HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Nguyễn Đình Quý

XÂY DỰNG MÔ HÌNH HỎI ĐÁP HỖ TRỢ SINH VIÊN TRƯỜNG ĐẠI HỌC XÂY DỰNG

Chuyên ngành: Khoa học máy tính

Mã số: 8.48.01.01

TÓM TẮT LUẬN VĂN THẠC SĨ KỸ THUẬT

HÀ NỘI – 2020

Luận văn được hoàn thành tại:

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Người hướng dẫn khoa học: **GS.TS Từ Minh Phương**

Phản biện 1: PGS.TS Bùi Thu Lâm

Phản biện 2: TS. Phùng Văn Ổn

Luận văn sẽ được bảo vệ trước Hội đồng chấm luận văn thạc sĩ tại Học viện Công nghệ Bưu chính Viễn thông

Vào lúc: …….giờ …… ngày…… tháng …… Năm ……

Có thể tìm hiểu luận văn tại:

Thư viên của Học viện Công nghệ Bưu chính Viễn thông

MỞ ĐẦU

Hiện nay trường đại học Xây dựng có khoảng 15.000 sinh viên và học viên đang theo học. Hàng ngày các phòng ban của trường nhận được rất nhiều các vấn đề thắc mắc của sinh viên và học viên về chương trình đào tạo, các thông tin về lịch học, lịch thi hay các quy định của nhà trường. Kênh thông tin chủ yếu của nhà trường là thông qua website chính thức hoặc trang quản lý đào tạo của sinh viên. Các quy định hay các thông báo tới sinh viên chủ yếu dưới dạng các văn bản nên gây khó khăn cho sinh viên trong việc tiếp cận và tra cứu thông tin. Chính vì thế khi có thắc mắc, sinh viên thường bỏ qua không đọc các văn bản hay thông báo mà sử dụng kênh hỗ trợ trực tiếp từ nhà trường, hiện tại là thông qua kênh email.

Một vấn đề đặt ra là số lượng email các câu hỏi của sinh viên gửi tới các phòng ban rất nhiều, một ngày có thể lên tới vài chục đến vài trăm câu hỏi. Vì vậy việc hỗ trợ sinh viên mà đặc biệt vào những dịp cao điểm như đăng ký môn học, thi hết học phần thường bị quá tải ở các phòng ban. Đồng thời sinh viên phải chờ đợi việc xử lý các câu hỏi và câu trả lời nên nhiều khi thông tin phản hồi không được kịp thời, gây ảnh hưởng đến quá trình học tập của sinh viên. Trong quá trình học tập của sinh viên, các nội dung liên quan đến quy định sẽ được thông báo dưới dạng văn bản hoặc tài liệu được đăng tải trên website đào tạo của nhà trường. Sinh viên quan tâm đến thông báo thường dựa trên tiêu đề thông báo, rồi sau đó mới đến nội dung thông báo, vì vậy nhiều thông báo bị sinh viên bỏ sót. Ngoài ra một số tài liệu quy định có nội dung dài nên sinh viên thường bỏ qua không đọc. Vì vậy nếu chỉ xây dựng hệ thống để quản lý văn bản, tài liệu để sinh viên tra cứu cũng không thật sự hữu ích với sinh viên. Cần phải xây dựng công cụ để tương tác với sinh viên dưới dạng đặt câu hỏi – trả lời mới giải quyết được vấn đề này.

Chính vì vậy, việc đưa ra một hệ thống trả lời câu hỏi tự động nhằm cung cấp cho sinh viên kênh hỗ trợ nhanh chóng, đồng thời làm giảm khối lượng công việc cho các phòng ban là vô cùng cần thiết. Một trong những kỹ thuật được sử dụng phổ biến hiện nay và mang lại hiệu quả cao là kỹ thuật truy xuất thông tin. Đề tài luận văn của em sẽ tập trung vào tìm hiểu các kỹ thuật này, dựa trên dữ liệu được cung cấp từ nhà trường để xây dựng hệ thống trả lời tự động có kết quả trả lời tốt nhất.

Nội dung của luận văn được bố cục thành 3 chương như sau:

- ***Chương 1*** tập trung vào giới thiệu về bài toán, dữ liệu đã có và kết quả dự kiến của đề tài. Trình bày khái về hệ thống hỏi đáp tự động, các loại hệ thống hỏi đáp, lịch sử phát

triển, đưa ra kiến trúc chung của hệ thống hỏi đáp đồng thời là các vấn đề cần quan tâm khi thiết kế.

- ***Chương 2*** tập trung vào lựa chọn mô hình và thuật toán để xây dựng mô hình hệ thống hỏi đáp. Trình bày về việc tìm hiểu các phương pháp tiền xử lý dữ liệu bao gồm: tách từ tiếng Việt, các hướng tiếp cận dựa trên từ và dựa trên ký tự; biểu diễn văn bản; rút trích đặc trưng văn bản như loại bỏ các stop word, trích chọn đặc trưng văn bản thành các biểu diễn của các vector; tiếp theo là đưa ra mô hình kiến trúc của hệ thống và kỹ thuật được sử dụng trong luận văn;
- ***Chương 3*** tập trung vào cài đặt, xây dựng bộ dữ liệu huấn luyện cho mô hình hỏi đáp từ dữ liệu thực tế hiện có của trường Đại học Xây dựng, sử dụng các kỹ thuật đánh giá mô hình hỏi đáp để đánh giá hệ thống, tiếp theo là tiến hành thử nghiệm tại trường để tiếp nhận những đánh giá từ người dùng cuối.

CHƯƠNG 1. TỔNG QUAN VỀ BÀI TOÁN HỎI ĐÁP TỰ ĐỘNG

1.1. Bài toán trả lời tự động cho sinh viên trường Đại học Xây dựng

Với thực trạng tại trường Đại học Xây dựng, hàng ngày sinh viên hỏi và thắc mắc rất nhiều vấn đề liên quan đến các chính sách, quy định và quy chế. Nhà trường phải bố trí bộ phận hỗ trợ sinh viên để giải đáp các thắc mắc và giúp đỡ sinh viên khi cần thiết, hiện tại bộ phận này sẽ tiếp nhận các câu hỏi của sinh viên qua kênh email sau đó trả lời các email đó. Tuy nhiên vấn đề vào các đợt cao điểm như đăng ký môn học hay thi kết thúc học phần thì số lượng các câu hỏi tăng đột biến làm quá tải cho bộ phận hỗ trợ. Hơn nữa rất nhiều các câu hỏi thường lặp lại và được trả lời giống nhau, bộ phận hỗ trợ thường dựa vào các câu trả lời trước đó đã phản hồi để trả lời các câu hỏi tương tự.

Giả sử như nếu sinh viên hỏi một trong các câu hỏi sau đây:

1. *E thưa cô, chả hạn e trả hết môn mà tích luỹ chưa đủ 2.0 thì e có dc nhận để làm đồ án tốt nghiệp không ạ*
2. *Điều kiện để nhận ĐATN là gì ạ?*
3. *Em đã hoàn thiện hết các môn nhưng chưa đủ tiêu chuẩn ngoại ngữ thì có được nhận ĐATN không ạ?*
4. *Điểm trung bình tích lũy bao nhiêu thì được nhận đồ án tốt nghiệp ạ*

Thì đều được trả lời là: "*Em trả nợ xong tất cả các môn và đạt CĐR ngoại ngữ là đủ điều kiện nhận ĐATN. Điểm TBC tích lũy từ 2.0 trở lên là điều kiện xét TN, không áp dụng khi xét giao ĐATN"*. Như vậy là khi sinh viên hỏi một câu hỏi nào đó mà tương tự với các câu hỏi đã có thì có thể trả lời bằng câu trả lời có sẵn.

Sau một thời gian trả lời qua email, bộ phận công tác sinh viên đã thu thập được một bộ các câu hỏi của sinh viên và câu trả lời do cơ quan chức năng của trường gửi lại gồm khoảng 3.500 câu hỏi, câu trả lời. Dựa trên tập câu hỏi, câu trả lời này, bài toán mà luận văn hướng tới giải quyết là xây dựng hệ thống cho phép tự động trả lời câu hỏi của sinh viên trong tương lai.

Kết quả dự kiến của luận văn: Luận văn này sẽ dựa vào một tập dữ liệu có sẵn gồm các câu hỏi và câu trả lời để xây dựng công cụ trả lời tự động các câu hỏi giống với các câu hỏi đã có trong tập dữ liệu.

1.2. Khái quát hệ thống hỏi đáp tự động

Nếu như trong hệ thống trích chọn thông tin khi người dùng muốn tìm kiếm thông tin họ cần, hệ thống trích chọn thông tin sẽ nhận truy vấn đầu vào của người dùng dưới dạng các từ khóa và trả về các tài liệu liên quan có chứa từ khóa thì hệ thống hỏi đáp sẽ nhận đầu vào dưới dạng ngôn ngữ tự nhiên (thường là các câu hỏi), sau đó trả lại câu trả lời tương ứng với câu hỏi đưa vào.

Đối với bài toán cần giải quyết, tận dụng dữ liệu của bài toán gồm các câu hỏi và câu trả lời có sẵn nên luận văn này sẽ sử dụng phương pháp trả lời tự động dựa *trên truy xuất thông tin (IR).*

1.3. Truy xuất và tìm kiếm thông tin (IR)

Hệ truy xuất thông tin (IR) xuất hiện trong các hệ thống thông minh từ những năm 1960, hệ thống tìm kiếm trên máy tính sớm nhất được ra đời vào cuối những năm 1940.

1.3.1. Mô hình dựa trên lý thuyết tập hợp:

Mô hình lý thuyết tập hợp biểu diễn tài liệu dưới dạng tập hợp các từ hoặc cụm từ. Từ đó, các phép toán dùng để tính độ tương tự thường sử dụng các phép toán dựa trên lý thuyết tập hợp. Các mô hình phổ biến thuộc loại này là: Mô hình Boolean chuẩn, mô hình Boolean mở rộng và mô hình Truy xuất mờ [8].

1.3.2. Mô hình đại số

Mô hình đại số biểu diễn các tài liệu và truy vấn dưới dạng vectơ, ma trận hoặc bộ giá trị. Một số mô hình thuộc loại này là: Mô hình không gian vectơ, mô hình Không gian vectơ tổng quát hóa, mô hình Không gian vectơ dựa trên chủ đề, mô hình Boolean mở rộng.

1.3.3. Mô hình xác suất

Cho câu truy vấn của người dùng *q* và văn bản *d* trong tập văn bản. Mô hình xác suất tính xác suất mà văn bản *d* liên quan đến cấu truy vấn của người dùng. Mô hình giả thiết xác suất liên quan của một văn bản với câu truy vấn phụ thuộc cách biểu diễn chúng. Tập văn bản kết quả được xem là liên quan và có tổng xác suất liên quan với câu truy vấn lớn nhất.

1.3.4. Mô hình ngôn ngữ

Mô hình ngôn ngữ là tập hợp các kiến thức trước đó về một ngôn ngữ nhất định, các kiến thức này có thể là các kiến thức về từ vựng, về ngữ pháp, về tần suất xuất hiện của các cụm từ, ... Một mô hình ngôn ngữ có thể được xây dựng theo hướng chuyên gia hoặc hướng dữ liệu.

1.4. Kết luận chương

Một câu hỏi có thể được trả lời bằng cách tìm xem nó giống với câu hỏi nào trong bộ dữ liệu câu hỏi – câu trả lời có sẵn. Bằng cách này sinh viên có thể nhận được câu trả lời ngay sau khi hỏi mà không phải chờ đợi người hỗ trợ trả lời từng câu hỏi.

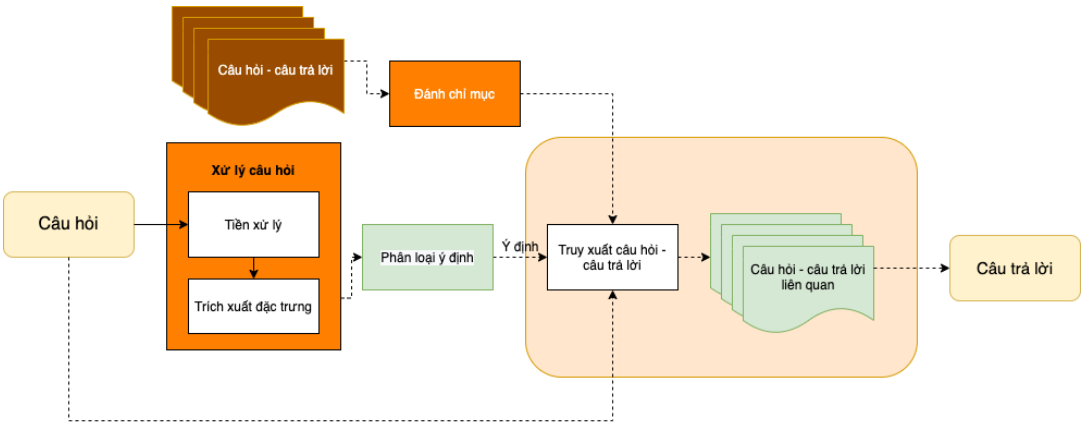
Có rất nhiều phương pháp trả lời tự động nhưng trong bài toán này, dựa vào đặc trưng của bài toán cần giải quyết và dữ liệu của bài toán nên luận văn này sẽ sử dụng phương pháp trả lời tự động dựa trên *truy xuất thông tin (IR).* Phương pháp này sẽ tận dụng được các câu hỏi và câu trả lời có sẵn trong tập dữ liệu đã xây dựng.

CHƯƠNG 2. PHƯƠNG PHÁP TRẢ LỜI TỰ ĐỘNG

Chương này trình bày về phương pháp trả lời tự động do học viên lựa chọn và phát triển dựa trên một số giải pháp đã có. Trước hết là kiến trúc chung của mô hình trả lời tự động, sau đó là mô tả chi tiết của từng thành phần trong mô hình. Mô hình được xây dựng để tận dụng bộ câu hỏi câu trả lời tích luỹ được tại các phòng chức năng của trường Đại học Xây dựng.

2.1. Kiến trúc mô hình

Bài toán lúc này được đặt ra như sau: có một người hỏi một câu hỏi *a*, sau đó hệ thống sẽ tìm kiếm câu hỏi *a* trong tập dữ liệu D gồm các câu hỏi - câu trả lời có sẵn đã được xây dựng từ trước. Hệ thống cần đưa ra cặp câu hỏi – câu trả lời trong D được xếp hạng cao nhất theo mức độ liên quan đến câu hỏi *a* và lấy câu trả lời ra làm câu trả lời cho câu hỏi a. Vì các câu hỏi của sinh viên hầu hết thường lặp đi lặp lại trong tập dữ liệu D nên chúng ta có thể sử dụng phương pháp truy xuất thông tin IR để xác định mức độ liên quan giữa các câu hỏi.



Hình 2.1: Từ câu hỏi đến câu trả lời: Mô hình xây dựng hệ thống hỏi đáp

Trong hình 2.1, hệ thống trả lời tự động có 2 thành phần chính:

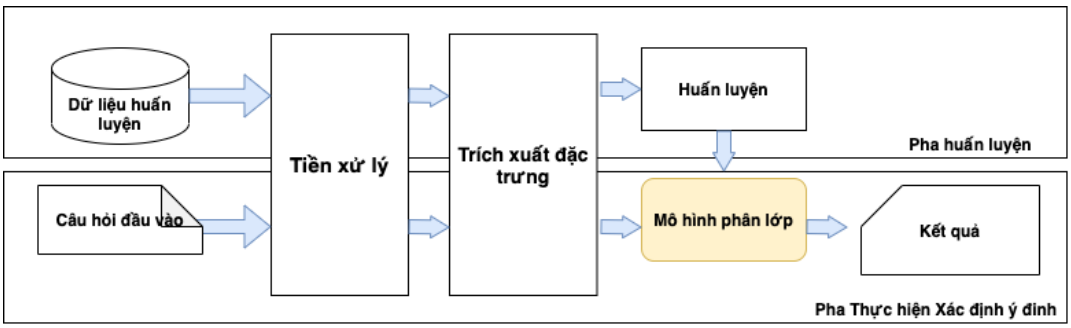
- Module *xác định ý định câu hỏi* sử dụng mô hình học sâu để xác định ý định của câu hỏi, module này giống như một bộ phân loại văn bản với đầu vào là câu hỏi và đầu ra là lớp được phân loại, mỗi lớp đầu ra tương ứng với một ý định của câu hỏi.
- Module *truy xuất thông tin* để tìm kiếm câu trả lời phù hợp với câu hỏi.

2.2. Phân loại ý định

Phân loại ý định (intent classification) là việc xác định ý định của người hỏi khi tương tác với hệ thống hỏi đáp thông qua câu hỏi hay câu truy vấn của người dùng.

2.2.1. Luồng xử lý phương pháp xác định ý định của câu hỏi

[Hình 2.2] mô tả luồng xử lý của bài toán phân loại ý định của câu hỏi. Trong hình vẽ chia ra thành 2 pha: pha huấn luyện mô hình và pha áp dụng mô hình vào để dự đoán.



Hình 2.2: Thuật toán phân lớp ý định của câu hỏi

Pha huấn luyện được thực hiện như sau:

a. Tiền xử lý

Dữ liệu trong tập dữ liệu huấn luyện sẽ được đưa qua bước *tiền xử lý*. Bước này sẽ tiến hành một số kỹ thuật sau đây:

- *Chuẩn hóa câu hỏi*
- *Tách từ Tiếng Việt*
- *Loại bỏ từ dừng*

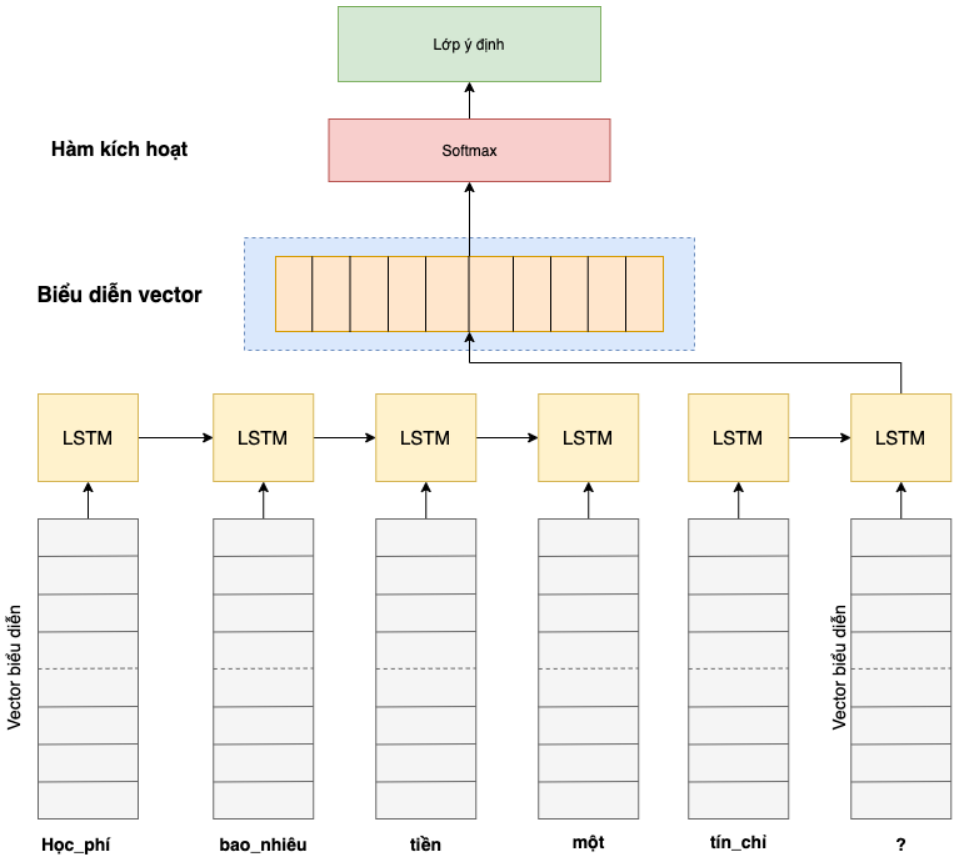
Sau khi thực hiện xong quá trình tiền xử lý, văn bản sẽ được đưa vào bước 2.

b. Trích xuất đặc trưng

Bước này sẽ thực hiện nhiệm vụ đưa văn bản thành các vector biểu diễn. Dựa trên vector biểu diễn từ có thể gộp lại để tạo thành biểu diễn văn bản. Vector biểu diễn văn bản có chiều dài cố định, văn bản nào có chiều dài nhỏ hơn chiều dài này sẽ được thêm padding để đưa về ma trận biểu diễn có kích thước giống nhau giữa các văn bản.

c. Mô hình phân lớp ý định

Mô hình phân lớp ý định sử dụng mạng nơ-ron học sâu LSTM được mô tả chi tiết như [hình 2.3].



Hình 2.3: Mô hình phân lớp ý định câu hỏi

2.2.2. Tiền xử lý dữ liệu

a. Phương pháp tách từ tiếng việt

Tách từ, về mặt biểu hiện, là gom nhóm các từ đơn liền kề thành một cụm từ có ý nghĩa.

b. Phương pháp loại bỏ từ dừng

Từ dừng là những từ trong bất kỳ ngôn ngữ nào không bổ sung nhiều ý nghĩa cho một câu. Chúng có thể được bỏ qua một cách an toàn mà không làm mất đi ý nghĩa của câu. Các từ dừng thường bị xóa khỏi văn bản trước khi đào tạo mô hình học sâu và học máy vì các từ dừng xuất hiện rất nhiều và không mang ý nghĩa, do đó cung cấp rất ít hoặc không có thông tin có thể được sử dụng để phân loại hoặc phân cụm.

2.2.3. Trích xuất đặc trưng

Trích xuất đặc trưng là tìm cách đưa văn bản về biểu diễn dưới dạng vector hay ma trận mà các biểu diễn này vẫn thể hiện được các đặc trưng của văn bản.

c. One-hot encoding (tạm gọi mã hoá số 1):

Đây là cách đơn giản để biểu diễn ngôn ngữ sang dạng vector với số chiều là kích thước từ điển. Giống như tên của nó, chỉ ở chiều mà vị trí một từ xuất hiện trong từ điển có giá trị là 1. Các chiều khác đều có giá trị là 0.

tf-idf Vector quan tâm cả tần số xuất hiện của từ trong toàn bộ tập dữ liệu, chính do đặc điểm này mà tf-idf Vector có tính phân loại cao hơn so với Count Vector. tf-idf (Term Frequency-Inverse Document Frequency) Vector là một vector số thực cũng có độ dài D với D là số văn bản, nó được tính bằng tích của 2 phần bao gồm tf và idf, công thức của mỗi phần tử của vector được tính như sau:

TF- term frequency – tần số xuất hiện của 1 từ trong 1 văn bản. Cách tính:

$$\mathrm{tf}(t,d) = \frac{\mathrm{f}(t,d)}{\max\{\mathrm{f}(w,d) : w \in d\}}$$

Thương của số lần xuất hiện 1 từ trong văn bản và số lần xuất hiện nhiều nhất của một từ bất kỳ trong văn bản đó. (giá trị sẽ thuộc khoảng [0, 1])

- *f(t,d)* - số lần xuất hiện từ t trong văn bản d.
- *max{f(w,d):w∈d}* - số lần xuất hiện nhiều nhất của một từ bất kỳ trong văn bản.

DF – inverse document frequency. Tần số nghịch của 1 từ trong tập văn bản (corpus).

Tính IDF để giảm giá trị của những từ phổ biến. Mỗi từ chỉ có 1 giá trị IDF duy nhất trong tập văn bản.

$$\mathrm{idf}(t,D) = \log\frac{|D|}{|\{d \in D : t \in d\}|}$$

- $|D|$ - tổng số văn bản trong tập D
- $\{d \in D : t \in d\}$: - số văn bản chứa từ nhất định, với điều kiện t xuất hiện trong văn bản d. Nếu từ đó không xuất hiện ở bất cứ 1 văn bản nào trong tập thì mẫu số sẽ bằng 0 => phép chia cho không không hợp lệ, vì thế người ta thường thay bằng mẫu thức $1 + |\{d \in D : t \in d\}|$.

d. Ma trận đồng xuất hiện

Co-occurrence Matrix có ưu điểm là bảo tồn mối quan hệ ngữ nghĩa giữa các từ, được xây dựng dựa trên số lần xuất hiện của các cặp từ trong ***Context Window***. Một ***Context Window*** được xác định bởi kích thước và hướng của nó.

e. Word embeddings (Tập nhúng từ)

Do các vấn đề của ma trận đồng xuất hiện nên đã có nhiều nghiên cứu hướng theo giải pháp học biểu diễn ở số chiều thấp hơn. Từ thời điểm này hàng loạt bài toán NLP được giải quyết với độ chính xác cao hơn nhiều so với trước.

Dựa vào ưu và nhược điểm của từng phương pháp cùng với yêu cầu của mô hình phân loại ý định, luận văn này sẽ sử dụng mô hình Skip-gram để thực hiện bước trích xuất đặc trưng :

Skip-Gram Model Sử dụng cửa sổ trượt với kích thước cố định để di chuyển từ trái qua phải của câu. Từ ở giữa là "target" và các từ bên trái và phải trong cửa sổ đó là các từ thể hiện ngữ cảnh. Mô hình skip-gram được huấn luyện để dự đoán xác suất của từ theo ngữ cảnh đưa ra.

2.2.4. Mô hình phân lớp

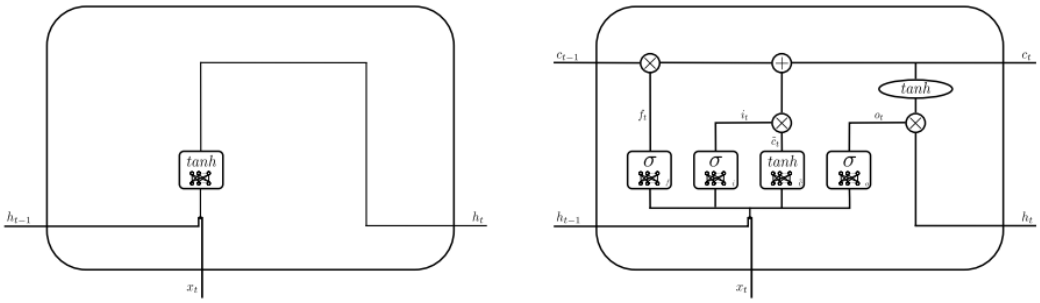
Để phân loại văn bản với đầu vào là các vector biểu diễn, luận văn sẽ sử dụng mạng nơ-ron học sâu để tiến hành phân lớp. Có nhiều kiến trúc mạng nơ-ron khác nhau, để giải quyết bài toán này tôi sử dụng Mạng bộ nhớ dài-ngắn (Long Short Term Memory networks) là một dạng của mạng hồi quy RNN.

a. RNN

Mạng này chứa các vòng lặp bên trong cho phép thông tin có thể lưu lại được. Đôi lúc ta chỉ cần xem lại thông tin đằng trước là đủ để biết được tình huống hiện tại. Trong tình huống này, khoảng cách tới thông tin có được cần để dự đoán là nhỏ, nên RNN hoàn toàn có thể học được. Một vấn đề gặp phải đối với mạng RNN đó là việc ghi nhớ xa. Trong nhiều tình huống ta buộc phải sử dụng nhiều ngữ cảnh hơn để suy luận.

b. Mạng LSTM

Mạng bộ nhớ dài-ngắn (Long Short Term Memory networks), thường được gọi là LSTM - là một dạng đặc biệt của RNN, nó có khả năng học được các phụ thuộc xa của ngữ cảnh.



(a) RNN (b) LSTM

Hình 2.4: Biểu diễn của mô hình LSTM và RNN

LSTM được thiết kế để tránh được vấn đề phụ thuộc xa (long-term dependency). Việc nhớ thông tin trong thời gian dài là đặc tính mặc định của chúng, chứ ta không cần phải huấn luyện nó để có thể nhớ được. Tức là ngay nội tại của nó đã có thể ghi nhớ được mà không cần bất kì can thiệp nào.

2.2.5. Tăng cường dữ liệu để huấn luyện mô hình phân lớp ý định

Sau khi trình bày các kỹ thuật để xác định ý định của câu hỏi tôi sẽ trình bày một kỹ thuật được áp dụng để bổ sung dữ liệu huấn luyện cho mô hình phân lớp ý định để tăng độ chính xác của mô hình. Và phần này sẽ tập trung chính vào kỹ thuật sinh câu hỏi tương ứng với ý định của người dùng để bổ sung thêm dữ liệu vào tập dữ liệu huấn luyện.

Tất cả các câu hỏi của sinh viên trong trường được chia ra thành các class như sau, mỗi class tương ứng với ý định hỏi của người dùng. Như vậy việc xác định ý định chính là việc phân lớp 1 câu hỏi thuộc vào class nào:

Để áp dụng kỹ thuật tăng cường dữ liệu cho bài toán này tôi sử dụng BERT được trình bày chi tiết dưới đây.

a. BERT

BERT là viết tắt của Bidirectional Encoder Representations from Transformers [22] là một mô hình ngôn ngữ được huấn luyện dựa trên tập dữ liệu văn bản rất lớn, mô hình học được cách biểu diễn vector của các từ theo ngữ cảnh 2 chiều của từ, thường được sử dụng để transfer sang các bài toán khác trong lĩnh vực xử lý ngôn ngữ tự nhiên. BERT đã thành công trong việc cải thiện những tác vụ gần đây trong việc tìm ra biểu diễn của từ trong không gian thông qua ngữ cảnh của nó.

b. Các nhiệm vụ của BERT

Masked LM

Để đào tạo một mô hình tìm ra biểu diễn từ dựa vào ngữ cảnh 2 chiều, chúng ta sử dụng một cách tiếp cận đơn giản để che giấu đi một số token đầu vào một cách ngẫu nhiên và sau đó chúng ta chỉ dự đoán các token được giấu đi đó và gọi nhiệm vụ này như là một "masked LM"(MLM).

Dự đoán câu tiếp theo

Để đào tạo được mô hình hiểu được mối quan hệ giữa các câu, chúng ta cần xây dựng một mô hình dự đoán câu tiếp theo dựa vào câu hiện tại, dữ liệu huấn luyện có thể là một corpus bất kỳ nào. Cụ thể, khi chọn câu A và câu B cho mỗi training sample, 50% khả năng câu B là câu tiếp theo sau câu A và 50% còn lại là một câu ngẫu nhiên nào đó trong corpus.

c. Kiến trúc mô hình

Trong mô hình transformer, đây là một lớp mô hình seq2seq gồm 2 pha là encoder và decoder. Mô hình hoàn toàn không sử dụng các kiến trúc Recurrent Neural Network của RNN mà chỉ sử dụng các layers attention để embedding các từ trong câu.

d. phoBERT [22]

phoBERT là mô hình dựa trên BERT nhưng được huấn luyện bằng tập dữ liệu Tiếng Việt rất lớn. Để thực hiện tạo thêm dữ liệu huấn luyện tôi tiến hành hiệu chỉnh lại mô hình BERT, tôi có sử dụng lại các trọng số đã được huấn luyện cho tập dữ liệu Tiếng Việt phoBERT [22]

e. Các bước thực hiện

Để thực hiện tăng cường dữ liệu cho mô hình phân loại ý định tôi thực hiện như sau [23]:

- ***Bước 1:*** Giả sử ta có mô hình phân lớp như đã được huấn luyện trong phần 3.3. Gọi mô hình giả thiết này là $h = \mathcal{A}(D_{train})$ dựa trên tập dữ liệu có sẵn là D_{train}. Mô hình này sẽ được sử dụng để lọc kết quả trong bước 4.
- ***Bước 2:*** **Fine-tune lại mô hình ngôn ngữ**

  Bước này tiến hành độc lập với bước 1, tôi fine-tune mô hình ngôn ngữ $\mathcal{G}$ với nhiệm vụ tổng hợp các câu được gán nhãn, để thu được mô hình ngôn ngữ tinh chỉnh $\mathcal{G}_{tuned}$. Ở đây, $\mathcal{G}$ được fine-tune cụ thể theo miền ngôn ngữ của D_{train} (gồm các câu, từ vựng, văn phong, v.v.), cũng như các lớp trong D_{train}.

Mục tiêu của việc fine-tune đó là chúng ta có thể sử dụng $\mathcal{G}_{tuned}$. để tạo một tập hợp câu có kích thước bất kỳ và mỗi câu được gắn nhãn bằng một lớp.

Ở đây tôi sử dụng $\mathcal{G}$ là mô hình BERT, tôi tiến hành fine-tune lại BERT với dữ liệu huấn luyện $D_{train} = \{(x_i, y_i)\}_{i=1}^{n}$. Sau đó tôi tiến hành nối các câu trong D_{train} lại để được U có dạng như sau:

$$U^* = y_1 SEP x_1 EOS y_2 SEP x_2 EOS y_3 SEP x_3 \dots y_n SEP x_n EOS$$

Ở đây tôi sử dụng token `SEP` để phân cách giữa nhãn của câu và câu tương ứng. Token `EOS` dúng để kết thúc một câu và tách nó khỏi nhãn của câu sau.

Để huấn luyện tôi vẫn sử dụng hàm mất mát được dùng trong mô hình BERT:

$$J_\theta = -\sum_j log P_\theta(\omega^j \mid \omega^{j-k}, \dots, \omega^{j-1})$$

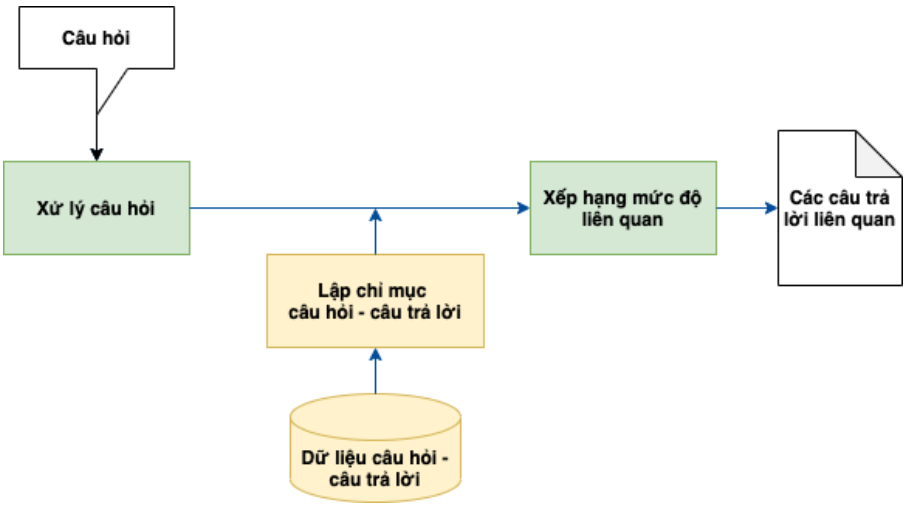
Các lớp mạng cách huấn luyện mô hình được giữ nguyên, tuy nhiên tôi sử dụng tập dữ liệu U^* thay vì U để huấn luyện cho mô hình này. Sau quá trình huấn luyện chúng ta được mô hình $\mathcal{G}_{tuned}$

- ***Bước 3:*** **Sinh ra câu mới.** Từ mô hình $\mathcal{G}_{tuned}$ tôi tiến hành sinh ra các câu mới bằng cách: Với mỗi nhãn thuộc vào trong tập nhãn y ∈ {1, ..., q} (với q là số lượng các nhãn), chúng ta có thể sử dụng mô hình ngôn ngữ $\mathcal{G}_{tuned}$ để dự đoán từ tiếp theo của chuỗi "y SEP" cho đến khi gặp token EOS thì dừng lại và kết thúc câu đã tạo. Thực hiện tương tự theo cách này cho mỗi lớp, ta có thể sinh ra được số lượng câu bất kỳ. Tôi đã tiến hành sinh ra các câu để cân bằng dữ liệu giữa các lớp và tạo ra thêm dữ liệu cho các lớp. Gọi tập dữ liệu được sinh ra là $D^* = \{(x_i', y_i')\}_{i=1}^{N}$.
- ***Bước 4:*** **Lọc lại các câu đã được sinh ra.** Từ các dữ liệu được sinh ra ở bước số 3 tôi sẽ tiến hành đưa vào mô hình h trong bước 1 để loại bỏ những câu kém chất lượng. Bằng cách đưa từng câu x_i' vào h, ta có $\hat{y}_i' = h(x_i')$. Nếu $\hat{y}_i' = y_i'$ nghĩa là nhãn dự đoán của mô hình và nhãn thực tế giống nhau thì ta sẽ bổ sung câu nay vào tập $D_{synthesized}$. Như vậy $D_{synthesized} \subseteq D^*$.

Như vậy sau bước sinh dữ liệu ta có tập dữ liệu $D_{new} = D_{train} \cup D_{synthesized}$.

2.3. Tìm kiếm và truy xuất thông tin.

Khi chúng ta tìm kiếm tài liệu, chúng ta muốn hệ thống IR truy xuất các tài liệu giống với truy vấn của chúng ta.



Hình 2.5: Kiến trúc mô hình truy xuất thông tin

2.3.1. Một số khái niệm

Thuật ngữ (term): Dùng để chỉ thành phần của một truy vấn, ví dụ ta có truy vấn: "Thủ đô của Hà Nội là gì", thuật ngữ của truy vấn sẽ là: 'Thủ đô', 'của', 'Hà Nội'. Hiểu đơn giản, thuật ngữ là các từ trong truy vấn/văn bản mang ý nghĩa.

- Tài liệu: Các văn bản thông thường cần tìm kiếm, truy vấn cũng có thể coi là tài liệu.
- Tần suất thuật ngữ hay còn gọi là tf: tần suất thuật ngữ xuất hiện trong tài liệu? 3 lần? 10 lần?
- Tần suất tài liệu nghịch đảo hay còn gọi là idf: được tính bằng số lượng tài liệu mà thuật ngữ xuất hiện. Tần suất tài liệu nghịch đảo (1 / df) cho biết mức độ quan trọng của thuật ngữ.

2.3.2. Công thức tính BM25

Để xác định mức độ liên quan giữa một truy vấn (tài liệu) với một tài liệu khác, chúng ta có thể sử dụng công thức tính BM25 như sau:

$$\text{BM25}(D,Q) = \sum_{i=1}^{n} IDF(q_i,D)\frac{f(q_i,D)\cdot(k_1+1)}{f(q_i)+k_1\cdot(1-b+b\cdot|D|/d_{avg}))}$$

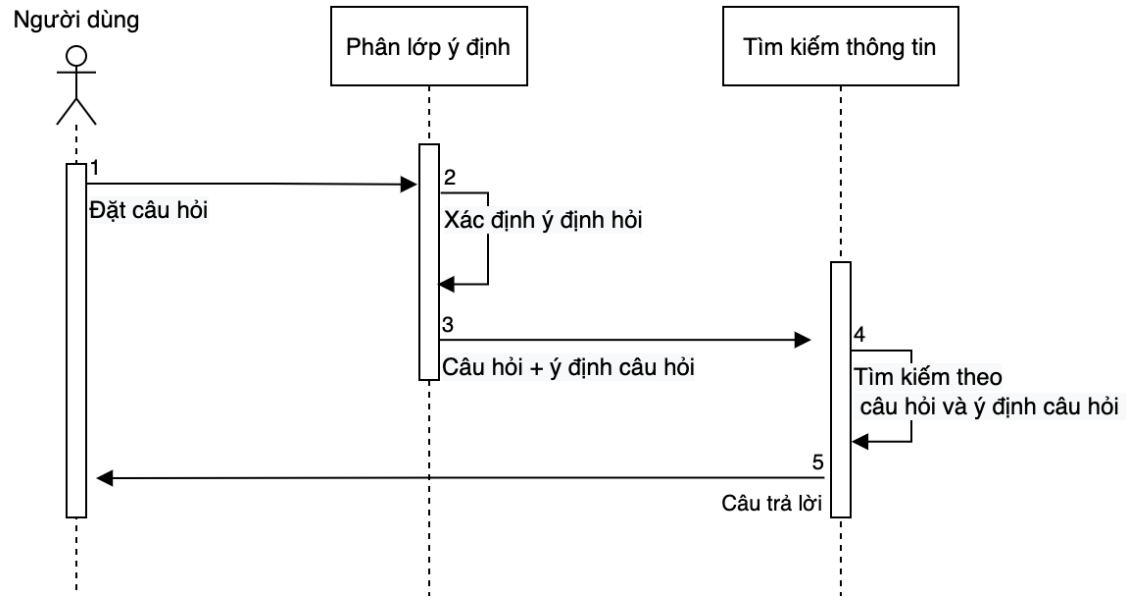
Trong đó:

- $f(q_i,D)$ Là số lần mà term qi xuất hiện trong tất cả các tài liệu D
- $|D|$ là số từ trong tất cả các tài liệu D
- d_{avg} là số lượng từ trung bình trong mỗi tài liệu

- b và k_1 là các tham số của BM25
- $f(q_i,D)$ cho ta thấy rằng nếu một từ xuất hiện trong tài liệu càng nhiều thì điểm của tài liệu càng cao.

  Với tham số k1, xác định tính bão hòa tần suất.

2.4. Kết hợp xác định ý định và truy xuất thông tin



Hình 2.6: Biểu đồ tuần tự các bước kết hợp xác định ý định và truy xuất thông tin

Hình 2.11 mô tả các bước của mô hình trả lời tự động bằng cách kết hợp xác định ý định câu hỏi và truy xuất thông tin. Các bước trong hình được mô tả như sau:

- *Bước 1:* Người dùng (là sinh viên) đặt câu hỏi cho hệ thống dưới dạng ngôn ngữ tự nhiên.
- *Bước 2 + 3:* Câu hỏi sẽ được đưa vào module phân lớp ý định để tìm ra ý định của câu hỏi. Sau khi xác định được ý định của câu hỏi, hệ thống tiếp tục đưa câu hỏi và ý định của người hỏi sang module truy xuất thông tin ở bước số 4.
- *Bước 4:* Tại bước 4 module truy xuất thông tin tiến hành tìm trong tập dữ liệu câu hỏi – câu trả lời có sẵn thỏa mãn điều kiện: 1 – câu hỏi trong tập dữ liệu phải thuộc loại ý định trong bước 2 và 2 - câu hỏi của người dùng phải gần giống nhất với câu hỏi và câu trả lời trong tập dữ liệu. Bước này sẽ xác định ra cặp câu hỏi – câu trả lời trong tập dữ liệu phù hợp với câu hỏi của người dùng.
- *Bước 5:* Từ cặp câu hỏi – câu trả lời tìm ra ở bước 4, hệ thống sẽ lấy câu trả lời để làm câu trả lời cho câu hỏi ở bước số 1.

2.4.1. Tổ chức dữ liệu để tìm kiếm thông tin theo ý định

Để tổ chức cấu trúc dữ liệu cho mô hình IR phù hợp với việc tìm kiếm theo ý định, mỗi cặp câu hỏi (q) và câu trả lời (t) sẽ dưới dạng d = {q, t, i}, trong đó *i* là ý định của câu hỏi. Mỗi cặp câu hỏi – câu trả lời trước khi đưa vào tập dữ liệu câu hỏi - câu trả lời *D* sẽ được gán nhãn ý định *i* bằng tay.

2.4.2. Tìm kiếm theo ý định và câu hỏi

Việc kết hợp *module xác định ý định câu hỏi* và *module truy xuất thông tin* giúp cho mô hình tìm kiếm chính xác hơn.

Giả sử người dùng có một câu hỏi a, sau khi đưa a vào mô hình xác định ý định câu hỏi, ta xác định được ý định của câu hỏi là *i'*. Thay vì tìm kiếm trong tập D câu hỏi giống với câu hỏi a thì ta sẽ tìm câu hỏi a trong tập D' với D' $\subset$ D, D' là tập câu hỏi - câu trả lời chỉ chứa các ý định *i'*.

Như vậy với câu hỏi a, tập dữ liệu D gồm các câu hỏi - câu trả lời có sẵn đã được xây dựng từ trước. Để tìm ra câu hỏi $d \in D$ sao cho d giống với a nhất với i là ý định của câu hỏi:

$$P(d \mid D, a) = \sum_{i \in I} P(i \mid a)\, P(d \mid D, a, i)$$

Với $P(i \mid a)$ là xác suất câu hỏi được phân loại là i được tính thông qua mô hình phân loại ý định. Vì mỗi câu hỏi chỉ mang 1 ý định nên $P(i \mid a) = 1$ nếu câu hỏi mang ý định i và $\mathrm{P(i \mid a)} = 0$ nếu câu hỏi không mang ý định i. $P(d \mid D,\ a,\ i)$ được tính thông qua mô hình truy xuất thông tin IR.

Như vậy trong chương 2 luận văn đã trình bày về kiến trúc chung của mô hình hỏi đáp sau đó đi sâu vào trình bày các kỹ thuật đã sử dụng để xây dựng nên mô hình bao gồm kỹ thuật xác định ý định của câu hỏi, kỹ thuật tìm kiếm và truy xuất thông tin IR. Trong chương 3 luận văn sẽ tiếp tục trình bày về các thực nghiệm và kết quả dựa trên các kỹ thuật đã thực hiện trong chương 2.

CHƯƠNG 3. THỰC NGHIỆM VÀ KẾT QUẢ

Chương này tập trung vào trình bày chi tiết các thực nghiệm và kết quả thực nghiệm trong quá trình xây dựng mô hình. Nội dung cụ thể sẽ được trình bày trong các phần dưới đây.

3.1. Các bước cài đặt

Trong phần này sẽ mô tả quy trình từng bước để xây dựng hệ hỏi đáp, bao gồm các thiết lập và môi trường

3.1.1. Dữ liệu huấn luyện

Dữ liệu huấn luyện cho bài toán được thu thập và gán nhãn thủ công dựa trên kênh hỗ trợ sinh viên của trường Đại học Xây dựng. Tập dữ liệu có tổng cộng 3.500 cặp câu hỏi – câu trả lời.

3.2. Cài đặt module truy xuất thông tin

Module truy xuất thông tin sẽ nhận truy vấn dưới dạng các câu hỏi, sau đó sẽ đo độ tương tự giữa câu hỏi đầu vào và câu hỏi – câu trả lời trong cơ sở dữ liệu và tiến hành xếp hạng để đưa ra câu trả lời gần đúng nhất với câu hỏi.

Trong nội dung luận văn này, tôi sử dụng thuật toán OKAPI BM25 như đã trình bày ở chương trước để cài đặt cho module truy xuất thông tin.

Trong kiến trúc này sẽ gồm 3 module nhỏ: Module *tiền xử lý văn bản*, *xếp hạng văn bản* và *đánh chỉ mục tài liệu*.

3.2.1. Tiền xử lý văn bản

Trong bước tiền xử lý sẽ thực hiện loại bỏ các ký tự đặc biệt, tách từ tiếng việt, loại bỏ từ dừng.

3.2.2. Đánh chỉ mục tài liệu

Bước đầu tiên trong việc đánh chỉ mục tài liệu đó là định nghĩ cấu trúc của tài liệu trong hệ hỏi đáp trường Đại học Xây dựng. Một tài liệu sẽ gồm các trường:

- *Nội dung câu hỏi*
- *Câu trả lời*, trường này sẽ lưu nội dung câu trả lời tương ứng với câu hỏi

- *Ý định của câu hỏi*, trường này sẽ nhằm xác định ý định của câu hỏi để giúp cho việc truy xuất thông tin chính xác hơn. Về việc xác định ý định câu hỏi tôi sẽ trình bày ở phần sau.

3.2.3. Xếp hạng văn bản

Để xếp hạng văn bản, tôi sử dụng thuật toán Okapi BM25 đã được trình bày trong phần [2.5.1].

3.2.4. Đánh giá mô hình IR

Một số phương pháp để đánh giá một hệ thống truy xuất thông tin:

a. Precision và Recall.

- **Precision (P):** là độ đo được tính bằng tỉ lệ số lượng kết quả *relevant* trên tổng số lượng kết quả trả về.
- **Recall (R):** là độ đo được tính bằng tỉ lệ số lượng kết quả *relevant* trên tổng số lượng kết quả *relevant* trong tập test.

b. Precision@K

Độ chính xác ở k tài liệu đầu tiên P@k cho biết bao nhiêu phần trăm các tài liệu liên quan nằm trong top k tài liệu nhưng không tính đến các vị trí của các tài liệu liên quan trong số k tài liệu trả về đó. Ví dụ P@10 biểu thị tỉ lệ bao nhiêu tài liệu liên quan nằm trong top 10 tài liệu trả về.

c. Trung bình các precision (Mean average precision - MAP)

Trung bình các precision được tính dựa trên toàn bộ các truy vấn trong tập kiểm thử, MAP sẽ tính bằng cách lấy trung bình các precision của tất cả các truy vấn theo công thức sau:

$$MAP = \frac{\sum_{q=1}^{Q} AveP(q)}{Q}$$

Trong đó Q là số lượng các truy vấn trong tập kiểm thử.

d. Discounted cumulative gain - DCG

DCG sử dụng thang đo mức độ phù hợp của các tài liệu từ tập hợp kết quả để đánh giá mức độ hữu ích hoặc của một tài liệu dựa trên vị trí của nó trong danh sách kết quả. Tiền đề của DCG là các tài liệu có liên quan cao xuất hiện ở vị trí thấp hơn trong danh sách kết

quả tìm kiếm sẽ bị phạt vì giá trị mức độ liên quan được phân loại bị giảm theo tỷ lệ logarit với vị trí của kết quả.

DCG ở vị trí p được định nghĩa là:

$$\mathrm{DCG_p} = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{\log_2(i+1)}$$

Hay được viết gọn lại thành:

$$\mathrm{DCG_p} = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log(i+1)}$$

3.2.5. Kết quả thực nghiệm

Để thực nghiệm mô hình IR tôi tiến hành chia tập dữ liệu như sau:

Về bộ câu hỏi test thì tôi chọn ngẫu nhiên 50 câu hỏi trong các cặp câu hỏi - câu trả lời được lưu trong elasticsearch.

- Đầu tiên cần lưu hết tất cả 400 cặp câu hỏi - câu trả lời trong tập dữ liệu vào trong elasticsearch.
- Sau đó trong 400 cặp câu hỏi - câu trả lời đó em lấy ra 50 câu hỏi để test. Trong quá trình test, khi truy vấn với nội dung của câu hỏi ta thêm điều kiện để loại bỏ id của câu hỏi vừa lấy ra để đảm bảo câu hỏi đó không có trong kết quả trả về.

  Thực hiện các thử nghiệm: so sánh *câu hỏi với câu hỏi, câu hỏi với câu trả lời* và so sánh *câu hỏi với cả câu hỏi và câu trả lời*. Trong thử nghiệm này tôi tiến hành đánh giá với các thuật toán tách từ (tokenizer) khác nhau với các tham số k-top và các phương pháp đo khác nhau. Kết quả được thể hiện theo bảng bên dưới.

  Cách 1: ***Tìm câu hỏi theo câu hỏi:***

Tìm câu hỏi theo câu trả lời						
Ngày tháng	Nội dung			Kết quả		
	INDEXING	*TOKENIZER*	*K*	Recal@K	NDCG@K	MAP@K
12/01/2021	TF-IDF	DEFAULT	1	0,180	0,180	0,180
12/01/2021			3	0,340	0,270	0,247
04/01/2021			10	0,480	0,321	0,271
12/01/2021	TF-IDF	ViTokenizer	1	0,240	0,240	0,240
12/01/2021			3	0,380	0,320	0,300
08/01/2021			10	0,480	0,358	0,319
12/01/2021	BM25	DEFAULT	1	0,140	0,140	0,140
12/01/2021			3	0,320	0,240	0,213
04/01/2021			10	0,400	0,268	0,225
12/01/2021	BM25	ViTokenizer	1	0,200	0,200	0,200
12/01/2021			3	0,300	0,255	0,240
08/01/2021			10	0,400	0,293	0,259

Bảng 3.1: Kết quả tìm kiếm câu hỏi theo câu hỏi

Cách 2: *Tìm câu hỏi theo câu trả lời*

Tìm câu hỏi theo câu hỏi						
Ngày tháng	Nội dung			Kết quả		
	INDEXING	*TOKENIZER*	*K*	Recal@K	NDCG@K	MAP@K
22/01/2021	TF-IDF	DEFAULT	1	0,206	0,457	0,457
22/01/2021			3	0,387	0,409	0,512
22/01/2021			10	0,575	0,488	0,495
22/01/2021	TF-IDF	ViTokenizer	1	0,201	0,449	0,449
22/01/2021			3	0,410	0,422	0,530
22/01/2021			10	0,598	0,497	0,500
25/01/2021	BM25	DEFAULT	1	0,206	0,457	0,457
25/01/2021			3	0,387	0,409	0,512
25/01/2021			10	0,575	0,488	0,495
25/01/2021	BM25	ViTokenizer	1	0,204	0,457	0,457
25/01/2021			3	0,410	0,424	0,533
25/01/2021			10	0,598	0,499	0,504

Bảng 3.2: Kết quả tìm kiếm câu hỏi theo câu trả lời

Cách 3: *Tìm câu hỏi theo câu hỏi và câu trả lời*

Tìm câu hỏi theo câu hỏi và câu trả lời						
Ngày tháng	Nội dung			Kết quả		
	INDEXING	*TOKENIZER*	*K*	Recal@K	NDCG@K	MAP@K
28/01/2021	TF-IDF	DEFAULT	1	0,207	0,457	0,457
28/01/2021			3	0,404	0,424	0,510
28/01/2021			10	0,620	0,514	0,509
28/01/2021	TF-IDF	ViTokenizer	1	0,201	0,449	0,449
28/01/2021			3	0,435	0,440	0,518
28/01/2021			10	0,613	0,512	0,524
01/02/2021	BM25	DEFAULT	1	0,207	0,457	0,457
01/02/2021			3	0,404	0,424	0,512
01/02/2021			10	0,620	0,514	0,509
01/02/2021	BM25	ViTokenizer	1	0,201	0,449	0,449
01/02/2021			3	0,435	0,440	0,519
01/02/2021			10	0,613	0,512	0,526

Bảng 3.3: Kết quả áp dụng IR tìm câu hỏi theo câu hỏi và câu trả lời

3.3. Cài đặt mô hình phân lớp ý định

Phân lớp ý định (intent detection) thực hiện bằng phương pháp text classification, tức là với mỗi câu hỏi thì người ta tiến hành phân loại vào một trong số các loại intent định nghĩa trước.

3.3.1. Xây dựng mô hình phân lớp ý định

Để thực hiện được bằng phương pháp này tôi tiến hành như sau:

a. Đánh nhãn dữ liệu

Để xây dựng mô hình xác định ý định câu hỏi, tôi sẽ sử dụng ontology là các cặp "câu hỏi - ý định" được thu thập từ sinh viên trường Đại học Xây dựng. Tôi sẽ đưa bài toán về việc xây dựng một mô hình phân lớp với các class là các ý định của người hỏi.

***b.* Phân lớp ý định bằng mô hình SVM:**

Sau khi biểu diễn câu hỏi thành vector, tôi đưa vào *mô hình SVM* để tiến hành huấn luyện cho mô hình phân lớp. Kết quả sau đây được thực hiện với 2 phương pháp trên:

	One-hot encoding	*word2vec*
Precision	0.56	0.38
Recall	0.56	0.38
F1-score	0.56	0.38

Bảng 3.4: Kết quả bài toán phân lớp ý định bằng mô hình SVM

Nhận xét: Việc xác định ý định bằng thuật toán SVM trong bài toán này cho kết quả tương đối thấp.

c. Phân lớp ý định bằng deep learning

Vì kết quả của SVM khá thấp nên tôi tiến hành sử dụng mô hình LSTM như đã đề cập trong phần [2.4] để tiến hành phân lớp. Kết quả sau quá trình huấn luyện được trình bày trong bảng dưới đây:

Model	Word Embedding	F1-score
LSTM	Word2Vec	0.906
LSTM	Fastext	0.912
CNN + LSTM	Word2Vec	0.866
CNN + LSTM	Fastext	0.879
BiGRU	Fastext	0.903
baseBERT	Fastext	0.933

Bảng 3.5: Kết quả huấn luyện mô hình phân loại ý định

3.3.2. Tăng cường dữ liệu cho bài toán phân lớp ý định

Sau khi tăng cường dữ liệu như đã đề cập trong phần [2.2.5] ta có tập dữ liệu

$$D_{new} = D_{train} \cup D_{synthesized}.$$

3.3.3. Kết quả huấn luyện sau khi tăng cường dữ liệu

Sử dụng tập dữ liệu D_{new} để huấn luyện mô hình trong mục tăng cường dữ liệu kết quả dựa trên tập kiểm tra như sau:

Model	Word Embedding	F1-score
LSTM	Word2Vec	0.917
LSTM	Fastext	0.923
CNN + LSTM	Word2Vec	0.871
CNN + LSTM	Fastext	0.882
BiGRU	Fastext	0.913
baseBERT	Fastext	**0.953**

Bảng 3.6: Kết quả huấn luyện mô hình phân lớp ý định sau khi fine-tune

3.4. Kết quả thực hiện sau khi kết hợp IR và phân lớp ý định

Sau khi áp dụng các kỹ thuật để cải tiến mô hình phân lớp ý định, tôi tiến hành đưa mô hình phân lớp ý định để lọc các câu trả lời không liên quan đến ý định của câu hỏi. Kết quả được trình bày trong bảng dưới đây:

Indexing	K	NDCG@K		MAP@K	
		Kết hợp ý định	**Không** kết hợp ý định	Kết hợp ý định	**Không** kết hợp ý định
TF-IDF	1	0.821	0.449	0.821	0.449
	3	0.837	0.440	0.833	0.518
BM25	1	**0.841**	**0.449**	**0.841**	**0.449**
	3	**0.852**	**0.440**	**0.866**	**0.519**

Bảng 3.7: Kết quả bài toán sau khi kết hợp IR và phân lớp ý định

Như vậy mô hình đã cho kết quả khá tốt, với tham số indexing BM25 và sử dụng ViTokenizer cho kết quả MAP@K [3.2.4] đạt 0.866.

KẾT LUẬN VÀ KIẾN NGHỊ

Áp dụng hệ thống hỏi đáp tự động giải quyết nhu cầu hỏi đáp, hỗ trợ sinh viên trường Đại học Xây dựng mang lại hiệu quả cao, giúp sinh viên dễ dàng tiếp cận thông tin từ phía

nhà trường đồng thời làm giảm khối lượng công việc tiếp nhận và giải quyết thắc mắc, nhu cầu thông tin từ phía sinh viên cho các phòng ban trong trường. Các tiếp cận xây dựng hệ thống trả lời tự động dựa trên truy xuất thông tin có thể sử dụng được dữ liệu các câu hỏi – câu trả lời được cung cấp bởi các phòng ban trong Trường phục vụ cho việc trả lời tự động. Để câu trả lời tự động dựa trên truy xuất thông tin được chính xác, việc xác định ý định câu hỏi và sử dụng ý định câu hỏi để loại bỏ các câu trả lời sai không phù hợp với câu hỏi mang lại kết quả tốt.

Tuy nhiên việc tiếp cận này vẫn còn nhiều hạn chế, hệ thống trả lời tự động hoạt động tốt với các câu hỏi giống nhau và thường lặp đi lặp lại nhưng kém hữu ích với các câu hỏi có nội dung mới của sinh viên và câu hỏi mang tính chất tư vấn và hỗ trợ cho từng cá nhân. Vì vậy cần phải bổ sung và cập nhật dữ liệu thường xuyên cho hệ thống.