM ƠN

Em xin được gửi lời cảm ơn đến TS. Trần Văn Đặng đã hỗ trợ em trong quá trình nghiên cứu và thực hiện **Project 3**, giúp em hiểu thêm một số kiến thức nền tảng để phục vụ cho việc thực hiện Đồ án tốt nghiệp vào kỳ tới. Trong quá trình thực hiện Project 3, vì điều kiện thời gian và kiến thức còn hạn chế, nên chắc chắn không tránh khỏi những sai sót. Em mong nhận được sự góp ý từ thầy để báo cáo này được hoàn thiện hơn.

TÓM TẮT NỘI DUNG BÁO CÁO

Với sự phát triển bùng nổ của công nghệ, chúng ta có thể tìm kiếm các thông tin mình cần trên mạng internet. Một thực tế cho thấy, dù thông tin có được cung cấp nhiều và đầy đủ đến đâu, thì có nhiều vấn đề mà chúng ta vẫn cần những người có hiểu biết, có chuyên môn giải đáp. Với nhu cầu đó, rất nhiều diễn đàn hỏi đáp đã ra đời, phục vụ cho việc giải đáp các câu hỏi về một chủ đề cụ thể do người dùng đặt.

Tuy nhiên, một vấn đề mà các hệ thống gặp phải là thời gian phản hồi. Không phải câu hỏi nào cũng được giải đáp một cách nhanh chóng và kịp thời. Cùng với đó, có rất nhiều thắc mắc về cùng một nội dung nhưng do cách hỏi khác nhau khiến hệ thống rất mất công sức và thời gian để giải đáp cùng một vấn đề.

Trong báo cáo này, em thực hiện xây dựng mô hình học sâu để tìm kiếm các câu hỏi tương đồng với câu hỏi người dùng đặt ra. Bộ dữ liệu được em thu thập bằng công cụ Scrapy trên trang web về hỏi-đáp pháp luật. Sau đó thực hiện tinh chỉnh mô hình huấn luyện trước E5 (dựa trên BERT) sử dụng phương pháp học tương phản SimCSE, và cùng với đó sử dụng mô hình huấn luyện trước Vietnamese Bi-Encoder của Trung tâm Nghiên cứu Quốc tế về Trí tuệ nhân tạo, Đại học Bách khoa Hà Nội (BKAI) để so sánh với mô hình tự tinh chỉnh.

MỤC LỤC

DANH MỤC HÌNH VẼ

DANH MỤC BẢNG BIỂU

DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

Thuật ngữ	**Ý nghĩa**
API	Giao diện lập trình ứng dụng (Application Programming Interface)
CSS	Định dạng kiểu theo tầng dùng để định dạng phần tử trang web (Cascading Style Sheet)
HTTP	Một loại giao thức mạng trên tầng ứng dụng (HyperText Transfer Protocol)
Python Django	Một khung làm việc (framework) phát triển web được viết bằng Python

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

Trong thời đại của công nghệ phát triển vượt bậc, khả năng truy cập thông tin trở nên dễ dàng hơn bao giờ hết. Internet đã mở ra một thế giới vô tận của kiến thức, nơi chứa đựng các thông tin mà chúng ta cần. Tuy nhiên, thực tế cho thấy, sự phát triển nhanh chóng của thông tin cũng tạo ra sự phức tạp không ngừng trong việc tìm kiếm và hiểu biết. Dù cho chúng ta có thể dễ dàng tìm kiếm thông tin trên công cụ tìm kiếm như Google, một thách thức lớn là việc đảm bảo câu trả lời nhận được là chính xác và có ích. Chúng ta luôn có những câu hỏi mà dù có đầy đủ thông tin vẫn không thể hiểu được, rất cần người có hiểu biết chuyên sâu giải đáp. Để đáp ứng nhu cầu này, nhiều diễn đàn hỏi đáp đã xuất hiện. Tuy nhiên, một vấn đề đặt ra cho các hệ thống này thời gian phản hồi. Không phải lúc nào câu hỏi cũng nhận được sự chú ý và giải đáp một cách nhanh chóng. Ngoài ra, có rất nhiều câu hỏi liên quan đến cùng một chủ đề, nhưng do thiết kế hệ thống mà người dùng không biết được điều này và vẫn đặt câu hỏi gây tốn kém thời gian không cần thiết.

Ngoài ra, sự phát triển nhanh chóng của công nghệ thông tin cùng với mong muốn nâng cao chất lượng cuộc sống của con người đã mở ra nhiều cơ hội mới cho việc áp dụng Trí tuệ nhân tạo (tiếng Anh: Artificial Intelligence, viết tắt: AI) vào nhiều lĩnh vực khác nhau. Một trong những lĩnh vực nghiên cứu của Trí tuệ nhân tạo đang được ứng dụng trong thực tế nhiều nhất hiện nay là Xử lý ngôn ngữ tự nhiên - tập trung vào nghiên cứu sự tương tác giữa máy tính và ngôn ngữ tự nhiên của con người, dưới dạng tiếng nói hoặc văn bản. Trong những ứng dụng của Xử lý ngôn ngữ tự nhiên, phải kể đến như nhận diện tiếng nói (chuyển giọng nói thành văn bản), dịch máy (sử dụng trong các hệ thống dịch thuật như Google Dịch), tóm tắt văn bản, . . . Các kiến trúc mạng như Transformer, BERT, GPT-4,... được giới thiệu trong thời gian gần đây giúp cho các tác vụ về Xử lý ngôn ngữ tự nhiên ngày càng được phát triển và thu hút rất nhiều nhóm nghiên cứu tham gia vào lĩnh vực này.

Từ thực tiễn nêu trên, em đã quyết định lựa chọn đề tài "Ứng dụng học sâu trong tìm kiếm câu hỏi tương đồng" để đáp ứng nhu cầu người dùng mong muốn cải thiện khả năng giải đáp và đề xuất câu hỏi tương tự của các hệ thống hỏi đáp, giúp người dùng tìm ra câu trả lời nhanh hơn và giảm tải hệ thống khi nhiều người cùng gặp một vấn đề.

1.2 Mục tiêu và định hướng giải pháp

Trên cơ sở các phân tích và đánh giá ở phần 1.1, chúng em hướng đến xây dựng mô hình tìm kiếm câu hỏi tương đồng với các mục tiêu như sau:

1. Mô hình có thể đề xuất được các câu hỏi tương đồng (có sẵn trên hệ thống) với câu hỏi người dùng một cách chính xác nhất.
2. Tốc độ xử lý của mô hình phải đủ nhanh.
3. Mô hình được huấn luyện bởi một tập dữ liệu đủ lớn.

Để đáp ứng được mục tiêu như trên, chúng em đề xuất giải pháp về tập dữ liệu và mô hình, cụ thể: i) thu thập dữ liệu từ trang web hỏi-đáp pháp luật để huấn luyện và đánh giá hiệu năng mô hình, ii) tinh chỉnh mô hình E5 huấn luyện trước bằng kỹ thuật học tương phản SimCSE-không giám sát và song song với đó sử dụng mô hình Vietnamese Bi-Encoder của BKAI đã được huấn luyện để so sánh hiệu năng dự đoán của cả hai. Cụ thể mô hình sẽ được mô tả chi tiết ở các chương sau.

1.3 Bố cục báo cáo

Phần còn lại của báo cáo này được tổ chức như sau:

Chương 2 trình bày về cơ sở lý thuyết được áp dụng để thực hiện xây dựng mô hình.

Chương 3 trình bày về phương pháp đề xuất được sử dụng. Trong chương này, chúng em đi sâu vào phân tích, xử lý tập dữ liệu huấn luyện và mô hình đề xuất.

Từ phương pháp đã đề xuất ở chương 3, chúng em tiến hành huấn luyện và đánh giá hiệu năng mô hình. Các kết quả trong quá trình thực nghiệm sẽ được trình bày cụ thể ở chương 4.

Chương 5 là chương cuối cùng. Ở chương này, chúng em tổng kết lại kết quả đã đạt được, những khó khăn gặp phải trong quá trình thực hiện bài tập lớn và phân tích các hướng đi mới cho phép cải thiện và nâng cấp mô hình.

CHƯƠNG 2. NỀN TẢNG LÝ THUYẾT

2.1 Phát biểu bài toán

Bài toán Tìm kiếm câu hỏi tương đồng mức ngữ nghĩa được phát biểu như sau:

- **Đầu vào:** Một chuỗi ký tự là câu hỏi truy vấn.
- **Xử lý:** Đầu vào được xử lý và đưa vào một mô hình học sâu có chức năng tìm kiếm các câu hỏi tương đồng trên bộ dữ liệu có sẵn.
- **Đầu ra:** Một danh sách các câu hỏi tương tự và câu trả lời tương ứng, xếp theo thứ tự giảm dần về độ tương đồng.

2.2 Kiến thức nền tảng

2.2.1 Công cụ Scrapy cho thu thập dữ liệu web

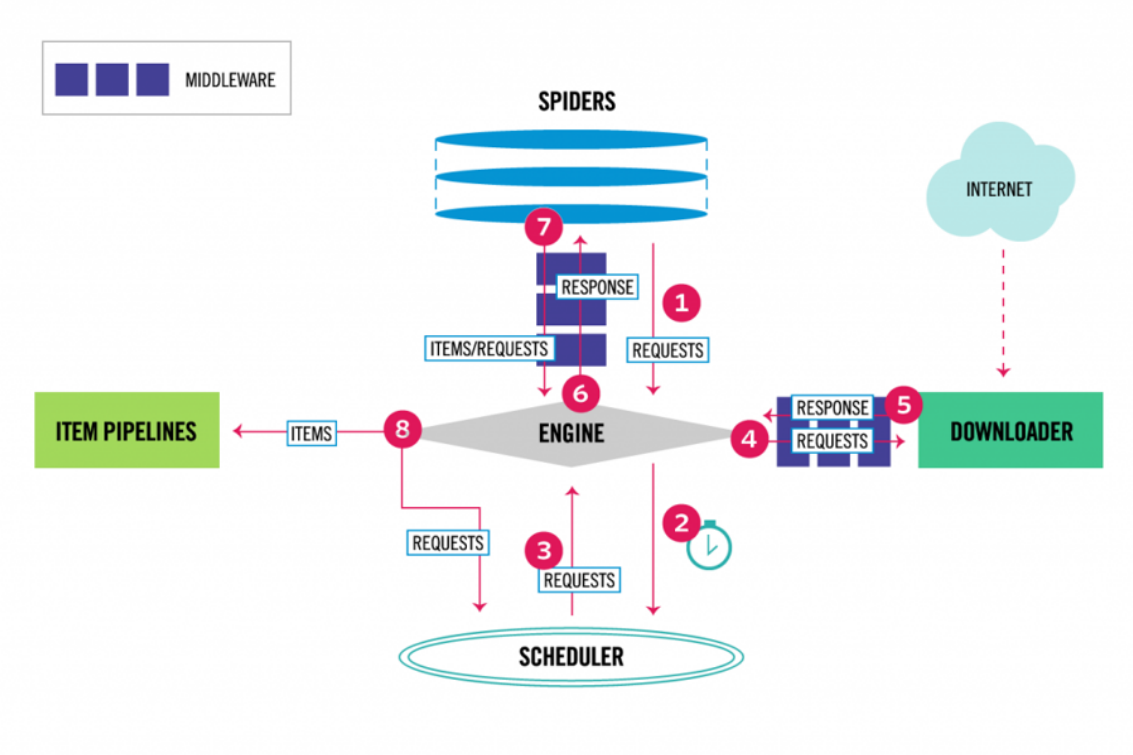
Scrapy là một khung ứng dụng (framework) dùng để thu thập dữ liệu các trang web và trích xuất dữ liệu có cấu trúc, những dữ liệu này có thể được sử dụng cho việc khai thác dữ liệu, xử lý thông tin hoặc lưu trữ lịch sử thông tin.

Mặc dù Scrapy ban đầu được thiết kế để quét web (web scraping), nhưng nó cũng có thể được sử dụng để trích xuất dữ liệu bằng API (chẳng hạn như nền tảng điện toán đám mây liên kết Amazon Associates Web Services) hoặc như một trình thu thập dữ liệu web có mục đích chung. Lấy cảm hứng từ Python Django, Scrapy không chỉ là một thư viện yêu cầu HTTP như Python Requests hoặc một thư viện phân tích cú pháp như BeautifulSoup hoặc lxml, nó là một khung quét web được xây dựng có mục đích bao gồm:

- Thực hiện gửi các yêu cầu HTML Request như GET, POST, v.v.
- Trích xuất dữ liệu từ trang web bằng bộ chọn đường dẫn CSS và XPath.
- Phát hiện các HTML Request gửi đi không thành công và tự động thử lại.
- Song song hóa các yêu cầu với chức năng đồng thời có sẵn.
- Thu thập dữ liệu toàn bộ trang web với phân trang, sơ đồ trang web và liên kết theo dõi.
- Làm sạch, xác thực và xử lý hậu kỳ dữ liệu đã thu thập bằng đường ống (pipeline).
- Lưu dữ liệu vào tệp CSV / JSON, cơ sở dữ liệu và lưu trữ đối tượng.

Kiến trúc Scrapy, như minh họa ở Hình 2.1, gồm các thành phần chính như sau:

- Scrapy Engine: Chịu trách nhiệm điều khiển luồng giữa các thành phần trong



Hình 2.1: Kiến trúc Scrapy cho thu thập dữ liệu web. Số thứ tự trên hình chỉ trình tự thu thập dữ liệu web.

hệ thống và kích hoạt sự kiện khi một số hành động nhất định xảy ra..

- Scheduler (Bộ lập lịch): Có nhiệm vụ nhận các yêu cầu (request) từ Engine và đưa nó vào một hàng đợi (queue) để sắp xếp các URL theo thứ tự tải (download).
- Downloader (Bộ tải trang): Có nhiệm vụ tải mã nguồn HTML của trang web và gửi nó về cho Engine.
- Spider: Là một lớp được viết bởi lập trình viên ,có nhiệm vụ phân tích HTML Response và truy xuất dữ liệu, lưu thành các đối tượng (item), khởi tạo URL mới và nạp lại cho Scheduler qua Engine.
- Item Pipeline: Có nhiệm vụ xử lý các đối tượng sau khi được truy xuất bằng Spider, sau đó lưu vào cơ sở dũ liệu.
- Downloader Middlewares (Bộ tải trang trung gian): Là móc nối giữa Engine và Downloader, chúng xử lý các yêu cầu được đẩy từ engine và các HTML Response tạo ra từ Downloader.
- Spider Middlewares: Là móc nối giữa Engine và Spider, chúng có nhiệm vụ xử lý đầu vào (HTML Response) của Spider và đầu ra (đối tượng (item) và Request).

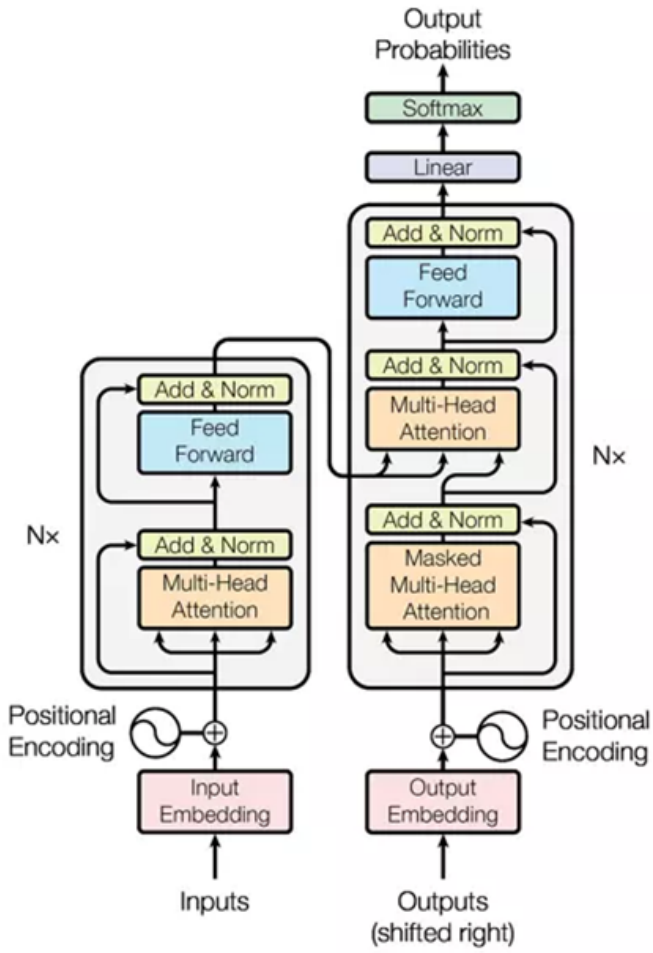
Quy trình thực thi hệ thống cũng được mô tả như trên Hình 2.1. Dưới đây cụ thể hóa các bước thực hiện thu thập dữ liệu bằng Scrapy.

1. Engine khởi tạo yêu cầu (Request) để bắt đầu thu thập/cào dữ liệu (crawl) từ Spider.
2. Engine lên lịch trình cho các Request nhận được từ Spider.
3. Bộ lập lịch (Scheduler) gửi Request tiếp theo đến Engine và yêu cầu Request tiếp theo cần thu thập dữ liệu.
4. Engine gửi Request đến Downloader, đi qua Bộ tải trang trung gian (Downloader Middleware).
5. Sau khi tải mã nguồn HTML hoàn tất, Bộ tải trang (Downloader) khởi tạo một đối tượng phản hồi (Response) trả về qua Engine, quá trình này đi qua Bộ tải trang trung gian.
6. Engine nhận phản hồi từ Bộ tải trang và gửi về Spider để xử lý, quá trình này đi qua Spider Middleware.
7. Spider xử lý phản hồi và trả về các đối tượng (item) đã được truy xuất ,sau đó khởi tạo Request đến Engine, thông qua Spider Middleware.
8. Engine gửi các đối tượng đã được xử lý đến Item Pipelines, sau đó gửi các Request đã được xử lý đến Bộ lập lịch và yêu cầu (nếu còn trong hàng đợi) Request tiếp theo để cào dữ liệu.
9. Tiến trình lặp lại như bước 1, cho đến khi không còn Request nào trong hàng đợi từ Bộ lập lịch.

Khi sử dụng Scrapy cần chú ý đến vấn đề pháp lý và đạo đức. Ta cần tôn trọng quyền của chủ sở hữu trang web đối với dữ liệu của họ. Nếu họ có Tiêu chuẩn loại trừ rô-bốt (thường được các trang cung cấp một tệp robots.txt) trong một số hoặc tất cả các phần của trang web của họ, điều đó có nghĩa là họ không muốn bất kỳ ai thu thập dữ liệu của họ mà không có sự cho phép rõ ràng, ngay cả khi nó có sẵn công khai, thì ta không nên cố gắng truy cập vào nó, dù Scrapy cung cấp một số công cụ có thể "qua mặt" được các trang web này. Ngoài ra cần tránh tải xuống quá nhiều dữ liệu cùng một lúc, vì điều đó có thể làm hỏng máy chủ của trang web và ta có thể bị gắn cờ là một cuộc tấn công từ chối dịch vụ DDoS.

2.2.2 Kiến trúc Transformer và mô hình BERT

Mô hình Transformer [1] là một kiến trúc mạng nơ-ron được giới thiệu bởi Ashish Vaswani và đồng nghiệp trong bài báo “Attention is All You Need” vào năm 2017. Kể từ khi ra đời, mô hình Transformer đã nhanh chóng cho thấy khả năng

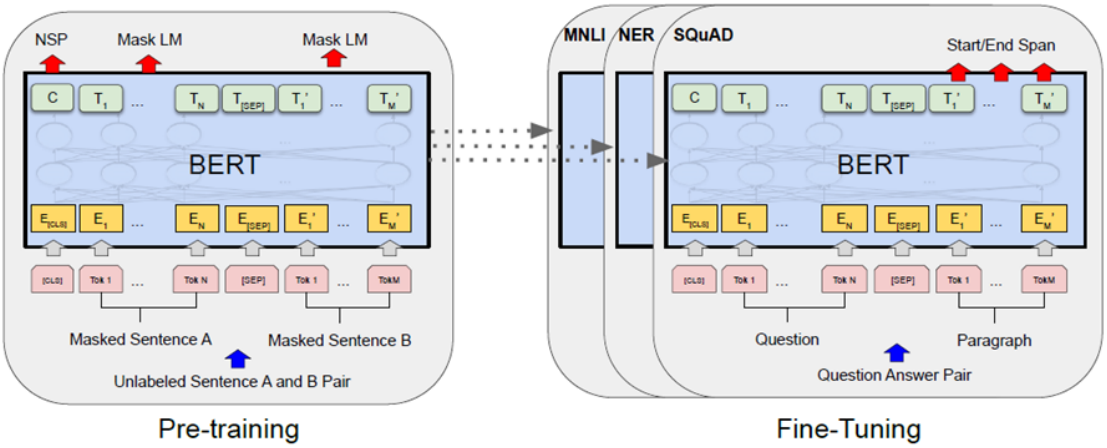


Hình 2.2: Tổng quan về kiến trúc Transformer với hai thành phần: Mã hóa-encoder (trái) và Giải mã-decoder (phải).

tuyệt vời trong việc giải quyết các bài toán trong nhiều lĩnh vực đặc biệt là các bài toán làm việc với dữ liệu dạng chuỗi. Mô hình này ban đầu được xây dựng theo kiến trúc mã hóa-giải mã (encoder-decoder) và dựa trên cơ chế chú ý (attention) để giải quyết bài toán dịch máy, cả phần mã hóa và giải mã của Transformer đều trở thành các thành phần không thể thiếu trong các mô hình ngôn ngữ lớn về sau.

BERT (Bidirectional Encoder Representations from Transformers) [2] là một trong những cải tiến đột phá trong lĩnh vực xử lý ngôn ngữ tự nhiên (NLP) và hiểu ngôn ngữ tự nhiên dựa trên kiến trúc Transformer, được giới thiệu bởi nhóm nghiên cứu tại Google AI Language vào năm 2018. Khi mới công bố, BERT đạt kết quả tốt nhất trên nhiều bài toán của lĩnh vực Xử lý ngôn ngữ tự nhiên như phân loại cảm xúc văn bản, trả lời câu hỏi, nhận dạng thực thể. BERT có khả năng mô hình ngữ cảnh của câu văn theo hai chiều, tức là cho một đoạn văn gồm các từ $x_1, x_2, \ldots, x_n$:

- Mô hình xuôi (Forward Autoregressive): dự đoán từ x_i dựa trên các từ trước đó $x_1, \ldots x_{i-1}$



Hình 2.3: Tổng quan mô hình BERT với hai giai đoạn: huấn luyện trước (pre-training) và tinh chỉnh (fine-tuning).

- Mô hình ngược (Backward Autoregressive): dự đoán từ x_i dựa trên các từ $x_{i+1}, \ldots, x_n$

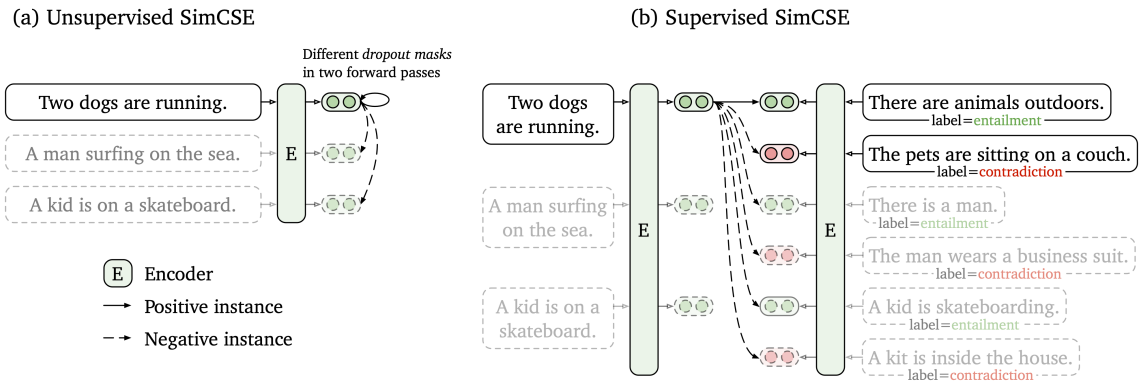
Quá trình huấn luyện BERT gồm hai giai đoạn là huấn luyện trước (pre-training) và tinh chỉnh (fine-tuning).

1. Trong giai đoạn huấn luyện trước, BERT được huấn luyện trên một lượng lớn văn bản không có nhãn, sử dụng phương pháp học tự giám sát (self-supervised) để giải quyết cho tác vụ dự đoán từ bị che giấu (MLM, Masked Language Model) và tác vụ dự đoán mối quan hệ giữa hai câu (câu thứ hai có phải là câu tiếp theo của câu thứ nhất hay không) (NSP, Next Sentence Prediction).
2. Trong giai đoạn tinh chỉnh, tiếp tục huấn luyện mô hình huấn luyện trước với một số tinh chỉnh để phù hợp với các tác vụ cụ thể hơn. Hầu hết các siêu tham số giữ nguyên như trong pre-training BERT, chỉ có một số tham số là thay đổi để phù hợp với tác vụ cần giải quyết.

2.2.3 Học tương phản (Contrastive Learning) với SimCSE

SimCSE (**Sim**ple **C**ontrastive Learning of **S**entence **E**mbeddings) [3] là một phương pháp học tương phản đơn giản cho các nhúng câu, được giới thiệu bởi Tianyu Gao và các cộng sự vào năm 2021.

Ý tưởng chính của học tương phản là tìm ra các cặp đặc trưng của dữ liệu có tính tương đồng - tương phản nhau trong bộ dữ liệu. Từ đó, với những cặp dữ liệu mang tính tương đồng, ta có thể "kéo" chúng lại gần để học được những đặc trưng bậc cao của nhau, và ngược lại với những cặp dữ liệu tương phản sẽ bị "đẩy" ra xa. Để làm được điều này, cần sử dụng các độ đo tương đồng (similarity metric) để tính



Hình 2.4: Ý tưởng của SimCSE. a) SimCSE dạng học không giám sát, sử dụng một câu để dự đoán chính nó chỉ bằng dropout làm nhiễu. b) SimCSE dạng học có giám sát, tích hợp các cặp được chú thích từ các bộ dữ liệu suy luận ngôn ngữ tự nhiên vào học tương phản bằng cách sử dụng cặp "entailment" làm mẫu dương và cặp "contradiction" làm mẫu âm.

toán khoảng cách giữa các vec-tơ nhúng (embedding vector) biểu diễn các điểm dữ liệu với nhau. Ta gọi một tập các mẫu được ghép cặp $\mathcal{D} = \left\{\left(x_i, x_i^+\right)\right\}_{i=1}^m$, trong đó x_i và x_i^+ có mối quan hệ ngữ nghĩa. Tiếp theo, gọi h_i là biểu diễn của x_i và h_i^+ là biểu diễn của x_i^+. Mục tiêu huấn luyện cho cặp $\left(x_i, x_i^+\right)$ với một mini-batch gồm N cặp mẫu là:

$$\ell_i = -\log \frac{e^{\text{sim}\left(\mathbf{h}_i, \mathbf{h}_i^+\right)/\tau}}{\sum_{j=1}^N e^{\text{sim}\left(\mathbf{h}_i, \mathbf{h}_j^+\right)/\tau}} \tag{2.1}$$

trong đó, τ là siêu tham số nhiệt (temperature hyperparameter), $\text{sim}\left(\mathbf{h}_1, \mathbf{h}_2\right)$ là độ đo tương đồng cô-sin $\frac{\mathbf{h}_1^\top \mathbf{h}_2}{\|\mathbf{h}_1\| \cdot \|\mathbf{h}_2\|}$. Trong mô hình này, nhóm tác giả mã hóa các câu đầu vào bằng cách sử dụng một mô hình ngôn ngữ đã được huấn luyện trước như BERT hoặc RoBERTa: $\mathbf{h} = f_\theta(x)$, và sau đó điều chỉnh tất cả các tham số bằng mục tiêu học tương phản như Phương trình (2.1).

Đối với học không giám sát cho SimCSE, thực hiện lấy một tập câu $\{x_i\}_{i=1}^m$ và cho $x_i^+ = x_i$. Thành phần quan trọng để mô hình hoạt động với với các cặp mẫu dương (positive pairs) giống nhau là thông qua việc sử dụng các mặt nạ dropout (dropour masks) được lấy mẫu độc lập cho x_i và x_i^+.

Trong quá trình huấn luyện chuẩn của kiến trúc Transformers, có các mặt nạ dropout được đặt trên các lớp kết nối đầy đủ cũng như xác suất chú ý (attention probability, mặc định $p = 0.1$). Ta gọi $\mathbf{h}_i^z = f_\theta\left(x_i, z\right)$ với z là một mặt nạ ngẫu nhiên cho dropout. Nhóm tác giả đề xuất đưa cùng một đầu vào vào bộ mã hóa (encoder) hai lần và được hai vec-tơ nhúng với các mặt nạ dropout khác nhau z, z', và mục tiêu huấn luyện của SimCSE trở thành:

$$\ell_i = -\log \frac{e^{\text{sim}\left(\mathbf{h}_i^{z_i}, \mathbf{h}_i^{z_i'}\right)/\tau}}{\sum_{j=1}^{N} e^{\text{sim}^{z_i}\left(\mathbf{h}_i^{z_i}, \mathbf{h}_j^{z_j'}\right)/\tau}} \tag{2.2}$$

cho một mini-batch với N câu. Lưu ý rằng, z chỉ là mặt nạ dropout chuẩn trong Transformers và mô hình không thêm bất kỳ dropout bổ sung nào.

CHƯƠNG 3. PHƯƠNG PHÁP ĐỀ XUẤT

3.1 Tổng quan giải pháp

Giải pháp cho tác vụ Tìm kiếm câu hỏi tương đồng mức ngữ nghĩa được tóm tắt như sau:

- Sử dụng bộ dữ liệu được thu thập từ trang web https://thuvienphapluat.vn, thực hiện các thao tác trên dữ liệu: i) phân chia tập dữ liệu thành tập huấn luyện và tập kiểm thử, ii) tiền xử lý dữ liệu bằng cách loại bỏ những mẫu không có câu trả lời, thiếu thông tin câu hỏi,...
- Xây dựng và tinh chỉnh mô hình E5 (Multilingual-E5-base, **E**mb**E**ddings from bidir**E**ctional **E**ncoder r**E**presentations) [4] với SimCSE, bằng việc chỉ đưa trường câu hỏi trong bộ dữ liệu thu thập vào bước tinh chỉnh. Nói về E5, đây là một mô hình dựa trên BERT, được huấn luyện theo phương pháp học tương phản với tín hiệu giám sát yếu (weak supervision signals) từ bộ dữ liệu cặp văn bản CCPairs.
- Xây dựng một giao diện người dùng đơn giản dựa trên thư viện Gradio cho Python.

3.2 Dữ liệu

3.2.1 Thông tin bộ dữ liệu

Bộ dữ liệu được thu thập từ trang web https://thuvienphapluat.vn/ bằng công cụ Scrapy, gồm 130972 dòng, mỗi dòng tương ứng là câu hỏi từ người dùng và câu trả lời có liên quan. Cấu trúc của bộ dữ liệu gồm một tệp tin **pair-question.jsonl** chứa các đối tượng (item) có các trường thông tin sau:

- 'url': Cho biết đường dẫn đến trang web chứa câu hỏi.
- 'title': Cho biết tiêu đề của câu hỏi.
- 'user_question': Cho biết nội dung câu hỏi mà người dùng nhập vào.
- 'question': Cho biết nội dung câu hỏi chính liên quan đến truy vấn người dùng.
- 'answer': Cho biết câu trả lời chi tiết cho câu hỏi.
- 'citation': Chú thích thêm, chỉ số ít câu hỏi có trường này, thông tin này không quan trọng trong dự án này.

3.2.2 Xử lý dữ liệu

Với bất kỳ mô hình học máy, học sâu, phần xử lý dữ liệu thực hiện các thao tác trên dữ liệu đầu vào sao cho phù hợp với yêu cầu của mô hình được sử dụng để

```
"url" :
string "https://thuvienphapluat.vn/hoi-dap-phap-luat/51A2C-hd-thay-doi-tru-so-kinh-doanh-trong-cung-quan-huyen-co-thay-doi-co-quan-quan-ly-thue.html"
"title" : string "Có thay đổi cơ quan quản lý thuế khi thay đổi trụ sở kinh doanh trong cùng quận, huyện?"
"user_question" :
string "Mình muốn hỏi doanh nghiệp mình muốn chuyển địa chỉ kinh doanh khác quận huyện có thay đổi cơ quan quản lý thuế không?"
"question" : string "Có thay đổi cơ quan quản lý thuế khi thay đổi trụ sở kinh doanh trong cùng quận, huyện?"
"answer" :
string "Tại Điểm a Khoản 1 Điều 13 Thông tư 95/2016/TT-BTC hướng dẫn về đăng ký thuế do Bộ trưởng Bộ Tài chính ban hành, có quy định. Các trường hợp thay đổi thông tin đăng ký thuế không làm thay đổi cơ quan thuế quản lý: - Tổ chức kinh tế, tổ chức khác, hộ gia đình, nhóm cá nhân, cá nhân kinh doanh và cá nhân khác thay đổi thông tin đăng ký thuế, trừ thông tin địa chỉ trụ sở. - Tổ chức kinh tế, tổ chức khác do Cục Thuế quản lý thay đổi địa chỉ trụ sở trong phạm vi cùng tỉnh, thành phố trực thuộc Trung ương. - Tổ chức kinh tế, tổ chức khác, hộ gia đình, nhóm cá nhân, cá nhân kinh doanh do Chi cục Thuế quản lý thay đổi địa chỉ trụ sở trong phạm vi cùng quận, huyện, thành phố trực thuộc tỉnh=> Như vậy, theo quy định nêu trên thì việc thay đổi địa chỉ kinh doanh trong cùng quận, huyện không làm thay đổi cơ quan quản lý thuế.Trân trọng."
"citation" : string ""
```


Hình 3.1: Ví dụ cho một đối tượng câu hỏi trong bộ dữ liệu.

huấn luyện mô hình một cách tốt nhất.

Phân chia tập dữ liệu

Tập dữ liệu được chia thành hai tập: tập huấn luyện (training set) và tập kiểm thử (test set), theo tỉ lệ 7: 3. Do đó:

- Số lượng mẫu trong tập huấn luyện là 91680.
- Số lượng mẫu trong tập kiểm thử là 39292.

Tiền xử lý và chuẩn bị dữ liệu cho việc huấn luyện

Để đảm bảo không có dữ liệu nào bị thiếu trường câu trả lời và câu hỏi, tiến hành xóa các dòng không có hai trường thông tin này.

3.3 Tìm kiếm câu hỏi

Sau khi tiền xử lý và chuẩn bị dữ liệu cho việc huấn luyện, tiến hành xây dựng mô hình và huấn luyện.

1. Sử dụng mô hình E5 đã được huấn luyện trước (intfloat/multilingual-e5-base) được cung cấp trên HuggingFace để tinh chỉnh.
2. Với SimCSE, ta chỉ lấy trường 'user_question' làm dữ liệu huấn luyện.
3. Sử dụng hàm mất mát Multiple Negatives Ranking Loss trong quá trình huấn luyện.

Sau khi huấn luyện xong mô hình, tiến hành bước đánh giá hiệu năng mô hình. Dưới đây là các bước thực hiện:

1. Chuyển tập câu hỏi có sẵn trong hệ thống (trường 'question') thành các vec-tơ nhúng câu.

2. Đối với một truy vấn từ người dùng, đo độ tương đồng cô-sin (bằng hàm semantic_search())
3. Đưa ra danh sách 5 câu hỏi có độ tương đồng cao nhất cùng câu trả lời tương ứng.

Kết quả đánh giá hiệu năng mô hình sẽ được trình bày ở chương 4.

3.4 Nền tảng triển khai và công cụ sử dụng

3.4.1 Kaggle

Kaggle, thuộc Google, là một nền tảng trực tuyến cho phép người dùng xây dựng các mô hình học máy và phân tích dữ liệu. Nó cung cấp các cuộc thi về khoa học dữ liệu, nơi những người tham gia sẽ cạnh tranh với nhau để tạo ra các mô hình tốt nhất nhằm giải quyết một số vấn đề cụ thể. Ngoài các cuộc thi, Kaggle cũng cung cấp cho người dùng các bộ dữ liệu công khai, các tài nguyên giáo dục, hướng dẫn và khóa học phong phú cho cả người mới bắt đầu và cả những chuyên gia. Kaggle cũng cung cấp nền tảng để các nhà nghiên cứu về khoa học dữ liệu, học máy và những người có đam mê kết nối, chia sẻ ý tưởng và cộng tác với nhau.

Điểm quan trọng của Kaggle là cho phép người dùng có thể thực thi tệp Jupyter Notebook trên đám mây thay vì thực hiện trên máy tính vật lý. Người dùng có thể tạo, chỉnh sửa, chạy và chia sẻ các tệp ngay trên trình duyệt web mà không cần thiết lập môi trường. Ngoài ra, Kaggle cung cấp quyền truy cập miễn phí vào GPU để tăng tốc độ tính toán và xử lý trong quá trình huấn luyện mô hình học sâu. Hiện tại, Kaggle đang hỗ trợ hai loại GPU là GPU T4 và P100, thời gian sử dụng miễn phí tối đa là 30 tiếng trong mỗi chu kỳ 7 ngày và 12 tiếng cho một phiên chạy.

3.4.2 Colab

Colab (hay Google Colaboratory) là một dịch vụ cung cấp môi trường làm việc trực tuyến để phát triển và chia sẻ các dự án Học máy và Khoa học dữ liệu. Được cung cấp hoàn toàn miễn phí bởi Google, Colab cung cấp một môi trường sử dụng Jupyter Notebook trên đám mây, cho phép người dùng tạo, chia sẻ và sử dụng các tệp notebook một cách dễ dàng mà không cần cài đặt phần mềm hay máy chủ riêng. Các ưu điểm của môi trường Colab bao gồm:

- **Chạy trên đám mây:** Colab chạy hoàn toàn trên môi trường đám mây, cho phép người dùng truy cập và làm việc từ bất kỳ thiết bị nào chỉ cần có kết nối internet. Không cần cài đặt phần mềm hay môi trường làm việc riêng.
- **Hỗ trợ GPU:** Colab cung cấp môi trường sử dụng GPU miễn phí, giúp người dùng có thể thực hiện các tác vụ nặng về tính toán một cách nhanh chóng và hiệu quả. Các loại GPU mà Colab hỗ trợ bao gồm NVIDIA T4 (không yêu

cầu trả phí), V100 và A100.

- **Hỗ trợ các thư viện phổ biến:** Colab hỗ trợ hầu hết các thư viện phổ biến nhất trong lĩnh vực Học máy và Khoa học dữ liệu như TensorFlow, PyTorch, Keras,...

3.4.3 Thư viện PyTorch

PyTorch là một thư viện mã nguồn mở được phát triển bởi Facebook. Nó là một khung làm việc (framework) dành cho học sâu với ngôn ngữ Python, cung cấp một loạt các công cụ và API để xây dựng và đào tạo các mô hình học sâu. PyTorch được sử dụng rộng rãi trong các lĩnh vực như thị giác máy tính, xử lý ngôn ngữ tự nhiên, và học máy. Nó là lựa chọn phổ biến cho các nhà nghiên cứu và kỹ sư học máy do tính linh hoạt và hiệu suất của mình. Các ưu điểm của Pytorch gồm:

- **Linh hoạt:** PyTorch cho phép người dùng có nhiều quyền kiểm soát đối với cấu trúc và hoạt động của mạng nơ-ron. Điều này giúp cho việc phát triển các và tùy chỉnh các mô hình học sâu trở nên dễ dàng hơn.
- **Hiệu suất cao:** PyTorch có thể chạy trên GPU / TPU, giúp tăng tốc độ huấn luyện và dự đoán của các mô hình học sâu.
- **Khả năng mở rộng mạnh mẽ:** PyTorch có một cộng đồng người dùng và nhà phát triển lớn, cung cấp nhiều tài nguyên và hỗ trợ cho người dùng

CHƯƠNG 4. ĐÁNH GIÁ THỰC NGHIỆM

4.1 Các tham số đánh giá

Để đánh giá hiệu năng mô hình, em sử dụng chỉ số Precision@5, Recall@5. Hai chỉ số này thường được dùng trong Hệ gợi ý để đánh giá mức độ liên quan của kết quả đề xuất. Nguyên nhân em sử dụng hai độ đo này vì với mỗi câu hỏi từ người dùng có thể có từ 1 đến 4 câu hỏi chính, trong xếp hạng 5 câu hỏi có độ tương đồng cao nhất có thể có một hoặc nhiều câu hỏi chính đó.

4.1.1 Precision@K

Precision@K là tỷ lệ các mục có liên quan đến truy vấn được đề xuất trong bộ top-K. Ví dụ, nếu 5 trong số 10 câu hỏi được đề xuất là tương đồng hoặc liên quan đến câu hỏi truy vấn, thì Precision@10 là $\frac{5}{10}$. Precision@K được tính bởi công thức sau:

$$\text{Precision@}K = \frac{\text{Số lượng đề xuất có liên quan trong top } K}{\text{Số lượng mục được đề xuất } (K)} \tag{4.1}$$

4.1.2 Recall@K

Recall@K là tỷ lệ giữa các mục có liên quan đến truy vấn trong top-K với các mục có liên quan trong bộ dữ liệu. Ví dụ, nếu có tổng cộng 8 câu hỏi trong tập dữ liệu là tương đồng hoặc liên quan, và 5 trong số chúng được đề xuất trong top 10, thì Recall@10 là $\frac{5}{8}$. Trong khi Precision@K tập trung vào việc đo lường độ chính xác của mô hình trong top K, thì Recall@K tập trung vào việc đo lường khả năng của mô hình trong việc bao phủ tất cả các mục có liên quan khi chỉ xem xét top K mục được đề xuất. Recall@K được tính bởi công thức sau:

$$\text{Recall@}K = \frac{\text{Số lượng đề xuất có liên quan trong top } K}{\text{Số lượng mục liên quan trong bộ dữ liệu}} \tag{4.2}$$

Vì số lượng câu hỏi có liên quan đến câu hỏi người dùng là khác nhau, nên đối với Recall@K ta thực hiện tinh chỉnh, nếu số lượng mục liên quan trong bộ dữ liệu lớn hơn K, thì mẫu số bằng K. Điều này để đảm bảo rằng Recall@K luôn nằm trong khoảng từ 0 đến 1.

4.2 Môi trường thực nghiệm

Em thực hiện xây dựng và huấn luyện mô hình trên môi trường Colab và Kaggle. Một số thông số về cấu hình trong môi trường chạy:

1. Đối với Kaggle (dùng để chạy mô hình Vietnamese-BiEncoder của BKAI):

- GPU: Sử dụng NVIDIA P100, bộ nhớ 16 GB.
- CPU: RAM 29 GB.

2. Đối với Colab (dùng để huấn luyện mô hình E5 với SimCSE):

- GPU: Sử dụng T4, bộ nhớ 15 GB.
- CPU: RAM hệ thống 12.7 GB, ổ đĩa 78.2 GB.

4.3 Kịch bản huấn luyện

Việc tinh chỉnh/huấn luyện mô hình giống với mô tả ở chương Phương pháp đề xuất với các thông số:

- Số lượng epochs = 1
- Kích thước batch_size = 8
- Thời gian huấn luyện: khoảng 2.5 tiếng
- Tốc độ học: 5.10^{-5}

4.4 Kiểm tra mô hình với tập kiểm thử (test set) và đánh giá

Sử dụng tập kiểm thử để kiểm tra mô hình, vì các mẫu trong tập kiểm thử có thể xem như những mẫu mới mà mô hình chưa từng được nhìn thấy. Do một số vấn đề khách quan về tài nguyên, em chỉ thực hiện kiểm thử trên 13098 mẫu (chiếm 10% số lượng mẫu trong tập dữ liệu).

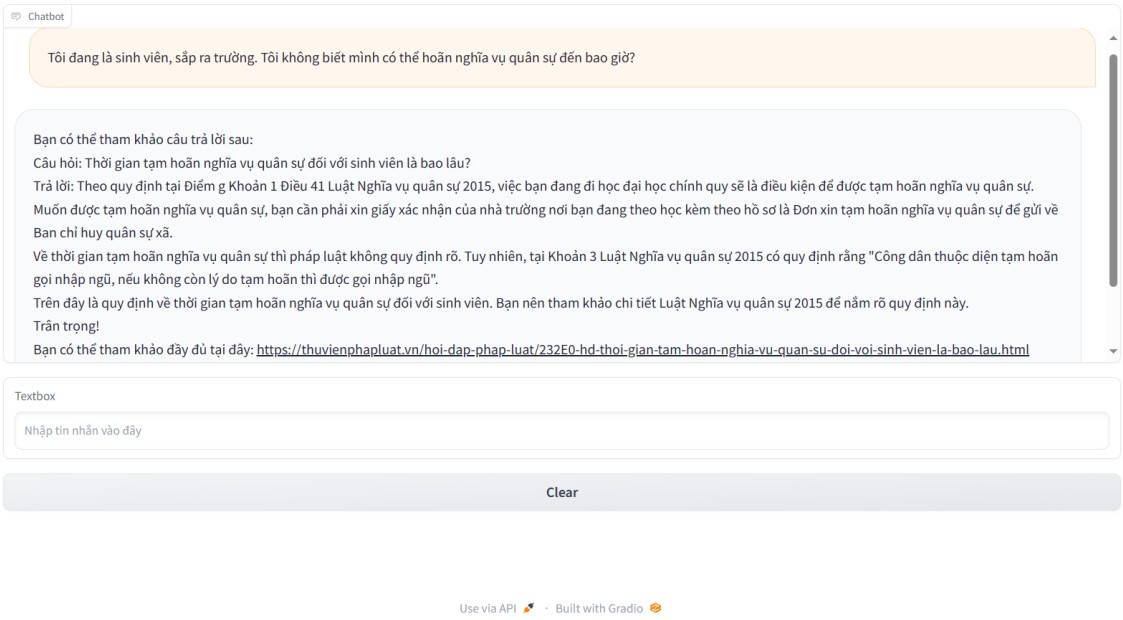
Phần kiểm tra mô hình sử dụng các tiêu chí Precision@5, Recall@5 và có thêm một chỉ số là **IsFirstRate** - chỉ số này cho biết tỷ lệ số mẫu có kết quả hiển thị top 1 là liên quan đến truy vấn. Kết quả được hiển thị trong bảng 4.1.

Bảng 4.1: Các chỉ số đánh giá mô hình đã được huấn luyện, trên tập kiểm thử (test set)

Mô hình	**Precision@5**	**Recall@5**	**IsFirstRate**
VN-BiEncoder	**0.1751**	**0.7364**	**0.6312**
$E5_{BASE}$ + SimCSE	0.1536	0.6283	0.5672

Sau đây là một số kết quả kiểm tra và đánh giá đối với cả hai mô hình.

1. Với mô hình đã xây dựng và số lượng câu hỏi kết quả đưa ra là 5, chỉ số Recall@5 cao cho thấy trong số 5 kết quả thực hiện thì số lượng các câu hỏi liên quan/có ích đến chủ đề được hỏi xuất hiện thường xuyên hơn.
2. Chỉ số Precision@5 thấp không đồng nghĩa với mô hình tệ, do có một số chủ



Hình 4.1: Ví dụ hiển thị kết quả đề xuất top 1 cho câu hỏi người dùng nhập vào.

đề (trường 'user_question') chỉ có 1 câu hỏi liên quan (do định dạng của trang web là cứ một câu hỏi người dùng sẽ có từ 1-4 câu hỏi con có nội dung liên quan), làm cho chỉ số này rất thấp.

3. Mô hình VN-BiEncoder tỏ ra ưu việt hơn so với $E5_{BASE}$ + SimCSE dù không được huấn luyện trên bộ dữ liệu đặc thù về câu hỏi pháp luật, vì dù gì đi chăng nữa, dữ liệu huấn luyện cho $E5_{BASE}$ + SimCSE và số lượng epoch còn khá khiêm tốn, trong khi mô hình VN-BiEncoder được huấn luyện trên vô cùng nhiều dữ liệu tiếng Việt và tốn rất nhiều tài nguyên, thời gian và công sức.

   Tuy nhiên, với kết quả trên cho thấy SimCSE học ngữ nghĩa rất tốt, dù dữ liệu huấn luyện và cách học khá đơn giản.

Hình 4.1 minh họa một kết quả đề xuất của mô hình $E5_{BASE}$ với SimCSE, sử dụng thư viện Gradio cho Python. Kết quả hiển thị khá tương đồng và phù hợp với nội dung câu hỏi nhập vào. Có một lưu ý, trong phần hiển thị này, em có đặt ngưỡng tương đồng là 0.2, nếu độ tương đồng nhỏ hơn 0.2 thì sẽ không đề xuất câu hỏi nào.

Tuy nhiên, mô hình $E5_{BASE}$ với SimCSE còn chưa hiểu rõ được ngữ cảnh do ảnh hưởng của các từ "gây nhiễu" không phải từ khóa chính. Ví dụ, với câu hỏi "*Tôi muốn đăng ký tạm trú phải làm thế nào?*", dưới đây là kết quả top 5 được đề xuất từ hai mô hình (xếp theo thứ tự giảm dần về độ tương đồng).
Với đề xuất từ **$E5_{BASE}$ + SimCSE:**

1. *Hướng dẫn thủ tục đăng ký tạm trú*

2. *Cách ghi sổ đăng ký tạm trú*
3. *Không đăng ký tạm trú có phải đi cai nghiện bắt buộc không?*
4. *Không đăng ký tạm trú đúng quy định bị xử phạt ra sao?*
5. *Không muốn đứng tên trong sổ đỏ thì phải làm thế nào?*

Với đề xuất từ **VN-BiEncoder:**

1. *Hướng dẫn thủ tục đăng ký tạm trú*
2. *Thủ tục đăng ký tạm trú tạm vắng*
3. *Cách ghi sổ đăng ký tạm trú*
4. *Phải khai báo tạm vắng mới được đăng ký tạm trú phải không?*
5. *Có buộc đăng ký tạm trú nhiều lần không?*

Ta có thể nhận thấy, rõ ràng kết quả top 1 là như nhau, tuy nhiên nhìn vào top 5 thì có sự khác biệt rất lớn. Trong khi mô hình VN-BiEncoder tỏ ra vượt trội khi các kết quả hiển thị đều có liên quan toàn bộ hoặc một phần với câu hỏi nhập vào, thì mô hình $E5_{BASE}$ + SimCSE lại đưa ra một vài đề xuất khá mơ hồ và không liên quan, chẳng hạn: "*Không đăng ký tạm trú có phải đi cai nghiện bắt buộc không?*", hay "*Không muốn đứng tên trong sổ đỏ thì phải làm thế nào?"*. Đây là một vấn đề cần xem xét trong tương lai để hướng tới việc cải thiện hiệu năng mô hình.

CHƯƠNG 5. KẾT LUẬN

5.1 Kết luận

Qua khảo sát thực trạng và những ứng dụng trong việc tìm kiếm câu hỏi tương đồng, em đã thực hiện đề tài "*Ứng dụng mô hình học sâu trong tìm kiếm câu hỏi tương đồng mức ngữ nghĩa*". Quá trình tìm hiểu cũng như những bước đầu trong việc xây dựng mô hình, bao gồm việc nghiên cứu cơ sở lý thuyết, xử lý dữ liệu, xây dựng, huấn luyện và đánh giá hiệu năng mô hình đã được trình bày đầy đủ trong báo cáo Project 3 này.

Nội dung đã đạt được

Project này đã đáp ứng được mục tiêu đã đặt ra. Cụ thể, em đã thực hiện thiết kế một mô hình có khả năng đề xuất các câu hỏi tương đồng trong bộ dữ liệu câu hỏi pháp luật. Trước hết, em đã thực hiện tìm hiểu sơ bộ về dữ liệu và tiến hành thu thập dữ liệu. Tiếp theo, em thực hiện phân tích sâu vào kiến trúc mô hình, các công cụ và nền tảng cần thiết để xây dựng mô hình. Sau khi thực hiện các nhiệm vụ trên, em đã tiến hành xây dựng và huấn luyện mô hình, từ kết quả huấn luyện để đánh giá hiệu năng mô hình.

Việc thực hiện Project 3 đã giúp em vận dụng được những kiến thức đã học trong chương trình Khoa học máy tính vào xây dựng một mô hình cụ thể, cho em cái nhìn cụ thể hơn về quy trình xây dựng và huấn luyện một mô hình học sâu, hiểu được tầm quan trọng của các bước tiến hành mà trước đây chỉ được nghe trong lý thuyết. Bên cạnh đó, việc thực hiện Project 3 cũng giúp em phát triển thêm các kỹ năng mềm, bao gồm kỹ năng phân chia thời gian, lập kế hoạch cụ thể cho từng công việc, kỹ năng tìm kiếm, nghiên cứu tài liệu và viết báo cáo. Những kiến thức trên là vô cùng quan trọng và quý báu, giúp em hoàn thiện bản thân hơn và phát triển công việc sau này.

Những khó khăn và hạn chế trong quá trình thực hiện bài tập lớn

Khó khăn lớn nhất mà em gặp phải là việc thu thập dữ liệu. Bởi lẽ, định dạng của dữ liệu trên trang web này không phải đồng nhất cho mọi trang chứa câu hỏi.

Hạn chế gặp phải là vấn đề về môi trường chạy. Do giới hạn bộ nhớ của Colab và Kaggle, nên em chỉ thực hiện huấn luyện trên số epoch và kích thước batch rất nhỏ. Điều này gây ảnh hưởng lớn đến kết quả huấn luyện, khiến cho mô hình không đạt được như kỳ vọng mà em đề ra ban đầu.

5.2 Hướng phát triển trong tương lai

Trong tương lai cần khắc phục được hạn chế về môi trường chạy, có thể bằng một số cách như: i) sử dụng thêm nhiều cạc (card) GPU với bộ nhớ cao và tốc độ xử lý mạnh, ii) trả phí để sử dụng tính năng mua thêm đơn vị GPU trên Colab, Kaggle hay các nền tảng khác.

Mô hình em đề xuất có thể sẽ không tốt hơn một số mô hình khác. Trong tương lai, có thể thực hiện thay đổi hoặc thêm một số tác vụ, cụ thể:

- Thu thập thêm nhiều dữ liệu hơn.
- Cấu hình để mô hình tập trung vào các từ khóa quan trọng để tránh bị ảnh hưởng bởi các từ gây nhiễu.
- Sử dụng các công cụ máy tìm kiếm (search engine) để đưa ra kết quả chính xác cho từng câu hỏi thay vì chỉ đề xuất câu hỏi tương đồng.
- Thử nghiệm trên các mô hình ngôn ngữ lớn tốt hơn với phương pháp học hiệu quả hơn.

TÀI LIỆU THAM KHẢO

[1] A. Vaswani, N. Shazeer, N. Parmar **andothers**, *Attention is all you need*, 2023. arXiv: 1706.03762 [cs.CL].

[2] J. Devlin, M.-W. Chang, K. Lee **and** K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, 2019. arXiv: 1810.04805 [cs.CL].

[3] T. Gao, X. Yao **and** D. Chen, *Simcse: Simple contrastive learning of sentence embeddings*, 2022. arXiv: 2104.08821 [cs.CL].

[4] L. Wang, N. Yang, X. Huang **andothers**, *Text embeddings by weakly-supervised contrastive pre-training*, 2022. arXiv: 2212.03533 [cs.CL].